Research and Development of a Reference Model for Intrusion Handling Systems in Wireless LANs

Shahram Salek Zamankhani

A thesis submitted in partial fulfilment of the requirement of
London Metropolitan University
for the degree of
Doctor of Philosophy

Communications Technology Group
Faculty of Computing
London North Campus

July 2011

LONDON
metropolitan
university

Abstract

Wireless Local Area Networks (WLANs) are different from the traditional wired LANs in terms of their easy exposure to potential threats and vulnerability to exploitation. Hence, there is an urgent need for effective Intrusion Handling Systems (IHSs) and the methods for their evaluation. Analyses of the IHSs (Snort Wireless, Bro and Kismet) for WLANs performed in this work have shown that they are all built as proprietary systems. These IHSs do not consider the existence of other IHSs nor do they try to determine ways to establish inter-IHS collaboration in order to achieve a better security for WLANs. Based on these analyses the unified presentation of the features and architectures of these IHSs have been developed.

In this thesis, it has been identified a distinct absence of a standardised reference model for IHSs. Hence, a novel reference model is proposed for a scalable distributed IHS that defines system architecture, specifying the need for identification and response systems with associated sets of methods and inter-IHS communications protocol. A model of IHS description for *ad hoc* networks with a present intruder has been developed and a need for Unified Intrusion Handling Report Format, Inter IHS Message and Inter IHS Communication Subsystem for inter IHS communications has been identified. Ontology-based approach for modelling of IHS has been suggested and formally adopted throughout. IHS Modeling Ontology (IHSMO) has been developed based on the previous analyses of IHSs. This IHSMO is unique because it has components specifically addressing WLANs. The IHSMO is built using the Hozo ontology editor that allows both specification and verification of the model's integrity, which established a common framework for comparison and benchmarking. For evaluation purposes the ontology-based models of the same IHSs (Snort Wireless, Bro and Kismet) have been built using Hozo ontology editor. Comparison of their ontologies was undertaken in a unified way using IHSMO as a reference. This enabled the functionality gap analysis to be performed using IHSMO concepts and slots in order to demonstrate lack of essential functionalities in some systems. This further informed in the evaluation of the proposed model. The approach allows adopting and extending the IHSMO if knowledge changes in future.

Acknowledgement

I would like to express my sincere gratitude towards my director of study and main supervisor Professor Algirdas Pakštas for his continued, consistent guidance, encouragement and academic support throughout the conduct of the research. His vast experience in this area of research and enthusiasm motivated me immensely. Without his support, it would not have been possible to complete the research work.

My special thanks to my second supervisor Dr. Bal Virdee for his kind support and encouragement during the research activity.

Furthermore, many thanks go to Dr. Tamara Matveeva for her thoughtful comments.

Finally, my special gratitude must be extended to my family.....many absent hours. Without their support and patience, this thesis would not have been completed.

Shahram Salek Zamankhani
London Metropolitan University
March, 2010

Published Papers

1. S. Salekzamankhani, A. Pakstas, B. Virdee, "An Ontology Approach to Construction of Identification Subsystem for Intrusion Handling Systems in Wireless LANs", World Congress on Engineering (WCE), the 2010 International Conference of Wireless Networks (ICWN'10), 30 June - 2 July 2010, London, UK, ISBN: 978-988-17012-9-9.

2. S. Salekzamankhani, A. Pakstas, B. Virdee, "Ontology Approach to Construction of Response and Management Console Subsystems for Intrusion Handling Systems in Wireless LANs", World Congress on Engineering (WCE), the 2010 International Conference of Wireless Networks (ICWN'10), 30 June - 2 July 2010, London, UK, ISBN: 978-988-17012-9-9.

3. S. Salekzamankhani, A. Pakštas, "Why we need a reference model for intrusion handling systems for Wireless LANs?" Telecommunications and Computer Networks, SoftCOM2007, 15th International Conference, 27-29 Sept, 2007 pp.1– 6. ISBN: 978-953-6114-95-5.

4. A. Pakštas, S. Salekzamankhani, B.Virdee, "Fighting Intrusions in Wireless LANs: A Need for the Reference Model". Proc. 2nd IEEE and IFIP International Conference in Central Asia on the Next Generation of Mobile, Wireless and Optical Communications Networks, (ICI 2006), Tashkent, Uzbekistan , Sep. 19, 2006, ISBN: 1-4244-0543-2.

5. S. Salekzamankhani, A. Pakštas, B.Virdee, "Towards Development of a Reference Model for Intrusion Detection Systems for Wireless LANs", IEEE Globecom 2005, Workshop on Adaptive Wireless Networks, AWIN Nov. 30, 2005, San Louis, USA.

Table of Contents

Abbreviations and Acronyms

ACL: Access Control List
ADS: Anomaly Detection System
AIS: Anomaly Identification System
ARS: Adaptive Rate Selection
AP: Access Point
API: Application Programme Interface
ARP: Address Resolution Protocol
ASCII: American Standard Code for Information Interchange
BSSID: Basic Service Set Identifier
CVS: Concurrent Versioning System
DARPA: Defence Advanced Research Projects Agency
DCLI: Data Capture library Interface
DDOS: Distributed Denial of Service
DHCP: Dynamic Host Configuration Protocol
DNS: Domain Name System
DOS: Denial of Service
DR: Design Rationale
FTP: File Transfer Protocol
HTTP: Hyper-Text Transfer Protocol
IBSS: Independent Basic Service Set
ICMP: Internet Control Message Protocol
IDMEF: Intrusion Detection Message Exchange Format
IDS: Intrusion Detection System
IDWG: Intrusion Detection Working Group
IDXP: Intrusion Detection Exchange Protocol
IHS: Intrusion Handling System
IHSMO: Intrusion Handling System Modelling Ontology
IIHSCP: Inter IHS Communication Protocol
IIS: Intrusion Identification Subsystem
IPS: Intrusion Prevention System
IR: Infrared
ISMO: Identification Subsystem Modelling Ontology
ISO: International Organisation for Standardisation
KB: knowledge base
LAN: Local Area Network
LLC: Logical Link Control
MAC: Media Access Control
MANET: Mobile *Ad hoc* Networks
MCO: Management Console Ontology
MITM: Man In The Middle
NIC: Network Interface Card
NIDS: Networked Intrusion Detection System
NIHS: Network IHS
OIIA: Open IHS Interconnection Architecture

OSI: Open Systems Interconnection
PRNG: Pseudo Random Number Generator
RNG: Random Number Generator
RF: Radio Frequency
RFC: Request for Comment
RPC: Remote Procedure Call
RSDB: Rule Set Data Base
RSMO: Response Subsystem Modelling Ontology
RSS: Received Signal Strength
RST: readability, strength, tone, used to record the quality of radio signals
RTT: Round Trip Time
SMI: Structure of Management Information
SNM: Simple Network Management Protocol
SSID: Service Set Identifier
SYN: Synchronize packet in transmission control protocol
TCP/IP: Transmission Control Protocol / Internet Protocol
TFTP: Trivial File Transfer Protocol
UDP: User Datagram Protocol
UIRF: Unified Intrusion Report Format
URI: Uniform Resource Identifier
UIHRF: Unified Intrusion Handling Report Format
WAP: Wireless Authentication Protocol
WEP: Wired Equivalence Privacy
WIDS: Wireless Intrusion Detection System
IHMD: Wireless Intrusion Handling Message Exchange Database
IIHSM: Inter IHS Message
WI-FI: Wireless Fidelity
WLA: Wireless Local Area Network
W3C: World Wide Web Consortium
XML: Extensible Markup Language
ZBIHS: Zone Based Intrusion Handling System

List of Figures

List of Tables

1. An Introduction to Security Issues in Wireless LANs

The rapid deployment of IEEE 802.11, known as Wireless Local Area Networks (WLANs) can be credited to their obvious benefits over the wired LANs due to the elimination of cables. Unlike wired networks in which clients (workstations) send and receive data across cables and wires, a WLAN, uses radio frequencies (RF) or infrared light (IR) for data transmission. The use of RF is far more popular than IR due to its longer range, higher bandwidth, and wider coverage.

Provision of sufficient communications and information security is an important issue for many organisations nowadays. A number of mechanisms and techniques have been applied for defending network resources. Firewalls, encryption, authentication, vulnerability checking and other techniques can all help to enhance the network security. However, WLANs are coming with embedded flaws such as potentially reduced security (see Appendix A and B for various attacks on WLANs). This is reflected in many publications (see e.g. [Harr1997, Zhan2000, Gill2006, Sale2007]) and research towards building more secure WLANs is still undergoing.

Hence this project is focusing on development of a new reference model for Intrusion Handling Systems (IHSs) in WLANs. Such IHS reference model for WLANs should be capable of being used in the same manner as the OSI/ISO model for computer networks (i.e. use of standardised terminology, clearly defined functional components, etc.), namely as a reference for evaluation of the current, and development of the future generations of IHSs for WLANs.

1.1 Operation Principles of WLANs

WLANs use the following main types of wireless network devices [Tane2003]:

- Wireless Network Interface Card (WNIC) is equivalent to an ordinary NIC used in wired networks. The difference is that WNICs use radio interfaces to communicate with each other.
- Wireless Access Point (WAP) is an equivalent of a hub that assumes like a logical "star" topology with typical coverage area of 30 meters indoors and 90 meters

outdoors. Like hub it retransmits packets to all other connections including ordinary wired LAN if it is connected to a WAP.

- Wireless Router is a combination of a WAP, a switch and a router which involves OSI layer two and three devices respectively.
- Wireless bridge (OSI layer two device) interconnects a wired LAN usually located in separate building to larger network in the same manner as a wired bridge.
- Range Extender is an equivalent of wired Ethernet repeater which simply extends the range of a WLAN by retransmitting and boosting the signal.

WLANs essentially operate in two modes: *ad hoc* and infrastructure (Figure 1.1 and 1.2)

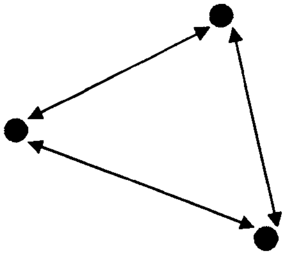


Figure1.1. *Ad hoc* mode wireless Network

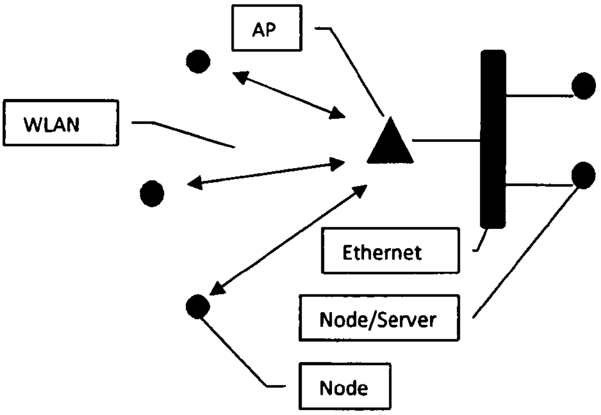


Figure1.2. Infrastructure mode wireless network

Ad hoc WLANs are formed by wireless nodes that communicate directly to each other using their WNICs. It works the same way as peer-to-peer networking or the so called Independent Basic Service Set (IBSS). The first *ad hoc* node in the area establishes an IBSS and starts sending beacons, which are essential to maintain synchronization among other *ad hoc* nodes. Other *ad hoc* nodes can also join the WLAN after receiving a beacon and accepting the IBSS elements and parameters, for example, beacon interval, etc found in the beacon frame's format. All nodes that join the *ad hoc* network must send a beacon periodically. If a node doesn't hear a beacon from another node within a very short delay period, then the node assumes that no other nodes are active in this *ad hoc* network, and a new beacon needs to be sent. Infrastructure networks (also called Extended Service Set-ESS) work by wireless nodes communicating via a WAP which retransmits received packets appropriately. The main role of a WAP is to form a bridge between wireless and wired LAN, i.e. they form a part of the wired network infrastructure. When there is an Access Point (AP) present, stations do not communicate on a peer-to-peer basis but do it via WAP. The absence of a WAP in *ad hoc* WLAN means that *ad hoc* WLAN nodes must take on more of the MAC Layer responsibilities.

In infrastructure mode, a node sends a probe request frame when it needs to determine which WAPs are within the range in order to obtain information from it and unlike the *ad hoc* mode, only that WAP sends beacons to announce its presence and relay information, such as timestamp, Service Set Identifier (SSID), and other parameters. Obviously, it is similar to the wired "star" topology and works in the same way.

To establish a relationship with a WAP, wireless clients scan the frequency band to select a WAP with the best communications quality. There are two types of scanning, active and passive.

An *active scan* is where the wireless client sends out a "probe request" on specific channels and assesses its response. A probe is a request from client to discover a WAP. A *passive scan* is where the wireless client assesses the communications quality from the beacon message of the WAP. The beacon is a message send by WAP to inform clients of their existence. The Access Point maintains a list of associated stations in the MAC database table. It also records the client's capability (data-rate). The client's Media Access Control (MAC) address is well maintained in the bridge learns table associated with the port it is located on.

To control access to the network infrastructure via an authentication, the clients identify themselves to other clients or a WAP prior to transmission of data traffic or association, using either the Wired Equivalent Privacy (WEP) methods which are open system or shared key authentications. Thus, WAPs primarily concerns are, allowing unauthenticated packets to enter WLAN. Because of that the easiest and the earliest method of defending such packets can be achieved by some system operating at WAPs.

1.2 Wireless Networks Security Threats and Vulnerabilities

All the attacks explained in appendix A.5 can threaten or damage the WLANs either destructively or non-destructively. Damages include but not limited to leaking confidential information, message contamination and host impersonation. From now on it will be focused on *ad hoc* mode of wireless networks due to its natural vulnerabilities compare to infrastructure mode. This means that wireless *ad hoc* networks do not have a clear line of defence hence hosts in *ad hoc* networks must be prepared for encounters with either direct or indirect adversaries [Zhan2003].

Mobile hosts are capable of roaming independently and thus, they and their infrastructure must be prepared to function in a format that they do not trust peers without appropriate authentications.

WLANs are special due to the lack of centralised audit/access points as compared to wired networks where traffic monitoring is usually done at switches, routers and gateways. The mobile *ad hoc* environment does not have such traffic concentration points where the IDS can collect audit data (logging history) for the entire network. Therefore, at any one time, the only available audit trace will be limited to communication activities taking place within a part of the network accessible by wireless devices [Zhan2000].

Due to decentralised format of wireless *ad hoc* environment, wireless network algorithms rely on the cooperative participation of all hosts and the infrastructure, therefore lack of this centralised authority means that the adversaries can exploit this vulnerability for new types of attacks designed to break the cooperative algorithms [Zhan2003].

Another issue with wireless networks is the limited battery power and amount of bandwidth, which makes the communications between mobile hosts much slower compare to wired networks.

Wireless *ad hoc* security threats and vulnerabilities have inherent features of open air environment, dynamic changing of its topology, cooperative algorithms, lack of centralised monitoring and management point and finally lack of clear line of defence.

In summary, the following is the most significant natural characteristics and vulnerabilities to attacks in *ad hoc* networks:

- Mobility causing dynamic network topology
- Scalability of area coverage
- Multi-hopping
- Limited bandwidth
- Resource limited operation causing a need for energy conservation
- Limited physical security
- Decentralised network control

1.3 Intrusion Handling Process in the Current Systems

A number of mechanisms and techniques have been used for defending network resources. Firewalls, encryption, authentication, vulnerability checking and other techniques can all help to enhance the network security. As mentioned earlier, the main problem with WLANs is due to its nature because *intrusion (unauthorised access)* may happen where the *intruder (unauthorised user)* is no longer gaining physical access to the network in order to compromise its security. It creates new vulnerabilities that do not exist in traditional wired networks; making many proven security measures ineffective [Zhan2000].

1.3.1 Defining Intrusion Handling System

Anderson first introduced the concept of *intrusion detection* in 1980 [Ande1980] and defined intrusion as "*an intrusion as an attempt or a threat to be a deliberate* *unauthorised attempt to access or manipulate information or render a system unreliable or unusable*". Handling of intrusions may involve various activities while their implementation depends on the available resources, intelligence of devices and existence of the enabling technologies. IDS is probably the earliest known term describing an embodiment of a set of intrusion handling functions in the particular hardware/software components which was varying from a vendor to vendor by its functionality, efficiency, used terminology, etc. The first task, here is *detection* of the presence of the intruder. To counteract this potential *threat,* the system should be capable of detecting the intruder and take appropriate re*action* by gaining possibility to read/write information or change system parameters in order to block its *access* and/or hostile its *activity* using a set of actions. Dealing with intrusions in any system involves various activities related to intrusion handling and because of the terms such as "detection" or "identification", "prevention" or "reaction" that are often used interchangeably, a variety of new functions have been created which is primarily used in the proprietary systems. These functions can be implemented in various commercial and open source systems under different names such as IDS, IPS, etc. Extensive research has been done in order to protect wired networks by developing efficient IHSs [Mukh1994, Skar1999, Cans2002, Huan2003, Kong2003, Krug, Vign2003, Vign2004, Bake2005, Fari2006]. More modern definition of intrusion related to networks in general is given in [lim2003] as: "a sequence of related actions by a malicious adversary that results in the occurrence of unauthorised security threats to a target computing or network domain. An intrusion consists of a number of related steps performed by the intruder who violates a given security policy". These systems usually monitor users, system and network level activities continuously and normally have a centralised decision making entity. In the rest of this thesis, a general term "Intrusion Handling System" (IHS) is used as discussed in [Sale2005, Pakš2006, Sale2007] except where more details about particular IDS/IPS systems are given.

1.3.2 IHS Functions and Basic Terminology

In [Lim2003], it is stated that Intrusion Detection "is a sequence of *identifying* and *responding* to malicious activities targeted at computing and network resources" (italics are emphasised by authors). Therefore, IHS can also be referred to as IIRS (Intrusion Identification and Response System). This is one good example of the confusing terminology where detection also includes response. The suggested term is *intrusion identification* because it better describes activities of the IHS before anything is done to deal with intrusion (i.e. before any response action).

There are two main approaches to intrusion identification; *analysing "live" packets* and; *use of audit data logs* (stored data about previous sessions). Methods based on the audit data logs can either be identifying *misuse* and/or *anomaly* in the user behaviour.

Intrusion identification is normally followed by the *response* actions which can be of these types:

- *Informing (passive)* action such as *logging* of the intrusion details or *real-time (alarming) notification* to a system administrator who may require human interaction,
- *Defending (active)* reaction such as disabling the entire network in order to protect valuable information from access. This normally can be pre-programmed for automatic implementation,
- *Defending/configuring (manual)* reaction performed by the system administrator.

1.3.2.1 IHS Classifications Based on Audit Data

Audit data are specific information which IHS systems can work on to determine occurrence of any intrusions. Most IHSs take either a network-based or a host-based approach for identifying and analysing attacks based on the audit data gathered in the system. These data are specific patterns that usually indicate malicious or suspicious intent [Lee2000]. Therefore, based on the audit data source of an IHS, it can be classified as following: (a) network-based, (b) host-based, or (c) hybrid.

1.3.2.1.1 Network-based

An IHS is network-based when it looks for traffic patterns in network traffic. For example, it is a system which is located in the gateway and inspects the forwarded packets. It monitors and analyses all traffic in real-time as it travels across the network. It can also analyse the payload within the packet in order to identify the type of the host application.

Advantages of network-based IHS

Firstly, it doesn't require modification of existing entities in the network. Secondly, it doesn't depend on existing entities in the network so they are more failure resilient and thirdly installation of network-based IHS boxes are quite straight forward.

Disadvantages of network-based IHS

Firstly, it examines the traffic only on the network segment to which it is directly connected and such localised vision is especially a problem in switched Ethernet environments. Secondly, the encryption at the application layer, high data rate and switched networks present significant problems for such systems. Thirdly, it is susceptible to a type of attack known as "evasion" i.e. the attacker is aware of the existence of the IHS system in the target network and is trying to evade the IHS detection [Basi2005].

1.3.2.1.2 Host-based

Host-based IHSs look for the specific patterns in log files and they can recognise and detect actions such as repeated failed access attempts or changes to critical system files. They perform this by accessing log files and monitoring real-time system usage on every host [Lee2000].

Advantages of host-based IHS [Yaro2000] are: (a) it provides much more detailed information than network-based IHS, e.g. it can determine what commands the user entered, what files, he/she accessed, etc.; (b) it tends to have lower false alarm rates than network-based IHS and this is because it provides more specific and focused information than network-based IHS; (c) it can be completely self-contained.

On the other hand the disadvantages of host-based IHS are: (a) it has to be installed on the particular device one wants to protect; (b) often configuration of protected entity has to be altered; (c) it also has a local vision restrictions and can be completely unaware of the network environment; (d) there are significant difficulties (compared to network-based IHS), when testing host-based IHS in an off-line manner and some researchers are directed to employ real-time testing [Basi2005].

1.3.2.1.3 Hybrid-based

A combination of a host- and network- based approach seems to be the most suitable for WLANs due to their changing topology.

1.3.3 Anomaly and Misuse Identifications

Anomaly identification monitors activities that significantly deviate from the established normal usage profiles known as *anomalies*. For example, the normal profile of a user may contain the averaged frequencies of some system commands used in his or her login sessions. If for a session that is being monitored, the frequencies are significantly low or high then an anomaly alarm will be raised. Therefore, anomaly identification does not require prior knowledge of intrusion and can thus detect new intrusions however it may exhibit high rates of false positives [Gaya2003]. Hence it requires the creation of statistical user profiles (profile of normal activities).

Misuse identification is also known as signature based identification and it relies on a large database of known intrusion attacks (*signatures*) in order to match and identify suspected intrusion. This database must be updated regularly. The misuse identification has a low rate of false alarm. However, it is not capable of detecting novel attacks.

Hybrid identification combines both anomaly and misuse identifications techniques at the same time in order to exploit the best features of both techniques.

In view of above definitions the anomaly identification is about finding the normal usage patterns from the audit data, whereas misuse identification is about encoding and matching the intrusion patterns using the audit data [Lee2000].

1.3.4 Recent research on Intrusion Identification

In 1998, U.S. Defence Advanced Research Projects Agency (DARPA) sponsored the first intrusion identification evaluation to survey the state-of-the-art of research in intrusion identification [Krug]. The results indicated that the *research* IHS systems were much more effective than the leading *commercial* IHS systems. However, even the best research systems failed to identify a large number of new attacks, including those that led to unauthorised user or root access [Zhan2000].

Sekar presented a new approach for network intrusion identification [Seka1999]; a key feature of which is a domain specific language for capturing patterns on normal and/or abnormal network packet sequences. The language supports concise and easy-to-write attack patterns. This in turn increases the confidence in attack specifications and reduces the development and debugging times needed for defending against new attacks. Convenient and excessive abstractions for aggregating data across multiple network packets are developed with also efficient implementation of these abstractions. In addition, efficient implementation of the pattern matching operations which is needed in the language was also developed.

Sekar also carried out specification-based anomaly identification through various experiments, showing that the new approach combines the primary benefits of anomaly identification and specification-based identification, namely, good identification of unknown attacks and low false alarm rates [Seka1999]. At the same time, the new approach alleviates the principal problems associated with either approaches. Specification development is simplified by protocol specifications and only a handful of protocols need to be specified in order to identify most attacks. In [Seka1999], it is shown that protocol specifications simplify the manual feature selection process that often plays a major role in other anomaly identification approaches. In particular, most attacks discussed in the experimentation section could be identified by simply monitoring frequency distribution information associated with state machine transitions. Identification of other attacks requires further partitioning of frequency information based on sources and destinations of network packets.

The work by Zhang has stated that a better architecture for intrusion identification in wireless *ad hoc* networks should be distributed and cooperative [Zhan2000]. A statistical anomaly identification approach should be used. The trace analysis and anomaly identification should be done locally in each node and possibly through cooperation with all nodes in the network. Further intrusion identification should take place in all networking layers in an integrated cross layer manner.

In [Dona2002], a system is presented for characterising the dynamics of conversations occurring across a boundary of an organisation's network. This system appears to have a significant potential for monitoring the health of networks as well as intrusion identification. The system is modelled after statistical mechanic methods for developing notional trends from large amount of data. This system has been observed operating in real network environments and has produced some intriguing results. Specifically, the system has been shown to identify phenomena that are not identified by any other network sensors. While not always an indicator of malicious activity, this has proven to be a significant resource for network administrators as an aid to understanding how their network is being used.

[Ning2002] is presented a practical method for constructing attack scenarios through alert correlation, using prerequisites and consequences of intrusions. The approach was based on the observation that in series of attacks, component attacks are usually not isolated, but related as different stages, with the earlier stages preparing for the later ones. It was proposed a formal framework to represent alerts along with their prerequisites and consequences, and developed a method to correlate related hyper-alerts together, including an intuitive representation of correlated alerts that reveals the attack scenario of the corresponding attacks. An online tool based on the formal framework was also developed. The initial experiments have demonstrated the potential of this method in correlating alerts and differentiating false and true alerts.

[Wagn2002] has shown how attackers may be able to evade identification in host-based anomaly IHSs, and there is presented initial evidence that some IHSs may be vulnerable. It is not clear how serious a threat mimicry attack will be in practice. Nonetheless, the lesson is that it is not enough of merely protect against today's attacks. One must also defend against tomorrow's attacks, keeping in mind that tomorrow's attackers might adapt in response to the protection measures, we deploy today. It was suggested that more attention should be paid in the intrusion identification community to ensure security against adaptive attackers.

In [Huan2003], a set of rules is presented for identifying the type of attacks for well known attacks. In some cases, the rules can also identify the attacking or misbehaving nodes.

A practical passive attack model against *ad hoc* routing protocols is presented in [Kong2003]. It's demonstrated that existing a*d hoc* routing protocols are vulnerable to passive attacks. The work shows the necessity to devise untraceable *ad hoc* routing schemes to protect wireless nodes' mobile privacy in hostile environments. In addition to the traditional content privacy concerns, mobile nodes need more support to ensure their location privacy, anonymity/identity privacy, and motion pattern privacy.

A methodology of fusing identification information from diverse intrusion sensors is outlined in [Shan2003]. The data fusion techniques have been applied to fuse alerts generated by different detectors that signal the potential presence of an intrusion. The advantage of this technique is that dissimilar and independent detectors of intrusion can be combined efficiently without increasing false-alarm rates. To relate the local identification to containment, further investigation was carried out on a mathematical model to analyse intrusion spreading and throttling effects. The results suggested that an autonomous response system that reacts at the local intranet level in under ten seconds can be effective in keeping a majority of aggressive intrusions from spreading unchecked.

The general notion of contextual signatures as an improvement on the traditional form of string-based signature-matching used by Network Intrusion Handling Systems (NIHSs) is developed by [Somm2003]. Rather than matching fixed strings in isolation, contextual signatures augment the matching process with both low-level contexts, by using regular expressions for matching rather than using simply fixed strings, and high-level contexts. This is achieved by taking advantage of the rich, additional semantic contexts made available by Bro's protocol analysis and scripting language. By tightly integrating the new signature engine into IHS system of Bro's event based architecture, several major improvements over other signature-based NIHSs will be achieved which frequently suffer from generating a huge number of alerts. By interpreting a signature-match only as an *event*, rather than as an *alert by itself* should leverage Bro's context and state-management mechanisms to improve the quality of alerts. In [Somm2003], several examples of the power of this approach are shown. These are matching requests with replies, recognising to exploit scans, making use of vulnerability profiles, and defining dependencies between signatures to model attacks that span multiple connections.

The design of a non-overlapping Zone Based Intrusion Handling System (ZBIHS) for mobile *ad hoc* networks is discussed in [Sun2003]. Based on a local Markov chain based anomaly identification engine, an aggregation algorithm for ZBIHS is presented and a suitable Mobile *Ad hoc* Network IHS (MANET IHS) alert data model is discussed. Using the routing disruption attack as the threat model, it's carried out extensive simulation studies and demonstrated the effectiveness of this system. Using the aggregation algorithm under the zone based framework, the false alarm ratio is reduced to an acceptable level, especially at high mobility levels. The gateway node can also present more diagnostic information about the attacks. Therefore, the local IHS agents and the aggregation algorithm under the zone based framework complement each other to form a complete MANET IHS.

The State Transition Analysis Technique (STAT) framework, the corresponding extension process, and the result of applying the framework to develop a family of IHSs have been described in [Vign2003]. The STAT framework is an approach to the development of IHS based on the State Transition Analysis Technique.

In [Dreg2004], it is stated that in large-scale environments, network IHS face extreme challenges with respect to traffic volume, traffic diversity, and resource management. This study is discusses the operational experiences with an IHS in a high performance network.

In [Holl2004], it is stated that an explicit concept of suspicion shows promises in assisting a model-based IHS. In addition, deductive links between the suspicions helped to confirm that these suspicions are correct, and also clarifies the relationships between events so that they can be clearly explained to a human analyst, or any other consumer. The separation of the deductive graph and the suspicion graph allows the use of certainties associated only with normal behaviour in the deductive graph, and focus exclusively on relationships between potential indications of malicious behaviour in the suspicion graph, only updating a view of the world once the suspicions have been confirmed.

A flow-based abnormal network traffic identification method and its system prototype is presented in [Kim2004]. This method is efficient, since it can reduce system overhead in the processing of packet data by aggregating packets into flows. It can detect the traffic of several attacks with similar traffic patterns using one identifying function. This function can cover even *mutant attacks* that use new port numbers or a changed payload. It also increased the detecting accuracy. When identifying abnormal traffic, the parameters that used can reflect changes in traffic characteristics during attacks. However, this proposed method strictly focuses on the identification of DoS/DDoS attacks. If an attack does not influence network traffic, it is difficult to identify this type of attack.

A technique for the black-box testing of network-based signatures and a tool based on such a technique is presented in [Vign2004]. The tool takes *exploit templates* and generates *exploit mutants*. These mutants are then used as test cases to gather some insight on the quality of the signatures used by network-based IHSs. This tool is applied to ten common exploits and applied the test cases against two of the most popular network-based IHSs. The results showed by composing several evasion techniques, it is possible to evade a substantial number of the analysed signatures. Therefore, even though the tool does not guarantee complete coverage of the possible mutation space, it is useful in gaining assurance about the quality of the signatures of an IHS.

The methodology for evaluating intrusion identification in tactical MANETS is presented in [Watk2004]. It is stated that intrusion identification is a necessity in tactical MANETS because encryption techniques and security policy are not enough to protect an open medium. There is clearly a need to identify the strengths and weaknesses of conventional IHS in the environment. If current IHS weaknesses in the environment are identified, directions to new techniques may be explored.

Two problems are addressed in [Yeo2004]. Wireless monitoring technique and its applications in MAC traffic characterisation and network diagnosis. It first identifies the pitfalls of wireless monitoring and provides two feasible solutions, namely merging multiple sniffers and their placements.

A methodology and a tool for system level optimisation using graph-based partitioning and tree-based matching of large intrusion identification pattern databases is discussed in [Bake2005]. By optimising at a system level and considering an entire set of patterns instead of individual string matching units, these tools allow more efficient communication and extensive reuse of hardware components for dramatic increases in area-time performance.

A scheme for the classification of intrusion identification sensor alerts based on their role as part of goal-oriented multistage attacks is discussed in [Math2005]. It's also shown some examples of Snort alert classifications based on this scheme and have demonstrated experimentally how this provides enhanced situation awareness.

It is explained how this fits into the Indications & Warning framework advocated by the military.

A Bayesian game formulation for IHS implementation in wireless *ad hoc* networks is proposed in [Liu2006]. In these games, each player tries to maximise its payoff. The attacker seeks to inflict the most damage in the network without being identified, while the defender tries to maximise his defending capabilities with a constraint on its energy expenditure for heavy traffic monitoring using IHS, and without complete information on the type of his opponent.

The accuracy, robustness and usability of the Received Signal Strength (RSS) and Round Trip Time (RTT) identification techniques and the event based correlation engine are demonstrated in [Gill2006]. Both the identification techniques accurately and precisely identified the session hijacking attacks at the correct frame numbers and hence proved to be reliable and effective. However, accuracy of the identification techniques depends upon selection of appropriate threshold values. Low threshold values tend to result in a greater volume of false positives while higher threshold values lead to more false negatives.

In [Fari2006], it is shown that reliable client identifiers can be created using signal strength measurements reported by access points acting as sensors. It's shown that while malicious clients can lie about their MAC addresses, the signal prints, they produce are strongly correlated with their physical location. It's also demonstrated that by tagging packets with their signal prints and crafting proper matching rules, a wireless network can identify a large class of effective denial-of-service attacks based on MAC address spoofing. There are a number of attacks presented that can be easily mounted in IEEE 802.11 networks.

The benefits of transitioning from taxonomies to ontologies and ontology specifications languages have been discussed in [Unde2003]. They have produced an ontology specifying a model of computer attack using the DARPA Agent Markup Language+ Ontology Inference Layer, a descriptive logic language. They have compared and contrasted the IETF's IDMEF, as an standard that uses XML to define its data model, constructed using DAML+OIL as an ontology language. In their research, they have focused on low level kernel attributes at the process, system and network levels, to serve as those taxonomic characteristics. They have illustrated the benefits of utilising ontology by presenting use case scenarios within a distributed IHS. Although they have clearly shown the boundary between taxonomies and ontologies and transmission of one to other but they have not discussed the benefit of developing a reference model for wireless IHS systems and they have clearly addressed this in their paper.

1.3.5 Taxonomy of IHSs and Relevant Methods

In general IHS systems can be classified as shown below.

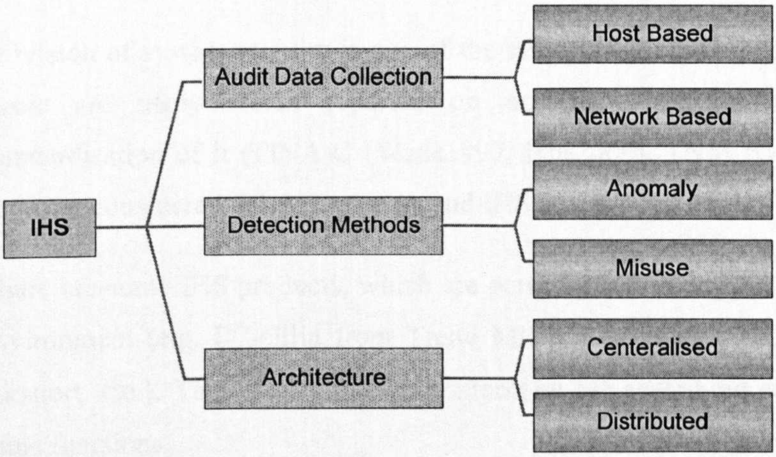


Figure 1.3. General IHS Taxonomy

Figure 1.3 shows a general IHS taxonomy however IHS systems can also be classified based on their structure as shown in Figure 1.4.

Although many of these techniques have proven to be effective but they often would not produce expected results when applied to WLANs. This is due to their nature, namely that the current wireless technology offers little means to control of the coverage area. As a result, WLANs are still susceptible to attacks from intruders who take advantage of the access via open air as well as exploitation of the operating system flaws [Spen2000, lim2003].

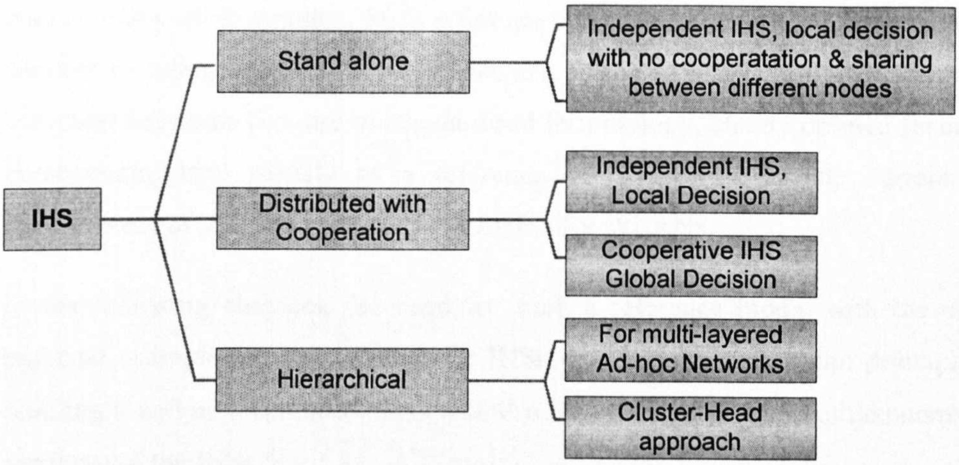

IHS
Stand alone
Distributed with Cooperation
Hierarchical
Independent IHS, local decision with no cooperatation & sharing between different nodes
Independent IHS, Local Decision
Cooperative IHS Global Decision
For multi-layered Ad-hoc Networks
Cluster-Head approach

Figure 1.4. Structural IHS Taxonomy

Provision of system security is one of the tasks for network management systems. There are many works reported on network management issues towards standardisation of it (TINA-C [Wade1997, Hils2000], TNM [Grif1997]) but they have not considered WLAN security and IHS issues.

There are some IHS products, which are partially addressing the needs of WLANs environment (e.g. PC-cillin from Trend Micro, Madge WLAN Probe 2, Kismet, Airsnort, etc.). They use different terminology for describing similar or even the same functions.

Much of the previous research on IHSs mentioned earlier [Mart2000, Zhan2000, Kach2003, Fari2006] either focus on only layer three of OSI model vulnerabilities or propose methods to secure or design their own proprietary IHS which is not compatible with open source or commercial systems as other existing IHSs.

Also, despite work undertaken on IHSs for wireless *ad hoc* networks, (see e.g. [Mart2000, Zhan2000, Zhan2001, Kach2003, Kong2003, Sun2003, Unde2003, Zhan2003, Yeo2004, Watk2004, Ihea2005, Stam2005]), there is still no clear standardised reference model for IHSs, which can help to design, compare and evaluate the IHSs. Thus, this is (a) confusing for the end-users when they have to choose right IHS for their purpose, and (b) does not help to design IHSs in the same way as many other systems. Such reference model for IHS for WLANs should be capable of being used in the same manner as the OSI/ISO reference model for computer networks (i.e. use of standardised terminology, clearly defined functional components, etc), namely as a reference for evaluation of the current, and development of the future generation of IHSs for WLANs.

In the following chapters, the need for such a reference model with the use of practical examples from most popular IHSs will be emphasised and principles of building it will be formulated. This will also lead towards creation of taxonomy and ontology of the IHSs.

1.4 Current Issues with IHSs for WLANs

The traditional techniques of protecting networks with current IHSs as it is done in the wired networks [Mukh1994, Bala1998, Seka1999, Krug2000, Lee2000, Yaro2000, Wagn2002, Basi2005] are no longer sufficient and effective in mobile *ad hoc* networks [Zhan2000, Zhan2003]. The most important difference is that WLANs are special due to the lack of centralised audit/access points as compared with wired networks where traffic monitoring is usually done at switches, routers and gateways. The mobile *ad hoc* environment does not have such traffic concentration points where the IHS can collect audit data (logging history) for the entire network. Therefore, at any one time, the only available audit trace will be limited to communication activities taking place within a part of the network accessible by wireless devices [Zhan2000]. Thus, the intrusion identification algorithms must be made to work on this partial and localised information. Standard tools, which are monitoring wired networks and ensuring their security, examine only network layer or higher layers based on the assumption that the lower layers are protected by the physical security of the wires which is not true in WLANs. Ideally, an IHS for WLAN should function at the data link layer or even physical layer if extremely high security is required [Mukh1994].

Issues related to routing are discussed in [Zhan2001, Lim2003] show integrity validation using redundant information from adjacent nodes such as those being used in secure routing also relies on the trustworthiness of other nodes. This could likewise be a weak link for sophisticated attacks if the adjacent nodes have already been compromised. Also, misuse and anomaly model cannot be clearly separated because a node that sends out false routing information could be the one that has been compromised or temporarily out of sync due to volatile physical movement.

Two of the most important issues related to WLANs are their energy constraints and their potential for developing faults. Thus issues related to routing in WLANs are fault-tolerance and energy-efficient permute routing protocols [Amit2005]. A station usually powered by battery, which cannot be recharged while on a mission. Hence any protocol designed for a WLAN should be energy efficient. Also it is possible that some stations may not work perfectly and therefore this protocol should also be fault-tolerant for permutation routing.

Issues of efficient use of resources such as utilisation of wireless link capacity are discussed in [Mukh1994]. Due to relatively high utilisation of wireless link capacity for the IHS internal activity as compared to wired networks, an IHS in WLANs must be extremely efficient in order to limit the management overheads. Also, IHS by itself requires dedicated host and can be prone to the network attack [Kach2003].

Special modulation [Tech2001, Tech2006] techniques can also be useful for identification purposes on the physical layer because they may help to distinguish between "friendly" and "unfriendly" signals (e.g. ignore "unfriendly" signals as noise in the same manner as FM radio receiver ignores AM broadcasts).

To summarise the issues, the general limitations and problems with the existing IHSs based on monolithic architectures are [Bala1998]:

- The central analyser is a single point of failure.
- Scalability is limited.
- It is difficult to reconfigure or add capabilities to the IHS.
- Analysis of network data can be flawed.

Finally, distribution of resources and processing may help to cope with a workload and such a trend was observed in the IHSs where system architecture changed to hierarchical. In such architecture data collection is distributed but data processing and analysis are done centrally. Nearly all contemporary commercial IHS systems follow a hierarchical architecture [Basi2005].

1.5 Conclusion

This chapter has looked at the current security issues in WLANs and recent research activities in this area. It has also looked at intrusion handling process in the current systems and defined the intrusion handling systems by looking at its legacy versions of IDS and IPS systems. The IHS functions and its basic terminology were also explained. The IHS systems were classified based on the audit data and their anomaly, misuse and hybrid identification methods were discussed. Finally this chapter has analysed the recent research activities on intrusion identification by looking at various research papers which led to identifying the current issues with IHSs for WLANs. It has shown the need for IHS reference model and how to design this model, using an ontology modelling approach. Currently there is no standardised reference model which may help to design, compare and evaluate the existing or future IHSs for WLANs. This research is thus focusing on the ontology based approach for defining the IHS reference model.

1.6 Overview of the Rest of the Thesis

Chapter two introduces the requirements and step by step design considerations of the new IHS reference model for *ad hoc* WLANs. It proposes the structure of new IHS reference model and explains in detail the major internal components of this new model by defining the identification and response subsystems and describing the necessary terms and definitions of building block of the new IHS reference model in a host based *ad hoc* structure.

Chapter three is focusing on some most popular IHS systems and their features including identification and response (prevention) subsystems. The new IHS is based on comparison approach of most popular open source legacy IDS/IPS systems and for this reason the research is focusing on development of an Ontology based reference model and not a system itself.

Chapter four starts with introducing Inter IHS Message (IIHSM) Format for the inter IHS communications subsystems of the new IHS reference model. It addresses the need of a UIHRF for translating the non standard intrusion report's formats from different type of IHSs to a UIHRF. This chapter also introduces the requirement for IIHSM which was intended to be independent of the IDP [Wood2007] or IDXP [Fein2007].

Chapter five which is the most extensive chapter focuses on designing of IHS modelling ontology in order to construct, design and develop the new IHS Modelling Ontology (ISHMO). An IHSMO was designed using Hozo ontology editor in order to act as a "reference model" to represent a unified IHS reference model and to standardise the building blocks of existing and new IHS systems.

Chapter six is looking at testing and evaluation of the new IHS reference model and its capabilities by using the IHSMO to evaluate the existing IHS systems. It is shown that an affective IHS system should be structured by following the intensive, comprehensive and effective structure of IHSMO reference model. This approach allows achievement of the most accurate, reliable, effective and efficient IHS system without the compatibility issues in a distributed network.

Chapter seven concludes the thesis and it includes relevant publications by author together with recommendation for future works in this subject area.

2. Analysis and Development of a Framework for IHS Reference Model in WLANs.

This chapter will discuss in detail the structure and components of most common open source IHS systems and focus on the methods of intercommunications of their internal components for the purpose of handling an intrusion.

At this stage, it's vital to investigate the architecture of most common systems and employ a relevant approach to prepare the foundation of the new reference model and to describe its components.

2.1 Analysis of architecture of the most common IHSs in WLANs

2.1.1 Snort Architecture

By examining the source code of Snort from snort.org, the architecture of Snort can be described as follows:

- The snort.c file which is the main Snort code.
- The decode.c file which is the packet decoder.
- The rules.c file which is the rules engine.
- The detect.c file which is the detection engine.
- The log.c file which is the logging engine.

Snort's major components are the packet decoder, the pre-processor, the detection engine, and the alert or logging components. All of these are implemented as plugins in order to increase flexibility.

In Figure 2.1 Snort using Libpcap Packet captures Library developed by LBNL's Network Research Group, which is also used by other IHS systems. Pcap is an application programming interface for packet capturing. The implementation of Pcap for Unix based systems is known as libpcap and the windows version of it is called WinPcap. Libpcap and WinPcap may be used by a program to capture packets travelling over a network and, in newer versions, it also be used to transmit packets on a network at the data link layer, with the capability of collecting a list of network interfaces that can be used with Libpcap or WinPcap. Libpcap and WinPcap are the packet capture and filtering engines of many open source and commercial network tools, including packet sniffers, traffic generators, network testers protocol analysers, network monitors and network intrusion handling systems. There is no packet filtering occurs in Libpcap in Snort architecture. Instead the packet decoder (1) would filter the packets based on their level of risk. The pre-processor[1] is used to take the packet data and process it before the data gets checked against the rules in the detection engine.

The Pre-processor decodes the raw data link layer packets; recognise different protocols, including Ethernet, 802.11, Token Ring, and other higher layer protocols such as IP, TCP, and UDP. The pre-processor arrange or modify data packets before the detection engine does some operation to find out if the packet is being used by an intruder. The pre-processor is very important for IHS to prepare data packets to be analysed against rules in the detection engine (5). Hackers use different techniques to fool an IHS in different ways. Packets are then examined and manipulated by the pre-processor before being handed to the detection engine. The pre-processor checks each packet to see if this packet is something it should look at, alert on, or modify.

The detection engine works by checking the data in each packet against a rule set. The rules set contains of misuse signatures and anomaly signatures are created on demand customisation of new rules.

The alert or logging component takes the output of the data after it has been checked against the rule set. The data can go straight into a log file in a text or binary format. In addition, the data can be stored in a SQL databases, or sent over the network through SNMP traps or through Win pop up messages.

The function of the packet capture and decoder engine is as follows. First, traffic is acquired from the network link via the libpcap library (1). Packets are then passed through to the decoder engine (2), which first fills out the packet structure for the

[1] Preprocessor is a program that processes its input data in order to produce output that in turn is also used as input to another program. The output is said to be a pre-processed form of the input data, which is often used by some subsequent programs i.e. compilers.

link level protocols, which are then further decoded for higher level protocols such as TCP and UDP ports. Then they are sent down to the pre-processor (3).

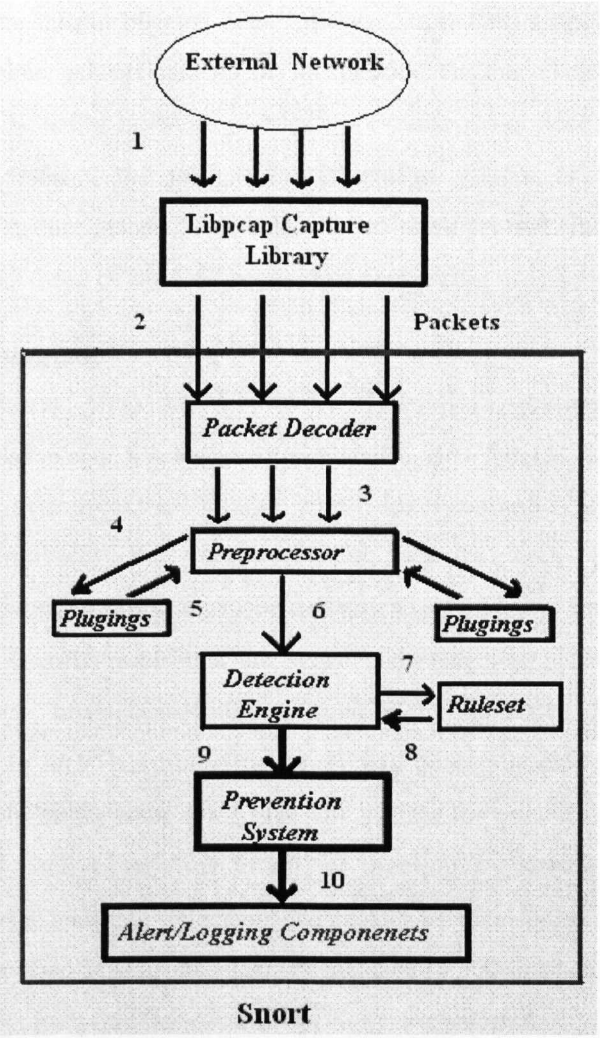


Figure 2.1. Snort Architecture

The function of the pre-processor plugins is as follows. Packets are sent through a set of pre-processors and they are examined and manipulated by the pre-processor's plugins before being handed to the detection engine (5). Each pre-processor determines if the packet is something it should look at, alert on or modify.

Packets are then sent through the detection engine (6). The detection engine checks each packet against the various options listed in the Snort rules files (7 & 8) by performing single, simple tests on an aspect or field of the packet. The detection plugins provide additional detection functions on the packets. Each of the keyword options in the rule is linked to a detection plugin that can perform additional tests. The function of the output plugins is as follows. Snort then outputs the alerts from the detection engine, pre-processors or the decode engine. Detection engine (8), either automatically using layer three based preventions i.e. access control lists or an alert (9) by sending the packet to prevention system or transfer them to administrator for further action (10). It is vital to mention that Snort is detecting at layer two and layer three of the OSI model but preventing only in layer three.

2.1.2 Bro Architecture

Bro is an open-source, Unix-based Network IHS (NIHS) that passively monitors network traffic and looks for suspicious activity. Bro detects intrusions by first parsing network traffic to extract its application-level semantics and then executing event-oriented analysers that compare the activity with patterns deemed troublesome. These rules might describe activities (e.g., certain hosts connecting to certain services), identify activities are worth alarming (e.g., attempts to a given number of different hosts constitutes a "scan"), or signatures describing known attacks or access to known vulnerabilities. If Bro detects something of interest, it can be instructed to either issue a log entry or initiate the execution of an operating system command such as sending email, or creating a router entry to block an address. Its analysis includes detection of specific attacks including those defined by signatures, but also those defined in terms of events and unusual activities such as certain hosts connecting to certain services or patterns of failed connection attempts [LBNL2007].

Bro's architecture is layered into three major components. It uses the libpcap packet capture library (1) and filters the packet stream in the kernel for high performance. This is a built in format of packet filtering, therefore there is no separate packet decoder system as in case with Snort.

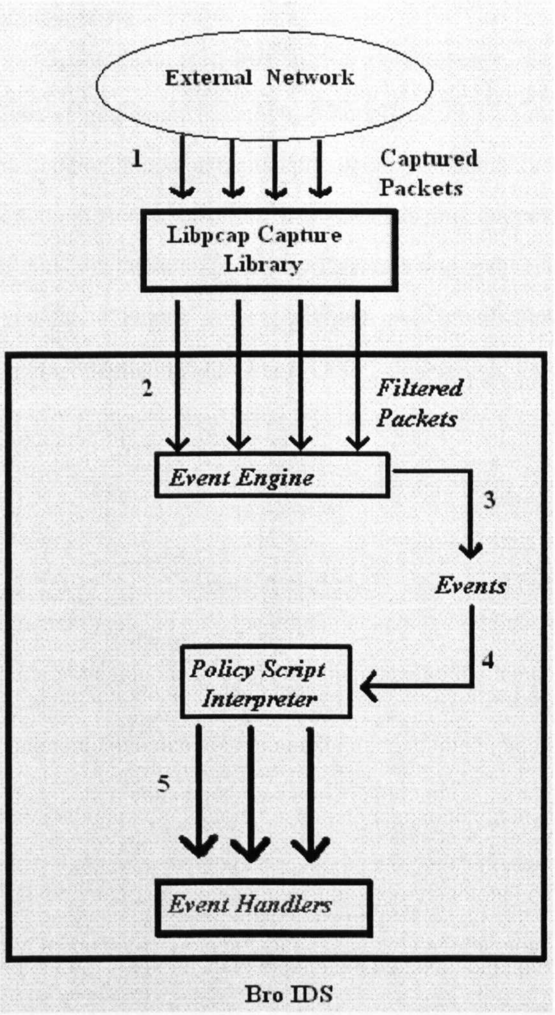


Figure 2.2. Bro IDS Architecture

The filtered packets are then passed along to Bro's event engine (2) or (detection engine for Bro), which reduces the filtered stream into a series of higher level events (3). These events reflect network activity in policy neutral terms. That is, what has happened but not why or whether it is significant.

Finally, Bro's policy script interpreter executes event handlers written in the custom Bro scripting language (5). These scripts express a site's security policy, such as what actions to take when the monitor detects different types of activity. Scripts can generate real time alerts and can also execute arbitrary programs which mean that Bro can respond to attacks. Policy script interpreter is basically a server of both anomaly and misuse signatures. Bro-IDS is only capable of layer three detections.

2.1.3 Kismet Architecture

Kismet is an 802.11 layer two wireless network detector, sniffer, and intrusion detection system. Kismet can sniff 802.11a, 802.11b, 802.11g and 802.11n with any wireless card which supports raw monitoring mode. Kismet is completely passive. It identifies networks by passively collecting packets and detecting standard named networks, detecting hidden networks, and inferring the presence of no beaconing networks via data traffic. Kismet is a signature and trend based Layer two IDS. Kismet runs on Linux, some BSD, and OSX operating environments. It will provide alerts based on fingerprints (specific netstumbler versions, other specific attacks) and trends (unusual probes, excessive disassociation, etc).

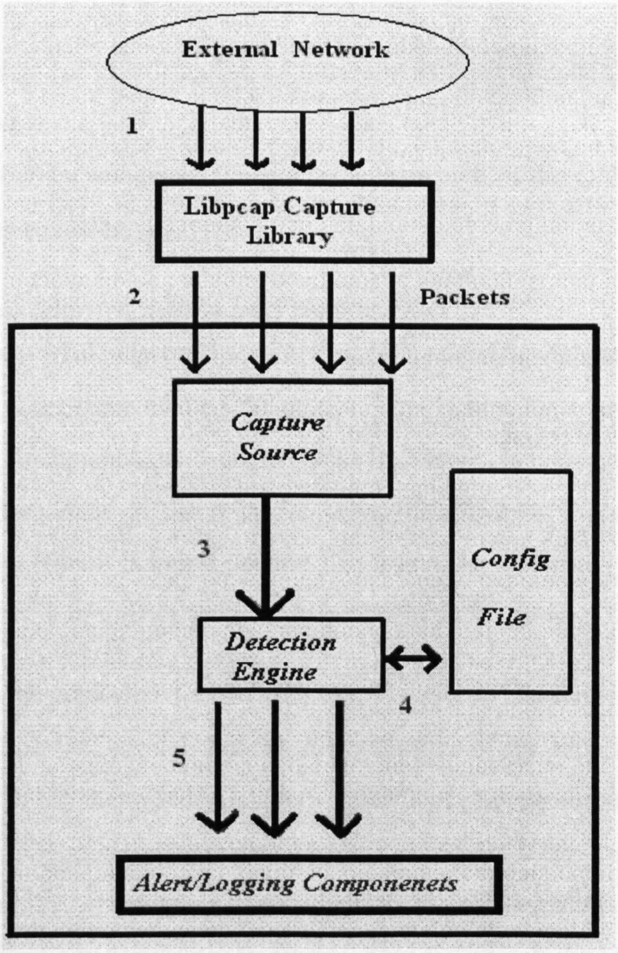


Figure 2.3. Kismet Architecture

Kismet also uses the libpcap packet capture library (with no filtering capabilities) to capture packets from the network (1). It then passes the packets on through the capture source (2). A capture source in Kismet is anything which provides packets to the Kismet engine (3). Capture sources define the underlying engine needed to capture data from the interface, how to change channel, and how to enter rfmon mode. The Kismet engine then matches settings based on the configuration files (4) to sent alerts and logs (5) to admin. Configuration file is a signature server only for issue type of signatures, no anomaly. Kismet only detects layer two attacks.

2.2 Comparison of the Identification and Response Subsystems

Analysis of the Snort, Bro and Kismet architectures has revealed several drawbacks such as:

- Lack of support for identification of intrusion at layer two of the OSI model in Snort and Bro;
- Lack of support for response to intrusion at layer two of the OSI model;
- They all are network based IHSs.

2.2.1 Layer Two Identification and Response

Snort is a very powerful network based IHS which identifies intrusion by matching signatures at the layer three of the OSI model. This is fine for a wired network, but for a wireless environment layer two, the data link layer, is extremely vulnerable to exploits due to weakness of the adjacent layer, the physical layer of OSI model. Similarly Bro IDS, which is based on Snort, is a very powerful network based IHS, however like Snort, it identifies only intrusion at the layer three of the OSI model.

Only Kismet can be considered as an IHS that is capable of identifying intrusion at the layer two of the OSI model. As far as prevention of the identified intrusion is concerned, none of the mentioned IHSs are capable of preventing an intrusion at the layer two of the OSI model. Snort can drop packets after it has been identified an intrusion by matching signatures at layer three of the OSI model. Bro IDS, which is based on Snort, can similarly handle events after the events have been identified and matched at layer three of the OSI model. Although Kismet has the capability of detecting an intrusion by matching intrusion signatures at the layer two of the OSI model, it completely lacks all response features. Table 2.1 based on the recent architectural comparison presents a comparison of the identification and response methods.

Table 2.1. Summary of the identification and response functions

	Snort-Wireless	Bro IDS	Kismet
Identification Features			
Real Time	✓	✓	✓
Syslog alert	✓	✓	
Logs to events log		✓	
Email/page administrator		✓	
Alert to UNIX Socket	✓		
Packet logged to ASCII/binary file	✓		
Win popup alert	✓		
Log to database	✓		
Log to cvs file	✓		
Arp decoding	✓		
Decode layer 2 header	✓		✓
Promiscuous mode	✓		
Test IP header	✓		
Check Payload size	✓		
Match Payload pattern	✓		✓
Test TCP header	✓	✓	
Test ICMP header	✓		
Dump application layer info of a session	✓		
Monitors packet information			✓
Monitor RPC	✓	✓	
Search URI portion of packets	✓		
Detect port scan	✓	✓	
Detect password scans		✓	
Detects vulnerability scans		✓	
Test frame information	✓		✓
rlogin/telnet/ssh analyser		✓	
Monitor HTTP/FTP sessions		✓	
DNS analyser		✓	
Monitor software versions		✓	
Detect unusual probes			✓
Detects excessive disassociation			✓
Prevention Features			
Severe connection	✓		
Block websites/hosts	✓	✓	
Edit router ACL		✓	
General Features			
*NIX based OS	✓	✓	✓
License : Open Source	✓	✓	✓
Network based	✓	✓	✓
Passive monitoring of network traffic	✓	✓	✓
Signature based	✓	✓	✓
Anomaly based		✓	

2.3 The Fundamentals of a New IHS Architecture

In general the above IHSs have the following major issues:

1. They do not have a consistent structure as a result of being various vendors' products.
2. They do not interoperate.
3. Due to sophisticated format of packet processing and handling in layer three (i.e. Snort) and therefore too much CPU overhead and reducing battery life some of the IHS systems are not practically suitable to use in WLANs. This could be counted towards their inefficiency issue in WLANs.
4. Layer three is not secure due to layer two higher vulnerabilities and its direct link to layer three which let malicious packets pass through layer two directly to its neighbour, layer three.
5. Not detecting and preventing attacks for both misuse and anomaly identification before reaches other layers.

Therefore the ideal IHS design should be efficient in term of resources and should protect layer two before attack reaches layer three of OSI model. The proposed architecture in Figure 2.4 would be as follows:

1. Packet passing to a "Filtering System" at (A).
2. Packets will be filtered and enters the pre-processor (B).
3. Pre-processor uses plugins (C) to process packets in order to prepare the captured packets for analysis.
4. The system protects application layer where it resides but identifies and responses only to layer two attacks which consequently protects other layers of OSI model by preventing attacks reaching those layers. This would reduce unnecessary processing of CPU and therefore saves the battery life of mobile hosts in WLANs.
5. It can be noted that, it is likely that the layer 2 frames mainly originated from internal threats without IP header information which statistics [Aird2006] shows these types of attacks are counted towards 60% of total existing attacks. In this case layer three attacks are assumed mainly external threats which would also be detected at layer two.

6. Identification system should be capable of updating the rule set server every time a new set of rules created, therefore in order to achieve the latest rules, the update of the rule set server should happen in regular intervals. This would enhance precision of the identification system.

   There are three approaches to identification which are based on frame's structure when frames reach identification system (D). They are automatic identification using misuse identification system, automatic or manual identification using anomaly identification system and no identification. Based on Figure 2.4, when a frame reaches identification system (D) one of the following will happen:

   a) If the frame consists of malicious codes (an attack) containing in frame header, trailer or payload of the frame and it matches with (6) the rule set (G), then identification system use misuse identification and will automatically produce an alert by sending an alert message (7) to response system. The response system in turn would send a message (9) to prevention action (I) causing an automatic prevention. A copy of this prevention action (10) would also be send to admin. The response subsystem would also issue an alert (H) for the admin's attention (11). In this scenario, (10) would inform the admin about the incident and (11) would show the action taken.

   b) When it comes to detecting anomaly attacks, the identification system will not be able to find any matching with rule sets. The identification system would perform mirror monitoring of frames and acts like a legacy Intrusion Detection System where abnormal/malicious frames are only stopped either automatically or by admin's response but there are no preventions. In anomaly attacks the identification system would detect unusual/abnormal changes to normal profile of frame's behaviour. These abnormalities of frame's header, trailer and payload are in form of a sudden increase in size or unusual changes in payload's format by malicious codes, etc. Subsequently identification system would send an alert (8) to response system (H) in order to either stop the frame and report to admin or just report to admin for further actions. The identification system would set a new rule for this new anomaly attack, so next time it will use this rule to stop the same type of attack like in the same way as misuse detection. Because IHS system would not repeat the same processes for the same attack, it would save on the battery life of the mobile system which it resides, by less processing of its CPU and memory. Therefore this can be counted towards achievement of system efficiency and energy saving using new IHS model as a reference.

c) If it's a normal frame with no suspicious codes, the identification system would send the frames out the interface to upper layers without any further action.

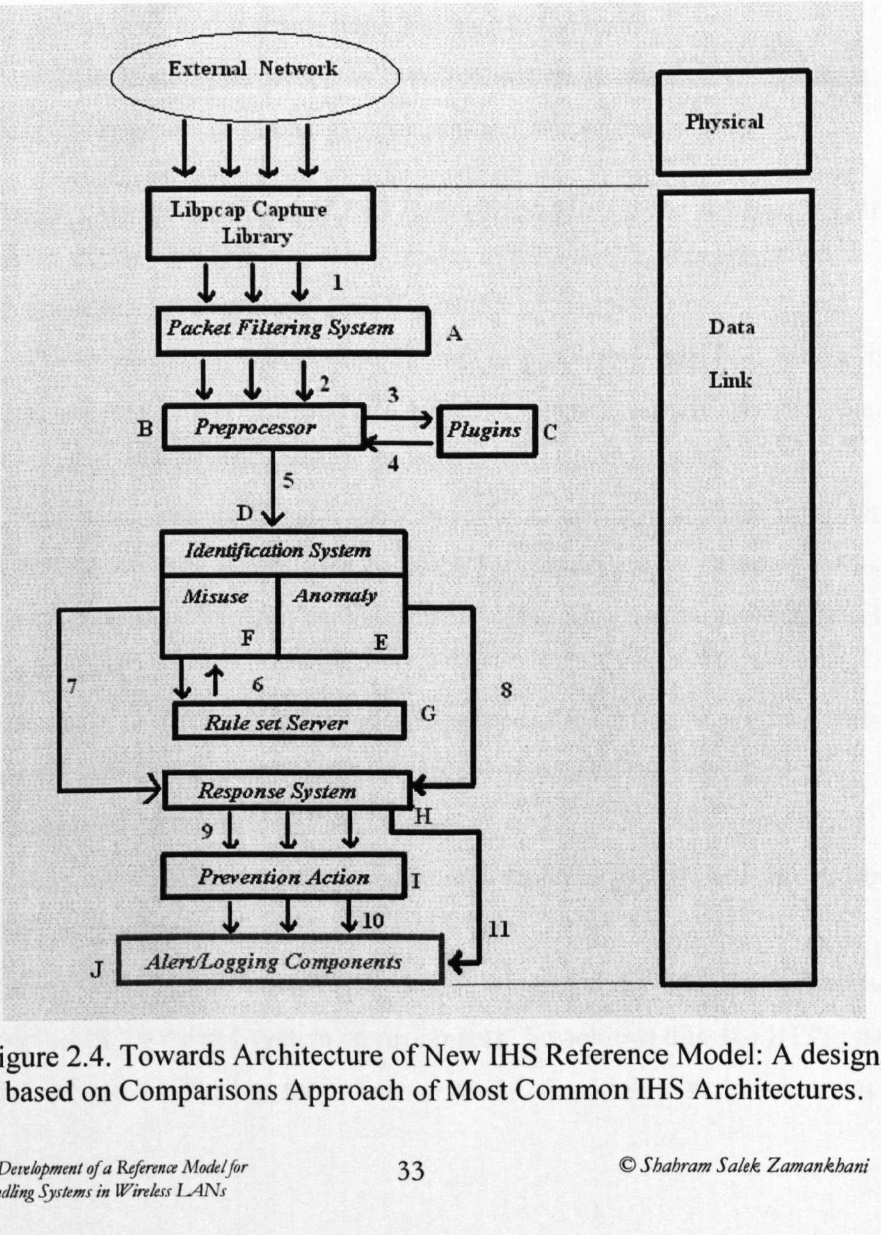


Figure 2.4. Towards Architecture of New IHS Reference Model: A design, based on Comparisons Approach of Most Common IHS Architectures.

The above design is generic and is not dependant on implementation's format of IHS (host, network and hybrid based).

2.4. Architecture of IHS Reference model for *Ad hoc* WLANs

2.4.1 Requirements to IHS Reference Model for Ad hoc WLANs

The open IHS reference model has to support the following requirements:

- Inclusion of both detection and prevention functions;
- Minimising false alarms or even total absence of them;
- Fault-tolerant operation;
- High performance and less overhead on platform which is running;
- Runs continuously and remains non-obtrusive to the user;
- Uses as little system resources as possible which is especially important for wireless environment (e.g. energy consumption, use of bandwidth).

2.4.2 Design Considerations of IHS Reference Model for Ad hoc WLANs

Development of the new IHS reference model in *ad hoc* WLANs are based on the following major design considerations [Yee2003]:

A. Each IHS is operating autonomously and is protecting one host where it is residing and because of the nature of WLAN where there is no any specific point of entry for packets, it must also monitor the traffic between the hosts;

B. IHSs may communicate to each other in order to update and share information about intrusions; this would help to block further access by detected intrusive devices before they become capable of engaging into the operational wireless LAN's system;

C. IHSs residing in different hosts may be produced by different vendors and use different methods for identification of, and responses to the intrusions;

D. System is built on the modular principles;

E. Data source with filtering functionality is a separate system and has not been included in the design.

The proposed ontological based IHS model presented in this thesis has been designed to work for hybrid system environments. To achieve this, the IHS systems which their structure adhere to the reference model should be distributed across the network based on a specific network design. The approach would be a cluster based design. Each cluster in turn, using a client also called an analyser (hop or centre of cluster) and a server also called a manager (spoke or branch of cluster) which in turn would create a star topology for IHS systems. While each wireless host is connected to another wireless host creating a full mesh topology, the intra communications is based on star or "hop and spoke" or "client and server" approach. By distributing and expanding the IHSs, an extended star design would be created. Where all star designed IHS systems working individually on each host and they detect the intrusion and response to it and transparently reports the incident to their local hop which is also a local IHS server. The communication between a local IHS server (hop) with the clients (spokes) member of its cluster happens in a fixed periodic time, for instance every interval of two seconds (can be changed). Then the hop and spokes are sending and receiving keep alive (beacons) messages every interval of two seconds (as explained in "802.11 Topologies" section of appendix A, the beacons periodic update within the *ad hoc* nodes needs to be configured for instance two seconds, a very short delay period). If the local server doesn't receive the keep alive frame (beacons) after for instance four seconds it would assume that the host has been attacked and would shut down its connection and would report to Central Super Server also called the main manager. This would reduce waste of bandwidth and unnecessary usage of resources as beacons are not the actual data. The local IHS server will inform other IHSs in its cluster by updating local managers (hops) IIHSM. The IHS servers in this extended star distributed design would be communicating and correlating events (IIHSM) across the network with other IHS servers using Inter IHS communication Protocol (IIHSCP). The IHS servers can be re-elected if for any reason they go down within the cells. This ensures full time connectivity if the servers their selves are also attacked. The IIHSCP should be compatible with the existing *ad hoc* routing protocols, such as *Ad hoc* On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR), Optimized Link State Routing Protocol (OLSR), and Zone Routing Protocol (ZRP) [Perk2002, John2002, Clau2002, Hass2002]. By correlating WLAN traffic across multiple IHS servers, the model reduces false positives (a recent major issue) while providing more accurate results. All IHS servers are also connected to the Central Super Server and can be manually controlled by the Network administrator.

In this design, any of IHS servers simultaneously act like a network based IHS system for inter IHS communication but the wireless connectivity is still in *ad hoc* mode. This way of establishing distributed star and extended star design would create a suitable hybrid defence system to protect a Mobile *ad hoc* WLAN. Figure 2.5 illustrates this approach, showing an IHS and server design, where there is full mesh connection between *ad hoc* devices. The dotted lines show the Inter IHS Communication between the local server and clients.

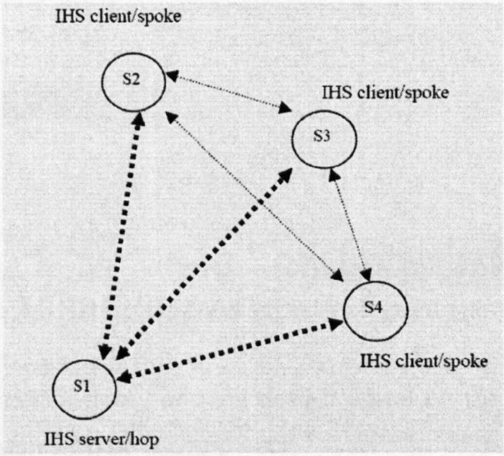


Figure 2.5. IHS client and server design

Figure 2.6 shows the communication between local servers and central super server, monitored by network administrator.

Thus, according to such scenario each IHS system must have three main subsystems:

1) Identification Subsystem;
2) Response Subsystem;
3) Inter IHS Communications Subsystem.

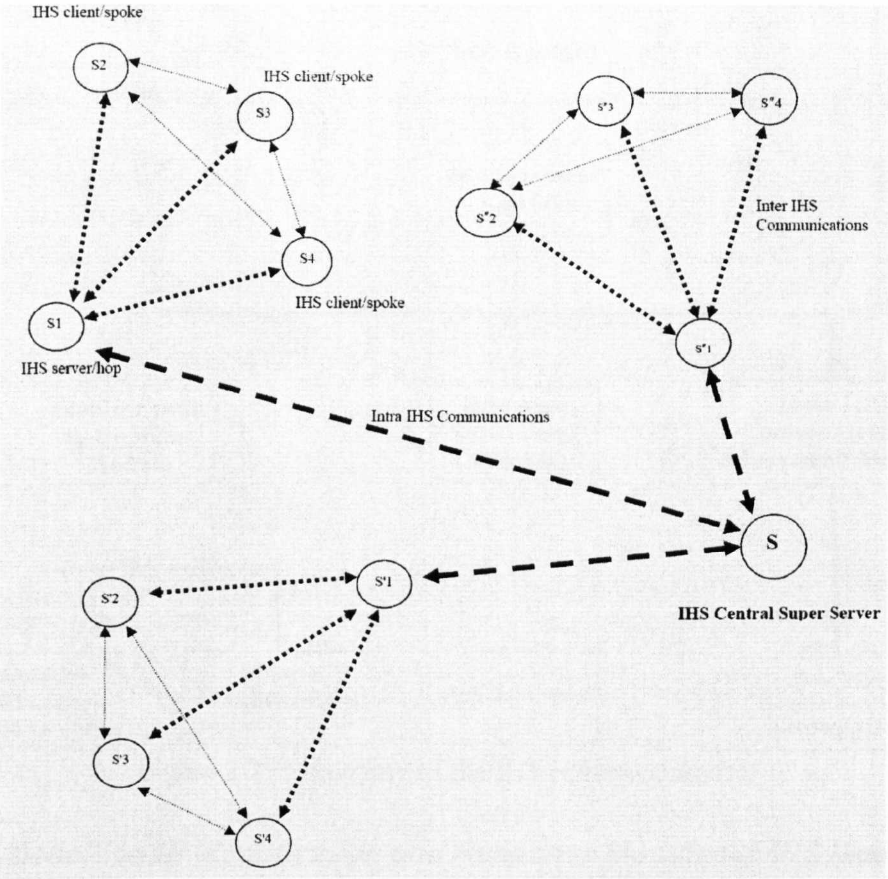


Figure 2.6. IHS client and server extended star design

Figure 2.7 presents the taxonomy of such design based on the above definitions for each modular component of IHS system.

The proposed reference model will be based on above taxonomy. It will show how the IHS system would communicate internally with the three layers of the OSI model which are data link, network and transport layers [Sale2005, Pakš2006, Sale2007, Sale2010a, Sale2010b].

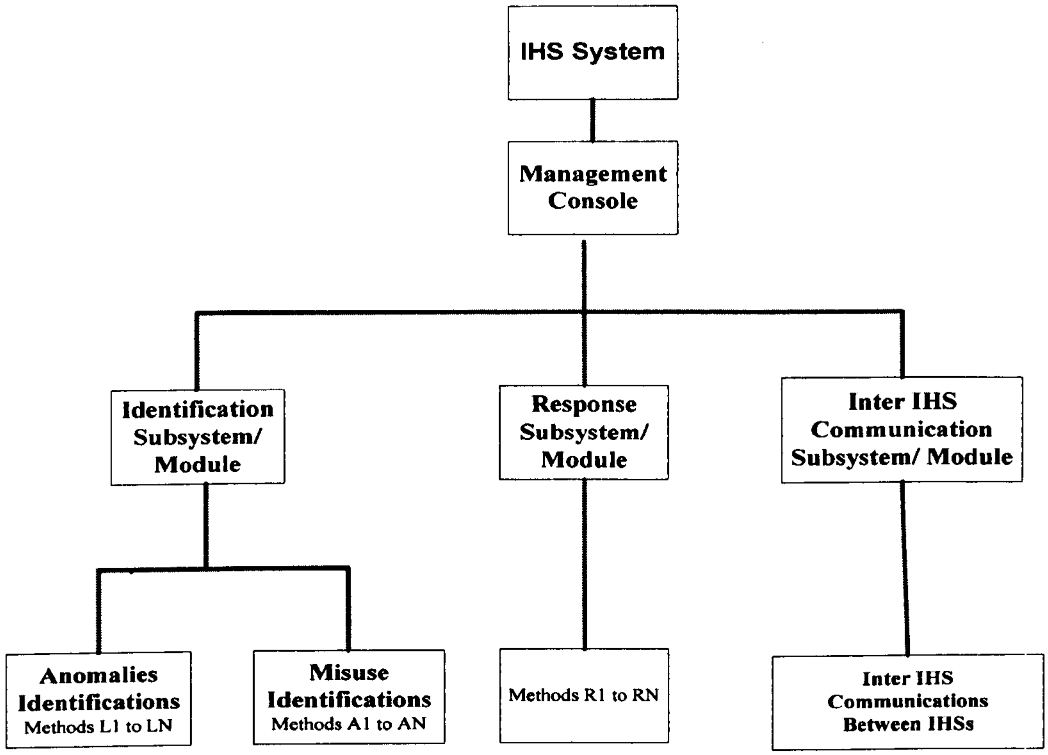

IHS System
Management Console
Identification Subsystem/ Module
Response Subsystem/ Module
Inter IHS Communication Subsystem/ Module
Anomalies Identifications Methods L1 to LN
Misuse Identifications Methods A1 to AN
Methods R1 to RN
Inter IHS Communications Between IHSs

Figure 2.7. Taxonomy of the IHS reference model

2.4.3 Definition of Identification and Response Modules in IHS Model

IHS use identification subsystem to identify only at data link layer but for response and inter IHS communications purposes the IHS should also communicate with transport, network and data link layer [Sale2005, Sale2007].

The identification subsystem is based on the two sets of modules implementing various detection methods (all methods will be discussed later):

- Analysis of "live" packets (Method L1, ... Method LN) for anomalies which must have direct access to the Data link layer;
- Analysis of recorded audit data (Method A1, ... Method AN) for misuse, which must have access to the audit data logs.
- Response Subsystem also has variety of different methods, Method R1, ... Method RN that must have direct access to the data link, network and transport layers.

Here the intrusion identification method is defined (using definition of intrusion from [lim2003]) as an activity for identification of a sequence of related actions by a malicious adversary that results in the occurrence of unauthorised security threats to a target computing or network domain and corresponding intrusion response method as an activity for blocking of the intrusion e.g. by not forwarding malicious packet to the destination application or preventing possible damages e.g. switching off the station as described earlier in Table2.1 . In general the proposed IHS which is built using proposed reference model shall perform two actions at the same time i.e. the IDS and the IPS legacy abilities and functions but with more intelligent features.

Obviously, acceptance of such a general type of reference model by the wider community is an ultimate goal which does not always depends on the quality of the reference model or performance of the implemented prototype but can be affected by the real life considerations such as vendor's competition. According to the anecdotal evidence collected by the authors at the event "Infosecurity Europe 2006" (London, 25-27 April 2006) the current situation in IHS-related market can be characterised as "immature". Many people expressed interest in the authors work but also noted that these ideas may not be used right away. One person (the owner of the small/medium size enterprise) commented: "Speaking technically I very much support this idea but commercially I cannot afford to give results of my 'know-how' away to the competition by simply exporting my knowledge to other vendor's IHS". On the other hand in a seminar hosted by AirDefence on 1st December 2006 in Canary Wharf UK, the AirDefence's chief security and one of the presenters of the seminar also expressed his interest in this project and emphasised that such reference model is vital for evaluation of current and development of future IHSs system to further contribute in enhancing security level of layer 2 and 3 of OSI model in WLANs which have always been vulnerable to intruders. Similar views and comments were also received in the recent event "Infosecurity Europe 2009" (London, 28-30 April 2009).

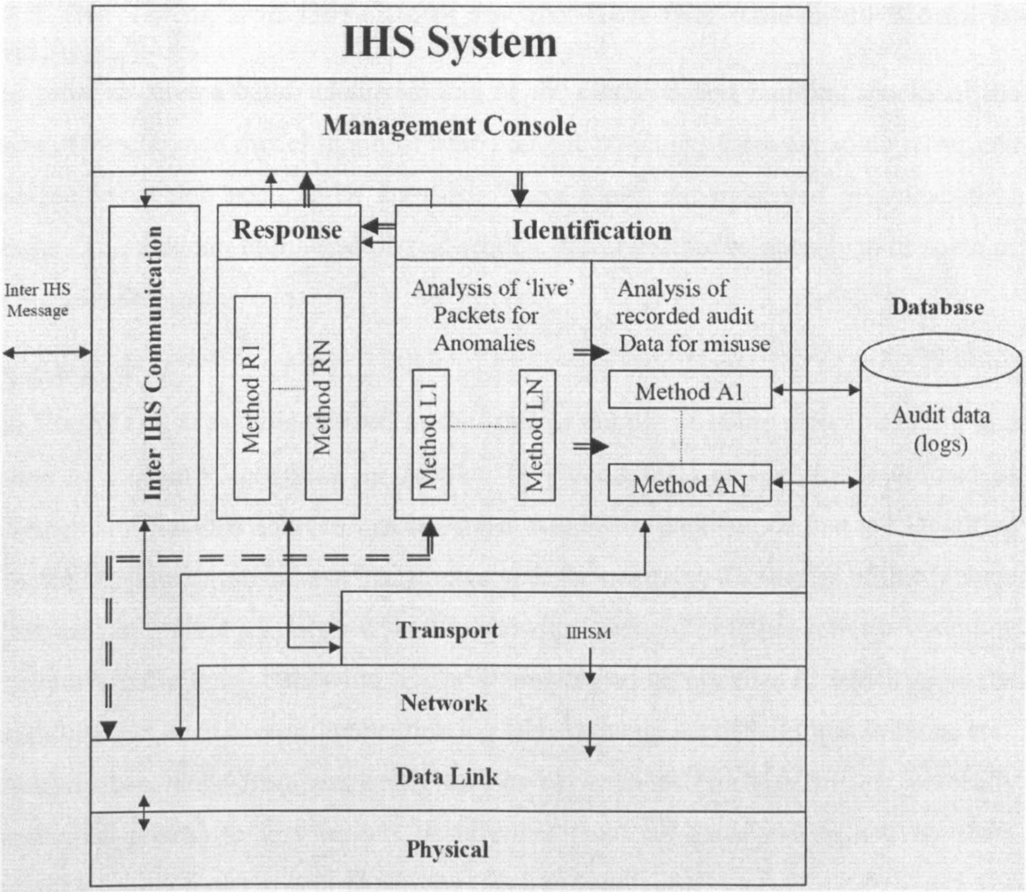


Figure 2.8. Architecture of IHS Reference Model

It thus seems that it is not an appropriate time yet for a new industrial forum for standardisation of IHSs (i.e. situation is similar to the times before X.25/OSI standardisation was started).

The IHS system should also be capable of protecting the most important and crucial type of WLANs implementation known as "host based IHS systems in *ad hoc* usage of WLANs". In order to achieve an IHS model with highest efficiency, integrity and performance, each components of model should be defined and relevant terms and definitions also need to be clarified. The next chapter will look at this importance.

2.5 The Terms and Definitions for the New IHS Reference Model in WLANs

In order to have a better understanding of the elements and building blocks of the new IHS reference model in a host based *ad hoc* structure; there are some terms and definitions which need to be clarified. These terms are presented in alphabetical order. The diagram at the end of this section, illustrates the relationships of some of the terms defined here in.

2.5.1 Activity

In [Refe2007] activity is defined as the state or quality of being active or being in a state of existence, progress, or motion. In [Wood2007] an activity is defined as elements of the data source or occurrences within the data source that are identified by the Detection system as being of interest to the operator. Examples of this include (but are not limited to) network session showing unexpected telnet activity, operating system log file entries showing a user attempting to access files to which he is not authorised to have access, application log files showing persistent login failures, etc . Activity can range from extremely serious occurrences (such as an unequivocally malicious attack) to less serious occurrences (such as unusual user activity that's worth a further look) to neutral activity (such as user login).

In the proposed model here, activity is defined as any suspicious incident which triggers the identification subsystem. This includes but not limited to, malicious frame, unauthorised access, an abnormal behaviour of users, etc.

2.5.2 Administrator

In [Refe2007], an administrator is defined as a person who manages or has a talent for managing. In [Wood2007] an administrator is shown as a human with overall responsibility for devising the security policy of the organisation, and thus, for making decisions about deploying and configuring the IHS. This may or may not be the same person as the operator of the IHS. In some organisations, the administrator is associated with the network or systems administration groups. In other organisations, it's an independent position.

In the proposed model here an administrator is also a person who set ups and configures the network and is responsible for management, security and maintenance. An administrator is also managing operators.

2.5.3 Alert

In [Refe2007], an alert is defined as an attitude of vigilance, readiness or caution before an expected attack. In the proposed model here an Alert is defined as a message from Identification (Anomaly) or Response (Misuse) subsystem to local/central IHS servers that an event of interest has been detected. An alert typically contains information about the unusual activity that was detected, as well as the specifics of the occurrence.

2.5.4 Identification Subsystem

In [Refe2007] the identification is defined the act of identifying by recognising or establishing as being a particular person or thing or discovering or ascertaining the existence, presence, or fact of. In the model proposed in this thesis, the IHS identification module or process that analyses the data for signs of unauthorised or undesired activity or for events that might be of interest to the local/central IHS server or manager.

2.5.5 Response Subsystem

In [Refe2007], a response is defined as an answer or a reply, as in words or in some action. In the proposed model here, the IHS response module or process that listens to identification module's alert in order to automatically prevent the unauthorised activity or attack.

2.5.6 Data Source

In [Refe2007], a data source is defined as a source or collection of pieces of information, as facts, statistics, or codes. In [Wood2007] the data source is defined as raw information that an intrusion identification system uses to detect unauthorised or undesired activity. In the proposed model here, the data source is defined as the raw information collected by the Data Capture library Interface (DCLI) (i.e. Libpcap) that an Intrusion Handling System uses to detect unauthorised or undesired activity. Common data sources include (but are not limited to) raw network packets, operating system audit logs, application audit logs, and system-generated checksum data.

2.5.7 Event

In [Refe2007], an event is defined as something that happens or it is happening. In the proposed model here, the event is defined as the occurrence in the data source that is detected by the IHS system and that may result in an IIHSM alert being transmitted, for example, an attack.

2.5.8 IHS

An intrusion handling system is combinations of one or more of the following components: identification and response subsystems, inter IHS communication subsystems, local/central IHS servers.

2.5.9 Manager

In [Refe2007], a manager is defined as a person who has control or direction of an institution, business, etc or of a part, division, or phase of it. In the proposed model here, manager is defined as one of the IHS components or process from which the administrator/operator manages the various components of the IHS system. Management functions typically include (but are not limited to) system configuration, event notification management, data consolidation, and also reporting.

2.5.10 Notification

In [Refe2007], a notification is defined as an act or instance of notifying, making known, or giving notice. In [Wood2007] notification is defined as a method by which the IHS manager makes the operator aware of the alert occurrence and thus the event. In many IHSs, this is done via the display of a coloured icon on the IHS manager screen, the transmission of an e-mail or pager message, or the transmission of a Simple Network Management Protocol (SNMP) trap, although other notification techniques are also used. In the proposed model here, the notification is an alert which the administrator/operator has received from IHS manager in variety of formats i.e. email, pager messages or transmission of any type of messaging protocols i.e. SNMP.

2.5.11 Operator

In [Refe2007], an operator is defined as a person who operates a machine, apparatus, or the like. In the proposed model here and in [Wood2007] human is defined as the primary user of the IHS manager. The operator often monitors the output of the IHS system and initiates or recommends further action.

2.5.12 Response

In [Refe2007], a response is defined as an answer or reply, as in words or in some action. In the proposed model here, similar approach has been taken to define response. Therefore response is defined as actions that have taken in response to an event. Responses may be undertaken automatically by some entity in the IHS architecture or may be initiated by an operator (human). Sending a notification to the operator is a very common response. Other responses include (but are not limited to) logging the activity; recording the raw data (from the data source) that characterised the event; terminating a network, user, or application session; or altering network or system access controls

2.5.13 Inter IHS Communication Subsystem

Inter IHS communication subsystems is a method by which the IHS system passes IIHSM to other IHS systems using IIHSCP.

2.5.14 Sensor (Network Based)

In [Refe2007], a sensor is defined as a mechanical device sensitive to light, temperature, radiation level, or the like, that transmits a signal to a measuring or control instrument. It is the IHS component that collects data from the data source. The frequency of data collection will vary across IHS offerings. The sensor is set up to forward events to the identification subsystem.

2.5.15 Signature

In [Refe2007], a signature is defined as a person's name or a mark representing it, as signed personally or by deputy, as in subscribing a letter or other document. In the model proposed here, signature is defined as a rule used by identification subsystem to identify interesting activity to the security administrator. One of the mechanisms to detect an intrusion is signatures matching however this is not necessarily the only mechanism by which IHSs detect intrusions.

2.5.16 Security Policy

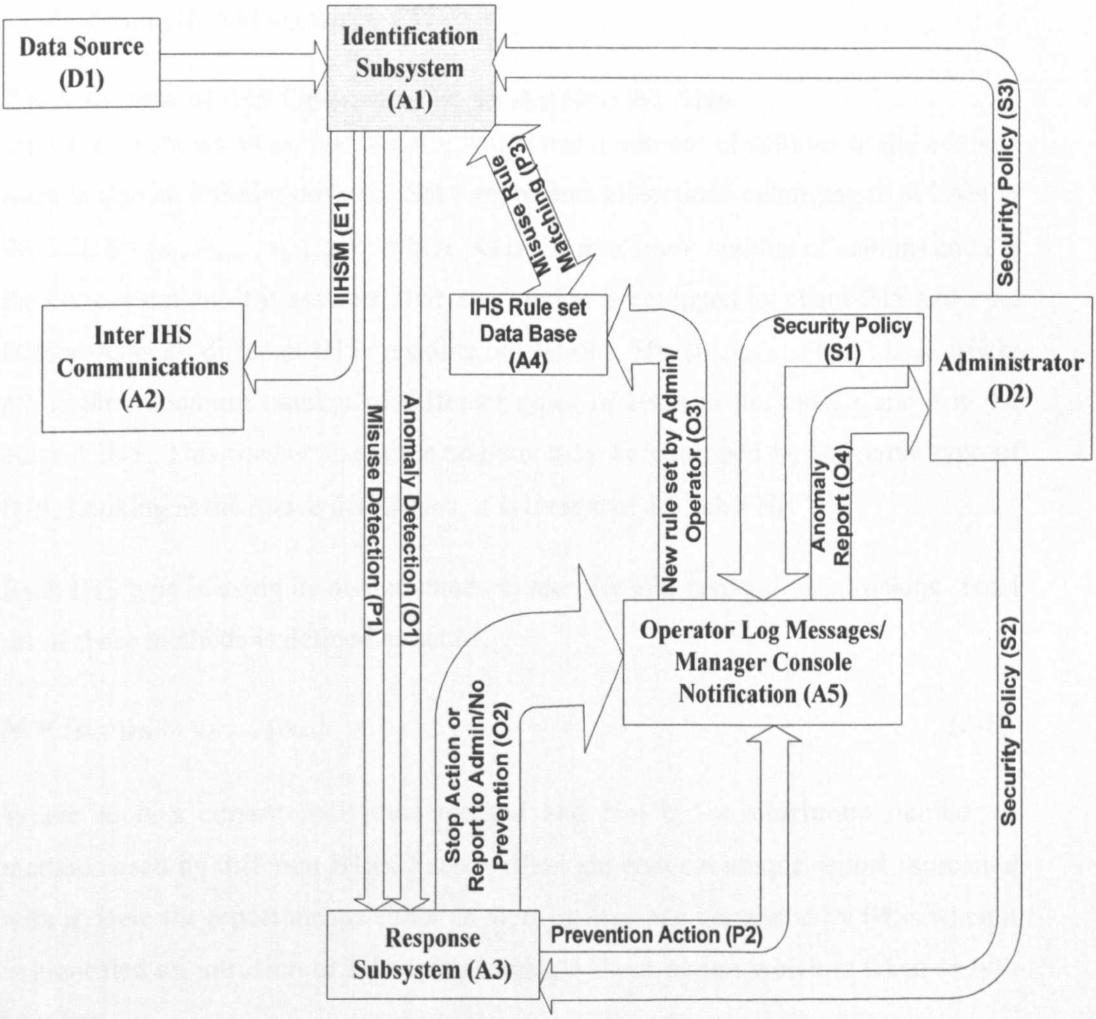
In [Refe2007] security policy is defined as definite course of action adopted for the sake of expediency, facility, etc. In [Wood2007] and in the proposed model here, security policy is defined as a predefined formally documented statement that defines what activities are allowed to take place on an organisation's network or on particular hosts to support the organisation's requirements. This includes, but is not limited to, identification of hosts that are to be denied external network access.

2.5.17 IHS Entities and Relationships

Figure 2.9 illustrates the IHS entities and their relationships. The alphabetic matching of arrows shows relative instances. The main components of IHS are also shown by different letters from A1 to A5. Other components which are also collaborating with IHS main components are shown by letters D. Not every IHS will have all of these separate components exactly as shown. Some IHSs will combine these components into a single module; some will have multiple instances of these modules.

As shown in Figure 2.9, if frame consists of an attack that matches the rule sets, the identification subsystem use misuse detection and would send a message (P1) to response subsystem. Response subsystem (A3) would automatically generate a prevention action and would send a notification message to manager/management console (P2).

When IHS system detects anomaly attacks, the identification subsystem will not be able to find any matching with rule sets. The identification subsystem would perform mirror monitoring of frames and acts like a legacy Intrusion Detection System where abnormal/malicious frames are only stopped either automatically or by admin's response but there are no preventions. In anomaly attacks the identification subsystem would detect unusual changes in frame header or payload in a format of sudden increase in size of payload or changes in payload's format by malicious codes, etc. Then identification subsystem would send a message (O1) to response subsystem in order to either stop the frame and report to admin or just report to admin for further actions. The management console would send this anomaly report (O4) to administrator. Administrator in turn would update the security policy (S1, 2 and 3).



Data Source (D1)
Identification Subsystem (A1)
Security Policy (S3)
Misuse Rule Matching (P3)
IIHSM (E1)
Inter IHS Communications (A2)
IHS Rule set Data Base (A4)
Security Policy (S1)
Administrator (D2)
Misuse Detection (P1)
Anomaly Detection (O1)
New rule set by Admin/ Operator (O3)
Anomaly Report (O4)
Operator Log Messages/ Manager Console Notification (A5)
Security Policy (S2)
Stop Action or Report to Admin/No Prevention (O2)
Response Subsystem (A3)
Prevention Action (P2)

Figure 2.9. IHS entities and relationships

Security policy originally created by head of security of the organisation and passed to administrator by creating a new rule set which would be send to the rule set database using S3 via A1 so then the similar type of attack could be prevented through response subsystems like a new misuse detection. As explained earlier this would count towards achieving most efficient IHS system by saving on battery life of mobile host. If it's a normal frame with no suspicious codes, detection engine would send the packet out the interface without any changes. The identification subsystem would also communicate with other IHSs through inter IHS communication module (A2) in order to update their security database and inform other IHSs about the incident using IIHSM shown by E1.

2.6 Analysis of IHS Operation in an *Ad Hoc* WLANs

Figure 2.10 shows an *ad hoc* WLAN which has a number of stations in the cell and there is also an intruder presents. Set S represents all stations belonging to WLAN in the cell, S= $\{s_1, s_2,\ldots, s_i,\ldots,s_{Ns}\}$ where Ns is the maximum number of stations and *i* is the current station. It is assumed that each station is equipped by client IHS and a set H represents all different IHSs residing on stations, H= $\{h_1, h_2,\ldots, h_k,\ldots,h_{Nh}\}$ where Nh is the maximum number of different types of IHSs in the cluster and *k* is the current IHS. This means that some stations may be equipped by the same type of IHS. Looking at the above definitions, it is clear that $1 \leq Nh \leq Ns$.

Each IHS type is using its own methods to identify and respond to intrusions. Total set of these methods is defined as set M,

$$M = \{\mu_1, \mu_2, \ldots \mu_j, \ldots, \mu_{Nm}\} \qquad (2\text{-}1)$$

Where μ_j is a current individual method and Nm is the maximum number of methods used by different IHSs. Each method generates a unique report associated with it. Here the report means either an alerting message generated by IHSs when it is identified an intrusion or informing message about action which is taken or will be taken.

The same method μ_j may have different instantiations denoted as $m_{j,k}$ of h_k in different IHSs and as a result, it produces semantically equal but differently structured reports denoted as $r_{j,k}$. Thus we have the sets of all reports R_k for all individual IHSs h_k

$$R_k = \{r_{1,k}, r_{2,k},\ldots, r_{j,k},\ldots r_{Nr,k}\} \qquad (2\text{-}2)$$

where $r_{j,k}$ is a current report format for IHS h_k and N_r is the maximum number of report instantiations.

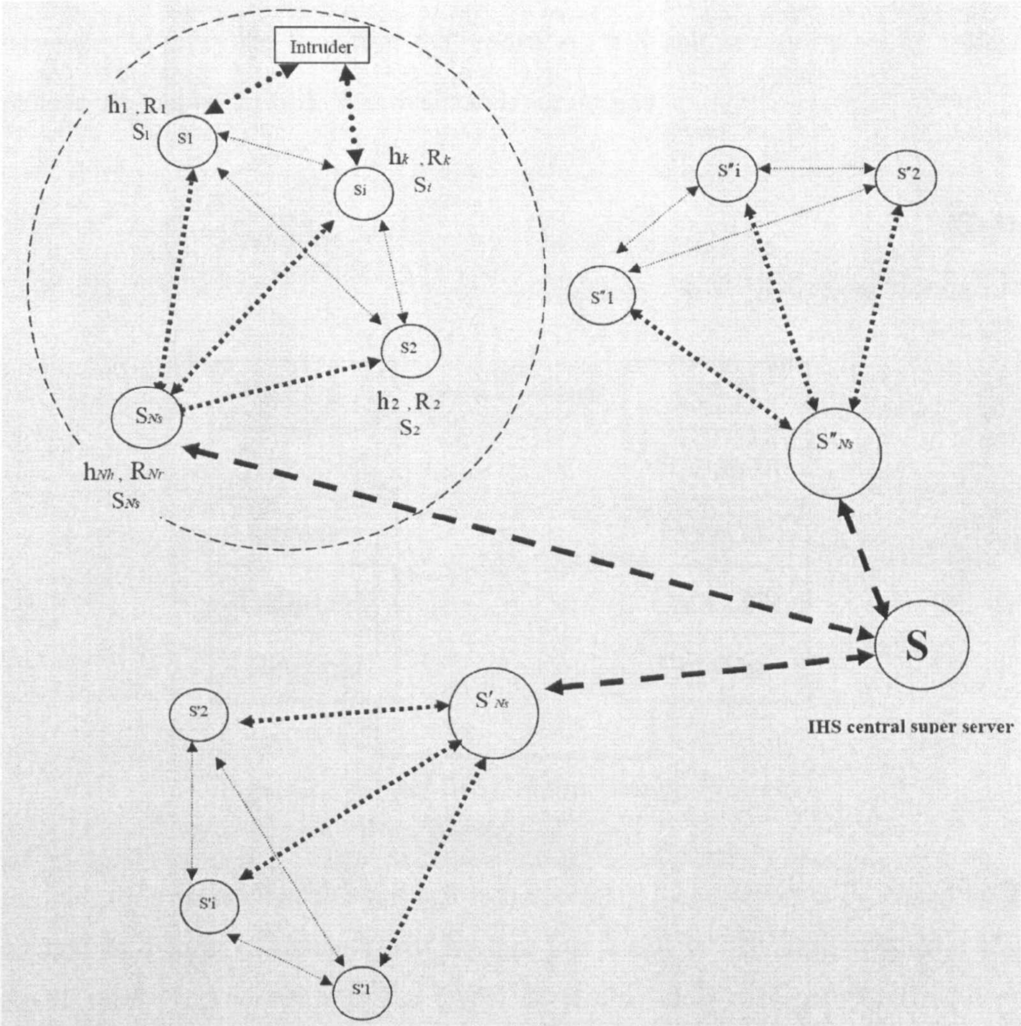


Figure 2.10. A model of *ad hoc* WLAN with present intruder

A total set $\Re$ contains all possible reports in all h_k and for all methods M:

$$\Re = \{R_1, R_2, ..., R_k, ...R_{Nr}\} = \{R_k\}_{k=1}^{Nr} \qquad (2\text{-}3)$$

Not all methods μ_j will be presented in all h_k and consequently different R_k may have different cardinalities $\# R_k$.

Thus we have a triples set of each stations with different reports for different IHSs showing as $<s_i, h_k, R_k>$, for each station s_i where $s_i \in S, h_k \in H, R_k \in \Re$.

Obviously, in order to exchange information about intrusions, it is necessary to define Inter IHS Communication Protocol (IIHSCP), (Figure 2.11). Also, it is necessary to unify the format of the reports $r_{j,k} \in \Re$ and as result a set of unified reports $\Re_j^u$ can be formed. Where u stands for unified.

$$\Re_j^u = \left\{ r_1^u, \ldots, r_j^u, \ldots, r_{N\mu}^u \right\} \tag{2-4}$$

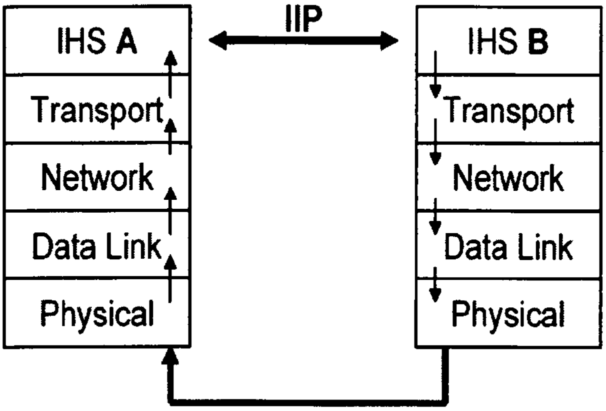


Figure 2.11. Inter IHS Communication Protocol

In order to implement such approach it is necessary that a report is produced by a method in a form of external text but not just a call to operating system or the Application Programme Interface (API). It is also clear that such text requires defining a structure which includes format, language, terminology, etc. Thus, based on $\Re_i^u$ a Unified Intrusion Handling Report Format (UIHRF) should be defined for the purpose of inter IHS communications.

In this distributed system all IHSs residing on different stations will be able to converge to the latest data in order to fulfil their function and enhance their security level more efficiently; this would be accomplished through client and server communications. This way each station individually would save a considerable amount of link capacity and efficient utilisation of CPU and battery life which are important issues for WLANs [Lim2003].

2.7 Introduction to ontology and its use in the engineering context

The ultimate task of this research is to consider an ontology modelling language to design and evaluate the new IHS reference model. Ontology is a term in philosophy and its meaning is a “Theory of existence”. Ontology is a model of some *domain of knowledge.* Domain ontology is an explicit list and organisation of all the terms, *relations* and *objects* that constitutes the *presentational scheme* for that domain [Genn1995].

2.7.1 Importance of ontology and its definition

In recent years the development of ontologies (explicit formal specification of the terms in the domain and relations among them [Grub1993]) has been moving from the realm of the Artificial-Intelligence laboratories to the desktops of domain experts. Many disciplines now develop standardised ontologies that domain experts can use to share and annotate information in their fields [Noy2001].

Some of the main reasons of ontology development are summarised in [Noy2001]:

- To share common understanding of the structure of information among people or software agents;
- To enable reuse of domain knowledge;
- To make domain assumptions explicit;
- To separate domain knowledge from the operational knowledge;
- To analyse domain knowledge.

Sharing common understanding of the structure of information among people or software agents is one of the more common goals in developing ontologies [Muse1992, Grub1993].

The construction of ontologies is difficult, time-consuming and expensive process [Grun2002]. It requires certain skills because it is still an art rather than a technology. Ontology involves the specification of concepts and relationships that exist in the domain, besides their definitions, constraints and properties [Mian2004, Mian2003]. For instance, in the mechanical design setting, previous designs are often used as a reference for design of new products [Mizo1997]. One of the critical issues in such cases is how to understand the intentions and justifications of various decisions made in them by different designers. They are collectively called a Design Rationale (DR). DR information is often implicit and the implicitness often causes difficulties in reusing the designs. Thus, DR is as important as design drawings. Ontology plays a role similar to DR in reusing knowledge bases. In order to reuse knowledge in a knowledge base (KB), underlying conceptualisation must be known which reflects the assumptions and requirements made in the problem solving using the KB. Although many KBs have been built to date, no such information has been described. Ontologies as DR information of KBs will contribute to reuse of KBs and play the roles of backbones of KBs. The future KBs should be built with explicit representation of ontologies.

In the current practices in industry, engineers have suffered from the difficulty of reusing technical documents of such functional knowledge, since the documents tend to be written in *ad hoc* manner using each engineer's vocabulary and are specific to products or domains. For resolving these difficulties, an ontological framework of functional knowledge was developed which includes an ontology of device and function as conceptual viewpoint and a functional concept ontology as a controlled vocabulary [Kita2005]. These ontologies play a role as guidelines or constraints to avoid *ad hoc* modelling. This framework was successfully deployed in a manufacturing company in Japan in daily activities such as design review, equipment improvement and patent application.

As discussed in [Mizo1997] there are eight levels of usage of ontology:

Level 1: Used as a common vocabulary for communication among distributed agents.
Level 2: Used as a conceptual schema of a relational data base. Structural information of concepts and relations among them is used. Data retrieval from a data base is easily done when there is an agreement on its conceptual schema.
Level 3: Used as backbone information for a user of a certain KB. Levels higher than this play roles of the ontology which has something to do with "content".
Level 4: Used for answering competence questions.

Level 5: Standardisation

5.1 Standardisation of terminology (at the same level of Level 1)

5.2 Standardisation of meaning of concepts

5.3 Standardisation of components of target objects (domain ontology).

5.4 Standardisation of components of tasks (task ontology)

Level 6: Used for transformation of data bases considering the differences of the meaning of conceptual schema. This requires not only structural transformation but also semantic transformation.

Level 7: Used for reusing knowledge of a KB using DR information.

Level 8: Used for reorganizing a KB on DR information.

Ontology has been defined in several ways by different researchers and various ontologists are found in the field of philosophy as well as in the area of computer sciences.

1. Ontology defines a logical common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them. Thus, ontologies are "a way of specifying the structure of domain knowledge in a formal logic, designed for machine processing" [Noy2001].

2. Ontology is explicit specification of the conceptualisation that inevitably continues to evolve over time due to domain changes, adaptations to different applications, changes to the conceptualisation or understanding of a domain, etc. Ontology evolves over time, when they are altered to correct time, to accommodate new information, or to adjust the representation of the domain as the world changes [Grub1993].

3. A definition of ontology in Artificial Intelligent (AI) community by Tom Gruber and The Knowledge Systems Laboratory at Stanford University [Grub1993] also indicates the same by this explanation: "Ontology is an explicit specification of a conceptualisation. The term is borrowed from philosophy,

where ontology is a systematic account of existence. For AI systems, what 'exist' is that which can be represented".

4. A definition of ontology in KB community is "a theory of primitive vocabulary/concepts used for building artificial systems" [Mizo1993].

5. Ontology is a term in philosophy and its meaning is a "theory of existence" [Mizo1997, Mizo2005]. Ontology consists of *concepts* and *relations* and their *definitions, properties* and *constraints* expressed as *axioms*. Ontology is simply a model of some *domain of knowledge*. A domain ontology is an explicit list and organisation of all the terms, *relations* and *objects* that constitutes the *presentational scheme* for that domain [Genn1995]. Ontology may take a variety of forms, but necessarily it will include a vocabulary of unambiguous term and some specification of their meaning.

The ultimate purpose of ontology engineering is defined as follows [Mizo1997]: *"To provide a basis of building models of all things in which computer science is interested"*. The definition number five above will be used as working definition through the rest of the thesis.

2.7.2 Rules in ontology design

It can be noticed that the ontology building methodology are not mature enough however there are some methodologies for the design and evaluation on the ontology [Usch1995, Lin1996, Noy2001, Kita2005].

The quality of the ontology can only be assessed in the application in which it has been designed for. There are some fundamental rules in ontology design. These rules may seem rather dogmatic but in many cases they can help to make design decisions. The rules are formulated as:

1) There is no one correct way to model a domain. There are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.

2) Ontology development is necessarily an iterative process.

3) Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain [Noy2001].

2.7.3 Ontology Terminology

Classes represent *concepts* in the domain and not the words that denote these concepts. It should be reminded that Ontology is a model of reality of the world and the concepts in the ontology must reflect this reality. Classes are the focus of most ontologies. The name of a class may change if we choose a different terminology, but the term itself represents the objective reality in the world. For example, a class of *Shrimps* can also be renamed as *Prawns* however the class still represents the same concept [Noy2001]. *Slots* are the properties of each concept describing various features and attributes of the concepts. It is also called a *role* or *property*. *Terminology* is a theory of the labels of *concepts*. The labels of concepts are named after coming to an arrangement on them which involves a process of discussion in the certain *community*.

A *Taxonomy* is a hierarchy of ontological concepts which defines relationship between concepts with the help of ontology links such as an "is-a" or "part-of" link. The synonym for ontology would be model (of something in data), and the synonym for taxonomy would be tree. The taxonomy of a tiger is that it is a subtype of cat (classification), but an ontological description may be that the tiger has a relationship to Asia, the continent on which it lives. Taxonomy is Ontology in the form of a hierarchy. In other words Taxonomies are things of interest arranged in a hierarchical structure, typically in a supertype/subtype relationship [Hobe2008].

A *vocabulary* is a set of words where each word indicates some concepts. Vocabulary is language dependent [Rahm2006b, Sale2010a, Sale2010b].

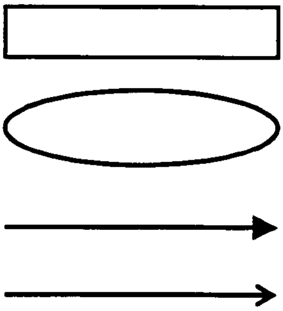
An *axiom* is a declaratively and rigorously represented knowledge which has to be accepted without proof. In predicate logic case, a formal inference engine is implicitly assumed to exist. Axioms have two roles in ontology description, rule 1 is to represent the meaning of concepts rigorously, and rule 2 is within the scope of the knowledge represented declaratively, to answer the questions on the capability of the ontology and things built using the concepts in the ontology [Mizo1997].

Finally a formal ontology is axiomatic description of an ontology. It can answer questions about the capability of ontology. Ontology is an explicit and less ambiguous description of concepts and relations among them appearing in the target thing. Such ontologies exist as many as the possible target things. There is no need to use logic to describe it. Formally ontology consists of terms, their definitions and axiom relating to them; terms are typically organised in taxonomy [Mizo1997, Sale2010a, Sale2010b].

2.7.4 Symbols and Axioms

A formal definition of the ontology requires certain instruments such as symbols including links to slots and concepts, etc as well as axioms. The following symbols and their meanings are used for the definitions of the ontology construction.

- Concepts/Class
- Slot/Attribute
- Link to (Sub) Concepts/Class
- Link to Slot/Attribute



Here the concept/class is represented by a rectangle and the slot/attribute is shown by ellipse/oval. It should be noticed that the links between concepts and between slots are represented by two different types of arrows indicating either *part-of* or *is-a* relationship.

Table 2.2 is showing symbols [Lin1996, Mian2004] which will be used for defining the axioms in this project.

Table 2.2. Symbols used to represent the axioms used for IHSMO

Symbol	Meaning
$\forall$	For all
$\exists$	There exist
$\neg$	Not
$\wedge$	And
$\vee$	Or
$\rightarrow$	Implication
$\leftrightarrow$	Equivalent
$\subseteq$	Belong to
$\cup$	Union

2.7.5 Axioms

Definitions of the axioms are based on [Lin1996, Mian2004, Rahm2006b].

A *part* is a *component* of the *artifact* being designed. The artifact itself is also viewed as a part. The concept of 'part' introduced here represents the physical identity of the artifact, software components and services. The structure of a part is defined in terms of the hierarchy of its components parts. The relationship between a part and its components is captured by the *predicate partOf*. Between two parts x and y, *partOf* (x, y) means that x is a part/components (subpart) of y.

The following two axioms state that a part cannot be a component of itself and it is never the case that a part is a component of another part which in turn is a component of the first part. This shows that the relation *partOf* is non-reflexive and anti-symmetric:

$(\forall x)\ \neg partOf(x,x)$ (A1)

$(\forall x,y)\ partOf(y,x) \rightarrow \neg\ partOf(x,y)$ (A2)

The relation *partOf* is transitive; that is, if a component of another part that is a component of a third part, then the first part is also a component of the third part.

$(\forall x,y,z)\ partOf(z,y) \wedge partOf(y,x) \rightarrow partOf(z,x)$ (A3)

A part can be a (sub) component of another part. But since each part has a unique ID (its name), it cannot be sub-component of two of more distinct parts that are not components of each other.

$$(\forall x,y,z)\ partOf(x,y) \wedge partOf(x,z) \rightarrow x \vee partOf(y,z) \vee partOf(z,y) \qquad \text{(A4)}$$

Parts are classified into two types depending upon the *partOf* relationship it has with the other parts in the hierarchy. The two types are primitive and composite.

- A primitive part is a part that can not be further subdivided into components. These types of parts exist at the lowest level of the artifact decomposition hierarchy. Therefore, a primitive part cannot have sub-parts.

  $$(\forall x)\ primitive(x) \rightarrow (\neg\exists y)\ partOf(y,x) \qquad \text{(A5)}$$

  Primitive parts serve as a connection between the design stage and the manufacturing stage.
- A composite part is a composition of one or more parts. A composition part cannot be a leaf node on the part hierarchy; thus, any part that is composite is not primitive.

  $$(\forall x)\ composite(x) \rightarrow \neg primitive(x) \qquad \text{(A6)}$$

More composite parts are assemblies that are composed of at least two or more parts.

$$(\forall x)\ assembly(x) \leftrightarrow (\exists y,z)\ partOf(y,x) \wedge partOf(z,x) \wedge y \neq z \qquad \text{(A7)}$$

Sometimes a designer may need to find out the direct component of a part. A part is a direct component of another part if there is no middle part between the two in the product hierarchy.

$$(\forall y,z)\ direct\_partOf(y,z) \leftrightarrow partOf(y,z) \wedge (\neg\exists x)\ partOf(y,x) \wedge partOf(x,z) \qquad \text{(A8)}$$

Where y is a direct part of z if y is a component of z and there is no x such that y is a part of x and x is a part of z.

If y is a part x then x is the whole of y

$$(\forall x,y)\ partOf(y,x) \leftrightarrow wholeOf(x,y) \qquad \text{(A9)}$$

Classes are disjoint if they cannot have any instances in common:

$$(\forall x,y)\ disjoint\ (x,y) \rightarrow (\neg\exists\, z)\ partOf(z,x) \wedge partOf(z,y) \qquad \text{(A10)}$$

2.7.6 Recommended steps in building ontology

The construction and development of the ontology includes [Noy2001]:

- Defining classes/concepts in the ontology;
- Arranging the classes/concepts in a taxonomic (subclass–super class) hierarchy;
- Defining slots (will be discussed in next section) and describing allowed values (cardinality) for these slots;
- Filling in the values for slots for instances.

It has also been suggested [Noy2001] that, to start the development of an ontology by defining its domain and scope. That is, answer several basic questions:

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?

The answers to these questions may change during the ontology-design process, but at any given time they help limit the scope of the model.

There are also several approaches in developing a class hierarchy in ontology construction which is suggested in [Usch1996]:

- A top-down development process starts with the definition of the most general concepts in the domain and subsequent specialisation of the concepts.
- A bottom-up development process starts with the definition of the most specific classes, the leaves of the hierarchy, with subsequent grouping of these classes in to more general concepts.

- A combination development process is a combination of the top-down and bottom-up approaches.

2.8 Conclusion

An in depth comparison is accomplished on the architecture of the most common existing open source IHS systems. None of the systems are compatible with each other and each employs different methods for intrusion identification and response. None of the systems could deal with malicious attacks in layer 2. Therefore, the proposed architecture is designed to protect layer 2 of the OSI model thus bringing more efficiency, reliability and higher security to WLANs.

The proposed format of client and server structure of IHS for distribution of the host based *ad hoc* networks is also proven to be vital in order to achieve more reliable, secure Wireless *ad hoc* networks. This approach would save a considerable amount of link capacity and efficient utilisation of CPU and batteries, which are important issues for WLANs and they are addressed in [Toh2001, Amit2005, Bono2007].

The ultimate task of this research is to consider a modelling language to design and evaluate the new and existing IHS reference model. Ontology can provide adequate guide and suggestion on building models. Therefore, an introduction to ontology and its use in the engineering context has been introduced. Rules in ontology design, terminology and steps on building ontology are also discussed.

The next chapter looks at the detailed comparison of the features of the real IHSs using case study approach.

3. A Detail Comparison of the Features of Real IHSs using Case Study Approach

There are many commercial IHSs for wired networks, among which Snort is an advanced open source and Linux based IHS system. However there are only a few Wireless IHSs in Wireless Networks, some are open source and others are proprietary/commercial. The followings are the list of the current most popular IHS systems for WLANs:

- Snort Wireless,
- Kismet,
- Bro-IDS
- Proventia Desktop Security (Windows),
- Airdefence

3.1 Snort Wireless

Snort-Wireless adds several new features of 802.11 IHS functionalities to the standard Snort's distribution [Lock2003]. These features allow one to specify custom rules for detecting specific 802.11 frames, rogue access points, *ad hoc* networks, and Netstumbler [Stum0000] like behaviour in the vicinity of the Snort-Wireless sensor. In order to accomplish this, Snort's rule engine has been augmented with support for a new "wifi" protocol. The remaining features are implemented as pre-processors that can be configured and tuned as desired according to the site of deployment.

The following is intended to be a guide for writing rules using the new "wifi" protocol and its accompanying plugins, as well as the pre-processors used for rogue network and netstumbler detection [Lock2003].

3.1.1 802.11 Detection Rules

Writing custom rules for detecting 802.11 frames matching your specific criteria is just as easy as writing any other type of custom Snort rule. The 802.11 rule engine of Snort-Wireless is built upon the rule engine of the standard Snort distribution and for the most part, it shares the same syntax. The only difference between the

two is that instead of specifying a source IP address and port or destination IP address and port, you simply specify source and destination MAC addresses.
Format: **<action> wifi <mac> <direction> <mac> (<rule options>)**

3.1.2 Rule Actions

The format of information in rule header defines who, where and what type of packets are involved. It also specifies what to do in the event that a packet with all the attributes indicated in the rule should show up. The first element in a rule is the rule action. The rule action tells the Snort what to do when it discovers a packet that matches the rule criteria. There are five available default actions in Snort, defined as follows [Lock2003].

- Alert - generate an alert using the selected alert method, and then log the packet;
- Log - log the packet;
- Pass - ignore the packet;
- Activate - alert and then turn on another dynamic rule;
- Dynamic - remain idle until activtaed by an activate rule and then act as a log rule

More details on these rules and the Snort wireless features can be found in Appendix C.

3.2 Kismet

Kismet is a popular program that combines 802.11 layer 2 wireless network discovery, packet sniffing and IHS capabilities. Kismet works with any wireless card which supports raw monitoring (rfmon) mode, and can sniff 802.11b, 802.11a, and 802.11g traffic. Kismet can identify networks by (a) passively collecting packets and detecting standard named networks, (b) detecting (and given time, decloaking) hidden networks, and (c) by inferring the presence of non-beaconing networks via data traffic [Kers2007].

3.2.1 Feature Overview

As seen in [Kers2007], Kismet has the following features which are useful in different situations for monitoring wireless networks:

- Ethereal/TCPdump compatible data logging;
- Airsnort compatible weak-iv packet logging;
- Network IP range detection;
- Built-in channel hopping and multicard split channel hopping;
- Hidden network SSID decloaking;
- Graphical mapping of networks;
- Client/Server architecture allows multiple clients to view a single Kismet server simultaneously;
- Manufacturer and model identification of access points and clients;
- Detection of known default access point configurations;
- Runtime decoding of WEP packets for known networks;
- Named pipe output for integration with other tools, such as a layer 3 IDS like Snort;
- Multiplexing of multiple simultaneous capture sources on a single Kismet instance;
- Distributed remote drone sniffing;
- XML output;
- Over 20 supported card types.

3.2.2 Useful Applications

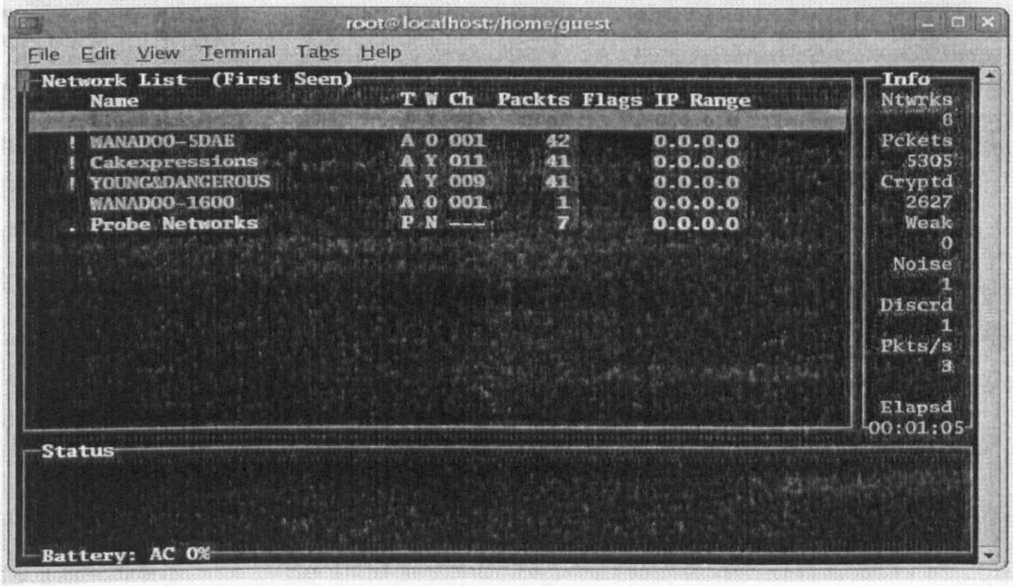
According to [Kers2007] common applications which Kismet is useful for:

- Wardriving for mobile detection of wireless networks, logging and mapping of network location, WEP, etc.
- Site survey for monitoring and graphing signal strength and location.
- Distributed IDS for multiple remote drone sniffers distributed throughout an installation monitored by a single server, possibly combined with a layer 3 IDS like Snort.

- Rogue AP Detection for stationary or mobile sniffers to enforce site policy against rogue access points.

3.2.3 Experimentation on Wireless Attack Using Kismet and Aircrack

The target network would have been found using a network discovery program, such as Kismet in Figure 3.1. After choosing the target network, collecting of the packets being sent across that network can be started for the purpose of breaking the WEP key. One of the tools for breaking WEP key is Aircrack [Crac] in Figure 3.2.

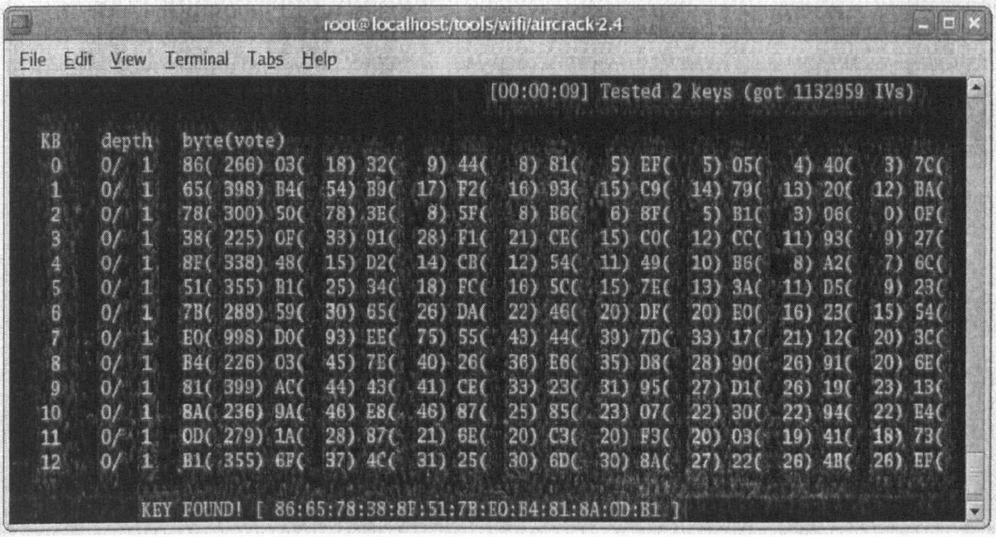


```
Network List—(First Seen)                                       Info
    Name                       T W Ch  Packts Flags IP Range     Ntwrks
                                                                      6
  ! WANADOO-5DAE               A O 001     42       0.0.0.0      Pckets
  ! Cakexpressions             A Y 011     41       0.0.0.0        5305
  ! YOUNG&DANGEROUS            A Y 009     41       0.0.0.0      Cryptd
    WANADOO-1600               A O 001      1       0.0.0.0        2627
  . Probe Networks             P N ---      7       0.0.0.0        Weak
                                                                      0
                                                                  Noise
                                                                      1
                                                                 Discrd
                                                                      1
                                                                 Pkts/s
                                                                      3

                                                                 Elapsd
                                                               00:01:05
Status



Battery: AC 0%
```

Figure 3.1. Using Kismet as a network discovery program to find the targeted network

Many other programs exist for cracking the WEP key including Airsnort, WEPlab, WEPAttack and WEPCrack.

There is a drawback of using this method to crack WEP encryption, and that is the long time taken to get the required amount of packets. One way of overcoming this problem is traffic injection by which more packets can be captured in a given amount of time. This is possible by listening for a certain type of packet, which is then re-injected into the network creating responses and therefore creating more wireless traffic on the target network and speeding up the process.



root@localhost:/tools/wifi/aircrack-2.4
File Edit View Terminal Tabs Help

```
                                   [00:00:09] Tested 2 keys (got 1132959 IVs)

 KB    depth   byte(vote)
  0    0/  1   86( 266) 03(  18) 32(   9) 44(   8) 81(   5) EF(   5) 05(   4) 40(   3) 7C(
  1    0/  1   65( 398) B4(  54) B9(  17) F2(  16) 93(  15) C9(  14) 79(  13) 20(  12) BA(
  2    0/  1   78( 300) 50(  78) 3E(   8) 5F(   8) B6(   6) 8F(   5) B1(   3) 06(   0) 0F(
  3    0/  1   38( 225) 0F(  33) 91(  28) F1(  21) CE(  15) C0(  12) CC(  11) 93(   9) 27(
  4    0/  1   8F( 338) 48(  15) D2(  14) CB(  12) 54(  11) 49(  10) B6(   8) A2(   7) 6C(
  5    0/  1   51( 355) B1(  25) 34(  18) FC(  16) 5C(  15) 7E(  13) 3A(  11) D5(   9) 23(
  6    0/  1   7B( 288) 59(  30) 65(  26) DA(  22) 46(  20) DF(  20) E0(  16) 23(  15) 54(
  7    0/  1   E0( 998) D0(  93) EE(  75) 55(  43) 44(  39) 7D(  33) 17(  21) 12(  20) 3C(
  8    0/  1   B4( 226) 03(  45) 7E(  40) 26(  36) E6(  35) D8(  28) 90(  26) 91(  20) 6E(
  9    0/  1   81( 399) AC(  44) 43(  41) CE(  33) 23(  31) 95(  27) D1(  26) 19(  23) 13(
 10    0/  1   8A( 236) 9A(  46) E8(  46) 87(  25) 85(  23) 07(  22) 30(  22) 94(  22) E4(
 11    0/  1   0D( 279) 1A(  28) 87(  21) 6E(  20) C3(  20) F3(  20) 03(  19) 41(  18) 73(
 12    0/  1   B1( 355) 6F(  37) 4C(  31) 25(  30) 6D(  30) 8A(  27) 22(  26) 4B(  26) EF(

             KEY FOUND! [ 86:65:78:38:8F:51:7B:E0:B4:81:8A:0D:B1 ]
```

Figure 3.2. Using Aircrack to recover the actual WEP key

An example of this is the ARP-request sent by hosts to bind the already known IP address to the unknown MAC address. This type of event happens fairly regularly, as the ARP tables of computers are wiped regularly and ARP-requests are sent to maintain communication between hosts. This type of wireless network attack can be performed by Aireplay, which is part of the Aircrack tools. Aireplay injects traffic by monitoring network activity, and as soon as it detects an ARP-request it copies it and replays it onto the network [Crac]. ARP-requests are a good choice of packet to replay onto a network, because they are always responded to and this activity can severely increase the packets on a wireless network.

The prevention system experiment can be found in Appendix B followed by a summary of Kismet's features in appendix C.

3.3 BRO-Intrusion Handling System

As introduced in [LBNL2007], Bro detects intrusions by first parsing network traffic to extract its application-level semantics and then executing event-oriented analyses that compare the activity with patterns deemed troublesome. Its analysis includes detection of specific attacks (including those defined by signatures, but also those defined in terms of events) and unusual activities (e.g. certain hosts connecting to certain services, or patterns of failed connection attempts). Bro uses a specialised policy language which allows a site to tailor Bro's operation, both as site policies evolve and as new attacks are discovered. If Bro detects something of interest, it can be instructed to either generate a log entry, alert the operator in real-time and execute an operating system command (e.g. to terminate a connection or block a malicious host on-the-fly). In addition, Bro's detailed log files can be particularly useful for forensics [LBNL2007]. Bro targets high data rate (Gbps), high volume intrusion detection. By judiciously leveraging packet-filtering techniques, Bro is able to achieve the necessary performance while running on commercially available PC hardware, and therefore can serve as a cost-effective means of monitoring a site's Internet connection [LBNL2007].

3.3.1 BRO Features and Benefits

Following describes different types of Bro's topologies and their specifications [LBNL2007].

3.3.1.1 Network Based

Bro is a network-based IHS which collects, filters, and analyses traffic that passes through a specific network location. A single Bro monitor, strategically placed at a key network junction, can be used to monitor all incoming and outgoing traffic for the entire site. Bro does not use or require installation of client software on each individual, networked computer.

3.3.1.2 Custom Scripting Language

Bro uses its own language to write policy scripts which contain the "rules" that describe what types of activities are potential to a threat. They analyse and monitor the network activities and initiate actions based on these analyses. Bro language would take some time to learn, once mastered, the user should be able to write or modify Bro policies to detect and alert on most types of network activities.

3.3.1.3 Pre-written Policy Scripts

Bro can use its policy scripts to detect the most common Internet attacks with minimum false positives, i.e., alerts that confuse uninteresting activity with the important attack activity. As mentioned in these supplied policy scripts will run "out of the box" and do not require knowledge of the Bro language or policy script mechanics.

3.3.1.4 Signature Matching Facility

Bro policies incorporate a signature matching facility coupled with a technique looking for specific traffic content. For Bro, these signatures are expressed as regular expressions, rather than fixed strings. It has been noticed that because Bro uses its own special language, it would add a great deal of power to its signature-matching capabilities. This also allows Bro to not only examine the network content, but also to understand the context of the signature which as a result would contribute towards reducing the number of false positives. Bro uses a set of high value signature policies which are selected for their high detection and low false positive characteristics.

3.3.1.5 Network Traffic Analysis

The important feature of Bro is its multitasking ability. It's not only looking for signatures, but also capable of performing the following:

- Analysing network protocols.
- Connections, transactions, data amounts, and many other network characteristics.
- Powerful facilities for storing information about past activities and incorporating them into an analysis of new activities.

3.3.1.6 Identification Followed by Action

Bro policy scripts can generate output files recording the activities seen on the network, including normal and non-attack activities. They can also generate a problem alerts to event logs, including the operating system syslog facility. In addition, scripts can execute programs, which can, in turn, send e-mail messages, page the on-call staff, automatically terminate existing connections, or, with appropriate additional software, insert access control blocks into a router's access control list. With Bro's ability to execute programs at the operating system level, the actions that Bro can initiate are only limited by the computer and network capabilities that support Bro.

3.3.1.7 Snort Compatibility Support

The Bro distribution includes a tool, snort2bro, which converts Snort signatures into Bro signatures. Along with translating the format of the signatures, snort2bro also incorporates a large number of enhancements to the standard set of Snort signatures in order to take advantage of Bro's additional contextual power and reduce false positives. See Appendix B for more detail.

3.3.2 How Is Bro Different From Snort?

Snort is solely *signature* based system, which means it looks for a very specific content in the network stream and reports each instance of a particular signature. Bro has a capability of analysing network traffic at a higher-level of abstraction, and also has facilities for storing information about past activities and incorporating them into an analysis of new activities. Bro also provides a signature mechanism similar to Snort's [LBNL2007].

3.3.3 Signature-Matching Facility

Like other IHSs, i.e. Snort, Bro also provides a signature-matching facility that looks for specific traffic content. In Bro, these signatures are expressed as regular expressions (for Snort, they are fixed strings). Bro adds a great deal of power to its signature-matching capability because its rich language allows it to not only examine the network traffic, but to understand the *context* of the signature, therefore greatly reducing the number of *false positives*, i.e., alerts that confuse uninteresting activity with the important attack activity.

3.3.4 Bro Uses Snort Signatures

Bro includes *Snort-compatibility* support. The Bro distribution includes a script, *snort2bro*, which converts a set of Snort signatures into Bro signatures. Along with translating the format of the signatures, *snort2bro* also incorporates a large number of enhancements to the standard set of Snort signatures to take advantage of Bro's additional contextual power and reduce false positives [LBNL2007].

3.3.5 Will Bro Catch All Intrusions?

Nothing can catch all intrusions. Intrusions are devised and executed by intelligent people who are often actively trying to avoid detection. A skilled attacker might be using a few known techniques that are very difficult to detect. However, experience has shown that most intrusions are attempted by intruders who use standard, well-known techniques that they have learned from others. Usually Bro can detect these. Even expert attackers are susceptible to eventual mistakes leading to tell signatures that Bro will detect. More over, rather than relying solely on unique signature identification, Bro's more advanced features can often discern network anomalies that are caused by hostile activity. Even if previously-unknown attack techniques are used, they can often be detected by observing the corresponding network activity that violates the rules of expected traffic [LBNL2007].

See Appendix B for more details. A summary of Bro's features can be found in appendix C.

3.4 Proventia Desktop Security

Proventia Network Anomaly Identification System (AIS) is another popular Network IHS which is a Windows based platform system [Prov2007]. This network behaviour analysis system enhances network intelligence and security by auditing network flow data from existing infrastructures. Proventia Network AIS provides a clear view of a network's behaviour while automatically detecting active security threats, risky user behaviour, performance issues and noncompliant activities, such as policy violations and unapproved network changes. Proventia Network AIS complements and expands upon protection from the existing Intrusion Prevention System (IPS) by continually examining the behaviour of users, networks and applications across the entire enterprise. Proventia Network AIS mitigates threats and complements firewall and IPS technologies. When anomaly detection is used in conjunction with IPS, businesses gain security and compliance advantages in three areas:

- Greater network visibility - enhancing compliance and overall security posture;
- Multi-layered protection available only from IPS;

- Maximised intrusion prevention and the ability to stop evolving and internal threats.

Proventia Network Anomaly Identification System (AIS) uses network flow data to determine which users and hosts communicate with each other and how. Proventia Network AIS continuously delivers a clear view of your network inventory and behaviour, while automatically detecting active security threats, risky user behaviour, performance issues and noncompliant activities, such as policy violations and unapproved network changes.

3.4.1 Protection Features

3.4.1.1 Stateful Flow Reassembly

Proventia Network AIS goes beyond the basic NetFlow collection in order to perform full reassembly of all disparate flow information, converting raw flow into actionable security information.

3.4.1.2 Relational Modelling

Proventia Network AIS uses patented technology to model networks and understand internal threats.

3.4.1.3 Multi-dimensional Identification

Proventia Network AIS leverages six unique identification systems to constantly monitor for threats, network abnormalities and misuse. These are:

- *Worm Identification* for identification of propagating patterns of abnormal behaviour;
- *Fingerprint Identification* for identification of traffic that violates a behavioural fingerprint;
- *Rate-based Anomaly Identification* for determining of sudden shifts from baseline traffic levels over time;
- *Recon Identification* for identification of slow scans, fast scans, "stealth" scans and host sweeps;
- *Insider Misuse* for identification of behavioural violations of specified security policy;

- *Availability Outages* for identification of drops in traffic on critical servers and links.

Other features can be found in Appendix C.

3.4.1.4 Multi-layered Protection

Proventia Network AIS integrates with Proventia Network Intrusion Prevention, Proventia Network Enterprise Scanner and other third party security technologies through Proventia Management Site Protector. As part of a multi-layered security strategy, Proventia AIS and Proventia IPS provide the most effective protection from security threats regardless of where the threat originates.

Proventia Network AIS complements Proventia IPSs with:

- Network-wide Visibility;
- Behavioural Identification;
- Network Usage and Performance.

Proventia Network AIS complements Proventia Network Enterprise Scanner with:

- Passive Network Assessment;
- Service Usage Statistics;
- Policy Violations.

3.4.2. Management Features

3.4.2.1. Centralised Management System Integration

Proventia Network AIS can be centrally managed using the Proventia Management Site Protector Security management system. Site Protector is a scalable system that allows your staff to control, monitor and analyse events efficiently. It delivers a single interface and consolidated workflow structure for managing, monitoring and measuring security [Prov2007].

3.4.2.2. Web-based Local Management

Proventia's web-based local management interface (LMI) simplifies device setup, management and monitoring.

3.4.2.3 SNMP Management

Proventia integrates with third-party network management products to provide key operational status indicators to network operations and security operations groups [Prov2007]. See Appendix C for more details.

3.5. AirDefense

AirDefense [Aird2006] is another popular proprietary 802.11a/b/g WLAN Intrusion Handling System that identifies security risks and attacks and provides real-time network audits and monitors the health of the WLANs.

3.5.1. Features of AirDefence

- It detects all rogue WLANs (Full functionality of AirDefense *RogueWatch*).
- It secures a wireless LAN by recognizing and responding to intruders and attacks, as they happen.
- It performs real-time network audits to inventory all hardware, tracks all wireless LAN activities and enforces WLAN policies for security and management.
- It monitors the health of the network to identify and respond to hardware failures, network interferences and performance degradation.

Based on the AirDefense platform, AirDefense guard consists of distributed sensors and server appliances. The remote sensors sit near 802.11 access points in order to monitor all WLAN activities and report back to the server appliance. It thus analyses the traffic in real time. The platform allows enterprises to easily upgrade from AirDefense RogueWatch to AirDefense guard.

3.5.2. Wireless Intrusion Identification & Response

AirDefense Guard provides intrusion identification for wireless LANs based on signature analysis, policy deviation, protocol assessment, policy deviation and statistically anomalous behaviour. AirDefense is designed to detect and respond to:

- Identity theft by stealing an authorised MAC address: AirDefense tracks the digital "fingerprints", vendor specific characteristics and personal trademarks of authorised users to identify intruders in the network.

- Denial of Service (DoS) attacks: AirDefense quickly recognises the early signs and protocol abuses of a DoS attack that jams the airwaves and shuts down a wireless LAN.
- MITM attacks: AirDefense detects MITM attacks by ensuring that access points only operate on set channels and proper protocols are used.

More on AirDefence can be found in Appendix C.

3.6 Comparison of Identification and Response Techniques for Selected Wireless IHSs

Table 3.1 compares the recent experimental IHS systems. The comparison is based on their identification and response techniques and capabilities in defending real time attacks, etc. Here the intrusion identification method is defined using the definition of intrusion from [lim2003] as "an activity for identification of a sequence of related actions by a malicious adversary that results in the occurrence of unauthorised security threats to a target computing or network domain". The corresponding intrusion response method is an activity for blocking of the intrusion (e.g. by not forwarding a malicious packet to the destination application) or preventing possible damage (e.g. switching off the station). Table 3.1 will be used to define the components of identification and response systems in the new IHS reference model in the next chapters.

Table 3.1. Detailed comparison of selected IHSs.

		Snort	Kismet	Bro-IDS	Proventia	AirDefence
Legacy IDS/IPS	IDS	x	x	x	x	x
	IPS	x	——	x	x	x
Type of IHS	Open source	x	x	x	——	——
	Proprietary	——	——	——	x	x
Network Type	Network Based	x	x	x	x	x
	Host Based	——	——	——	——	——
Identification Techniques	Anomaly*	——	a,b	a,b	a,b	a,b
	Misuse*	c,d	c,d	c,d	——	c,d
	Live Packet analysis	——	x	x	x	x
Prevention Techniques	Active*	e,f,g	——	e,g	e,f	e,g
	Passive*	h,i,j,k	——	h,k	h,i,j	h,i,k
Coding Format		Open Source	Open Source	Open Source	Proprietary	Proprietary
Signature Compatibility with Snort		Yes	Yes	Yes	N/A	No
Signature Compatibility with other IDS/IPS		No	No	Only with Snort	N/A	No
Identifies all intruders		No	No	No	No	No
Platform		Linux	Linux	Linux	Windows	Windows
Real-time tracking		——	x	x	x	x

*Anomaly Methods: *a) Policy compliance/script,b)Statistically anomalous behavior / unusual activities.*Misuse Methods: *c) Signature recognition, d) signatures based on WLAN Protocols.* Active response (Automatic Prevention Methods): *e) Blocking Traffic, f) Host Termination, g) ACL creation.* Passive response (Manual Prevention Methods): *h) Alerting Administrator, i) Blocking Traffic.j) Host Termination, k) Access list.*

3.7 Conclusion

Looking at above selected IHSs and their structure including identification and response (prevention) subsystems, the systematic categorisation described below has been achieved and used later to define the components of our novel reference model.

Ad hoc networking allows portable devices to establish a communication independent of a central infrastructure. However, the fact that there is no central infrastructure and devices can move randomly makes this type of networks more vulnerable to intrusion threats. Therefore, the author would focus on wireless *ad hoc* networks architecture. All the IHSs (Legacy IDSs/IPSs) described here are only capable of working in an infrastructure mode of WLANs and they are all network based. Further more, they rely on an Access Point (AP) as the only concentration point. Not only none of these systems are supporting Wireless *ad hoc* networks but also some of them (i.e. AirDefence) have been designed in a way that identification of an *ad hoc* network would be considered as a possible rogue AP attack causing a security risk to their networks. In network based WLANs traffic monitoring is usually done at the gateways or AP. The mobile *ad hoc* environment does not have such traffic concentration points where the IHS can collect audit data (logging history) for the entire network [Zhan2000]. The intrusion identification algorithms should thus be made to work on this partial and localised information. For these reasons, a combination of host and network based (Hybrid System) seems to be the most suitable approach for WLANs due to their changing topology.

4. Wireless Inter IHS Message Format

According to the design requirements of a new IHS, each IHS must operate autonomously and protect one host where it resides. Because of the nature of WLANs where there is not any specific point of entry for packets, it must also monitor traffic between the hosts and communicates with other IHSs in order to update and share information about intrusions. This would help to block further access by the detected intrusive device before they become capable of attacking other working WLAN devices.

IHSs residing in different hosts may be produced by different vendors and use different methods for identifications of and responses to the intrusions. Therefore, in addition to communication between the internal components of IHS systems residing in the individual host, a Unified Wireless Intrusion Handling Report Format (UIHRF) should also be used to translate the non standard intrusion report's formats from different type of IHSs to a UIHRF. The UIHRF will use IIHSM standard data format to prepare the reports for delivery between IHSs using IIHSCP. The rationale behind the wireless format of this standard was described in chapter 1.4. A good example for this can be a letter (analogous to UIHRF) that needs to be sent to multiple overseas recipients who don't talk the same language. The letter is written in English (UIHRF) so it can be read by all recipients and sent by airmail standard envelopes (analogous to IIHSM). The post man and aircraft (analogous to IIHSCP) will be used to deliver this letter to destination.

The need for IIHSCP was first identified in [Pakš2006, Sale2007]. Later it was discovered that there was an ongoing work by IETF towards the IDMEF and its requirement [Deba 2007]. In fact, this RFC has never acquired acceptance, which further confirms the complexity of this topic. Thus currently there is a new version of this RFC [Wood2007], complementary to RFC4765 [Deba2007].

In [Wood2007] the requirement of IDEMF is presented by the author, and it is proposed that such requirement should be used to evaluate existing communications protocols for wired networks. Obviously, it would not be acceptable to ignore this effort but one have to keep in mind the fact that it is generic and not targeting Wireless IHSs and relevant specific issues. In addition to that, in [Deba2007], the IETF disclaims any knowledge of the fitness of this RFC for any purpose, and it warns that readers of this RFC should exercise caution in evaluating its value for implementation and deployment.

4.1 Analysis of the IETF Activities on different IHS Report Format

The UIHRF messages should be stored in Intrusion Handling Message Exchange Database (IHMD) within inter IHS communication subsystem/module of different IHSs using IIHSM as shown in Figure 4.1 below.

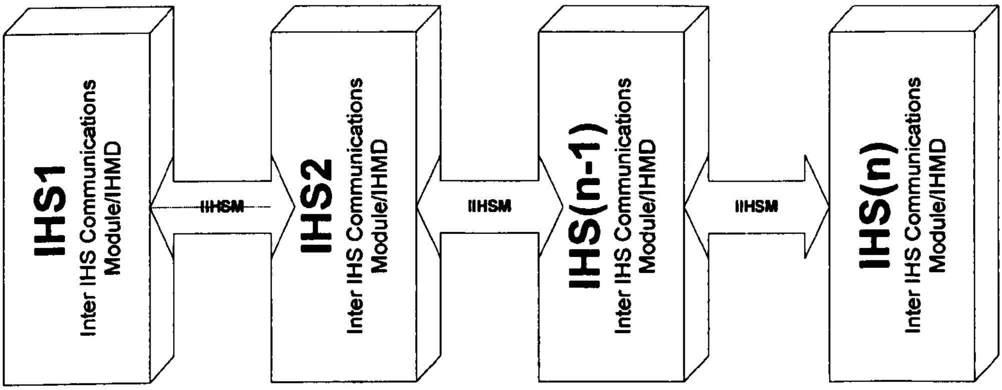

IHS1
Inter IHS Communications Module/IHMD
IIHSM
IHS2
Inter IHS Communications Module/IHMD
IIHSM
IHS(n-1)
Inter IHS Communications Module/IHMD
IIHSM
IHS(n)
Inter IHS Communications Module/IHMD

Figure 4.1. Inter IHS communication module with a database of unified messages stored in IHMD.

The IIHSM should be considered as a standard data format that automated intrusion handling systems can use to report alerts about events that they deem suspicious in the unified format (UIHRF). The main reason for this, its compatibility with the developed new IHS reference model, which should also run as efficient as possible in a host based *ad hoc* WLANs contrary to the proposed IDMEF in [Wood2007] which is developed to work only on Network based IHS systems in Wired LANs. The development of IIHSM standard format will enable interoperability among any types of Wireless IHS Systems, including commercial, open source, and research systems. This will allow users to mix-and-match the deployment of their systems according to their strong and weak points to obtain an optimal implementation. Therefore, it should be assumed that IIHSM is protocol independent and can be used by any type of IIHSCP i.e. Wireless format of the IDMEF Communication Protocol (IDP) [Wood2007] or Intrusion Detection Exchange Protocol (IDXP) [Fein2007]. Figure 4.2 shows the relations between these three RFCs with a comparison to IIHSM.

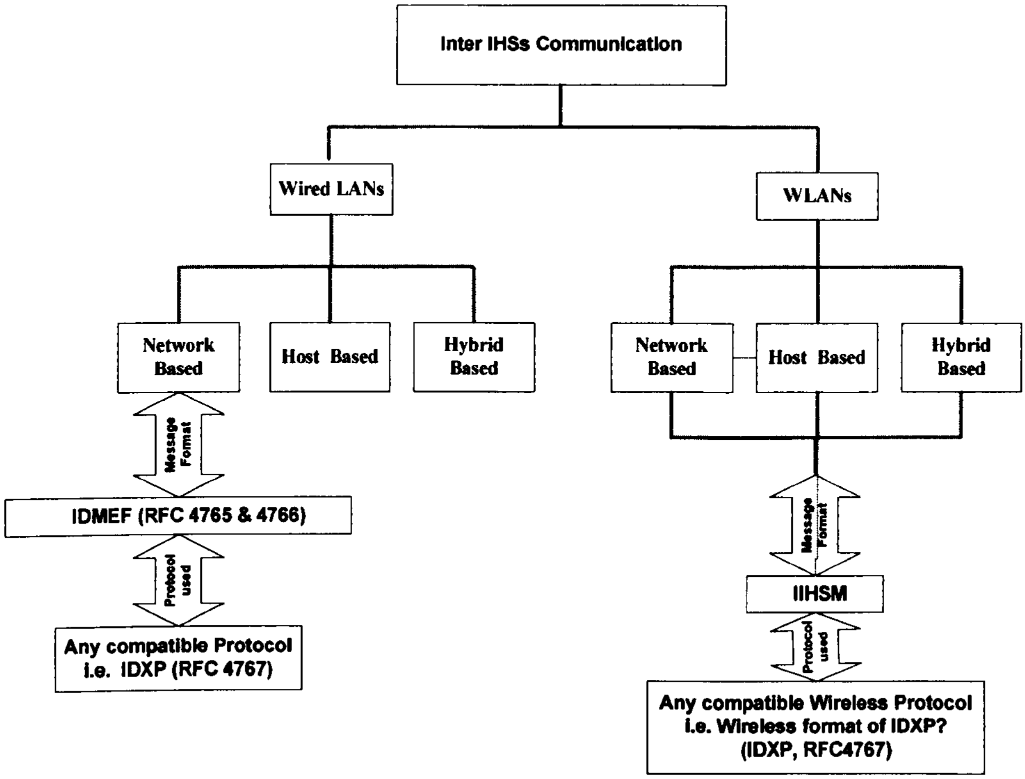


Figure 4.2. Diagram of relationships between RFC 4765, 4766 and 4767 with a comparison to IIHSM.

4.1.1 IHMD Implementation Assumptions

IHMD is a database of unified messages within IICS of individual IHSs. These messages are carried between IHSs using IIHSM. The implementation of such database has been explained in IDMEF development in RFC4765 [Deba2007].

In [Deba2007] it is suggested that the best place to implement the IHMD database is in the data channel between an intrusion identification analyser (or identification subsystem, can also be a sensor for network based) and the manager (or console) to which it sends alarms. This fully complies with our design where the IHMD is placed in Inter IHS communication module which is residing in between Identification/Response module and management console. In [Deba2007] there are also other places suggested where the IHMD can be useful:

1. A single database system that could store the results from a variety of intrusion handling systems would make it possible for data analysis and reporting activities to be performed on "the whole picture" instead of just a part of it;

2. An event correlation system that could accept alerts from a variety of legacy intrusion detection products would be capable of performing more sophisticated cross-correlation and cross- confirmation calculations than one that is limited to a single product;

3. A graphical user interface that could display alerts from a variety of intrusion handling systems would enable the user to monitor all the products from a single screen, and require him or her to learn only one interface, instead of several; and

4. A common data exchange format would make it easier for different organisations (users, vendors, response teams, law enforcement) to not only exchange data, but also communicate about it. The diversity of the application of the IHMD, necessitate consideration when selecting its method of implementation.

Previous research [Wood2007] also defines requirements for the Intrusion Detection Message Exchange Format (IDMEF) which is a product of the Intrusion Detection Exchange Format Working Group (IDWG). IDMEF was planned to be a standard format that automated Intrusion Identification Systems (IISs) can use for reporting what they have deemed to be suspicious or of interest. The four reasons IDMEF format should be useful are given by [Wood2007] and described below. Note that similar approach should also be applied to IIHSM:

- A number of commercial and free IHSs are available and more are becoming available all the time. Some products are aimed at detecting intrusions on the network; others are aimed at host operating systems, while still some others are aimed at applications. Even within a given category, the products have very different strengths and weaknesses. Hence it is likely that users will deploy more than a single product, and users will want to observe the output of these products from one or more manager(s). A standard format for reporting will simplify this task greatly.

- Intrusions frequently involve multiple organisations as victims, or multiple sites within the same organisation. Typically, those sites will use different IHSs. It would be very helpful to correlate such distributed intrusions across multiple sites and administrative domains. Having reports from all sites in a common format would facilitate this task.

- The existence of a common format should allow components from different IHSs to be integrated more readily. Thus, research in the Intrusion Handling reference model should migrate into commercial products more easily.

- In addition to enabling communication from IHS identification to an IHS manager, the IIHSM notification system may also enable communication between varieties of IHS components.

As defined in the terms above, it's been assumed that identification subsystem determines somehow that a suspicious event has been detected or an alert has been sent to a management console/operator. It's also been assumed that the identification subsystem and management console are separate components, and that they are communicating pair wise across a TCP/IP network [Wood2007].

4.2 Requirement for IIHSM

The IIHSM is intended to be independent of the IDMEF and its Communication Protocol (IDP) [Wood2007] or IDXP [Fein2007]. It should be possible to use a completely different transport mechanism without changing the IIHSM format. The main reason behind this requirement is to ensure a clear separation between semantics and communication mechanisms. This makes IIHSM to be remaining compatible with future enhanced new protocols.

The requirement of IIHSM may structurally be similar to IDMEF (RFC4566) despite IDMEF has never been defined for *ad hoc* WLANs. Therefore, the future research will look at this matter. To find out about the requirement of IIHSM see Appendix D.

4.3 Cisco Approach to IIHSM

Cisco had also adapted the IDMEF and named it Cisco Intrusion Detection System Message Exchange Format (CIDMEF). It is a message exchange format for Cisco IDS architecture data and the specification of CIDMEF format is based on XML/1.0 schema document [Cisc].

4.4 Using XML for IIHSM Communications

Extensible Markup Language (XML) is becoming the universal format for data exchange between applications [Milo2003]. XML is gaining widespread attention as a language for representing and exchanging documents and data on the Internet, and as the solution to most of the problems inherent in Hyper Text Markup Language (HTML). It is a very flexible text format and is a simplified version of the Standard Generalised Markup Language (SGML), syntax for specifying text markup defined by the ISO 8879 standard [Sper2000]. XML was published by the World Wide Web Consortium (W3C) as a metadata language i.e. a language for describing other languages that enable an application to define its own markup. It allows the definition of customised markup languages for different types of documents and different applications. This differs from HTML, in which there is a fixed set of identifiers with preset meanings that must be "adapted" for specialised uses. Both XML and HTML use elements (tags) (identifiers delimited by '<' and '>') and attributes (of the form "name='value'"). But where "<p>" always means "paragraph" in HTML, it may mean "paragraph", "person", "price", or "platypus" in XML, or it might have no meaning at all, depending on the particular application [Deba2007].

4.4.1 XML Implementations for IHS

Two implementations of the IDMEF were originally proposed to the Intrusion Detection Working Group (IDWG) [Wood2007]: one using the Structure of Management Information (SMI) to describe a Simple Network Management Protocol (SNMP) MIB, and the other using a Document Type Definition (DTD) to describe

XML documents. These proposed implementations were reviewed by the IDWG at its September 1999 and February 2000 meetings. It was decided at the later meeting that the XML solution was best at fulfilling the IDWG requirements [Deba2007].

A standard intrusion signature representation model using the XML specification has been shown in [Cans2002]. The importance of intrusion signatures and their usage by intrusion detection systems, like ACME!-IDS are also discussed. The research has paid special attention to the need of a unified way of storing, processing, analysing and reporting intrusion patterns. The research also shows that the modularisation is a very important feature of the standard intrusion signature representation model. It provides the possibility for different systems to share information related to intrusive events. This modularisation also makes the parsing of Anomaly Identification Systems (AISs) easy and diminishes the process overhead which is a good and desirable feature of IHSs. The use of XML comes with a new standardisation tendency regarding information sharing. It supplies pre-processing of data, organisation uniformity and easiness of graphical signature representation.

4.5 Conclusion

This chapter introduced the new IIHSM as a standard data format for the Inter IHS communications subsystems of the new IHS reference model. In addition to communication between the internal components of IHS systems residing in the individual host, a UIHRF should also be used to translate the non standard intrusion report's formats from different type of IHSs to a UIHRF. This chapter has also introduced the requirement for IIHSM which was intended to be independent of the IDMEF Communication Protocol (IDP) [Wood2007] or IDXP [Fein2007]. The XML Implementations for IHS has also been explained. The two implementations of the IDMEF that were originally proposed to IDWG [Wood2007] were also explained. These proposed implementations were reviewed by the IDWG and it was decided at the February meeting that the XML solution was best at fulfilling the IDWG requirements [Deba2007]. At the time of this research, it had been planned to use XML for IIHSM implementation however it came to the author's attention that a new RFC [Wood2007] has been published, which uses a similar approach but it is only network based and it doesn't precisely specify the type of physical layer of LAN which the IHS may operate. An important message has also been noted in RFC4766 which reads as: "This RFC is not a candidate for any level of Internet Standard. The IETF disclaims any knowledge of the fitness of this RFC for any purpose and in particular notes that the decision to publish is not based on complete IETF review for such things as security, congestion control, or inappropriate interaction with deployed protocols". This gives an indication that the RFC may be modified or updated based on our requirement for XML implementation with IIHSM or may use similar approach with some modifications. Next chapter will look at the ontology approach to development of IHS reference model.

5. Ontology Approach to Construction of New IHS Reference Model

An Intrusion Handling System Modelling Ontology (IHSMO) is constructed and discussed phase by phase in this chapter. This ontology is based on the concepts of various ontology modelling and simulation tools. Careful attention is given to consistent development of ontologies in order to support two functions. Firstly, to be able to manage the dependencies between ontologies and at the same time to keep and restore consistencies of them if they are changed in order to accommodate new information. Secondly, to adjust the representation of the domain according to the future changes and updates to ontologies as future requirement and expectation of domain of knowledge may change.

5.1 Design of IHS Modelling Ontology

A combination process has been considered and will be applied to the construction of IHS Modelling Ontology (IHSMO) since different components of new IHS model were previously defined in this project, it is feasible to take the combination approach instead of a top-down or bottom-up process [Usch1996].

One of the key important factors in designing a new ontology is efficiency in design. This can be achieved by splitting the ontology into several component ontologies. We call this a "collaborative design". In collaborative design each component of ontologies will be built first, and then they all are compiled into a unique and unified ontology. To accomplish this, it is necessary that every component of ontologies is identified separately according to their domain or conceptual level [Sale2010a].

Therefore in order to design the Identification Subsystem Modelling Ontology (ISMO), all the component of the ISMO will be designed separately, and then they are all compiled and subsequently composed a single unified ISMO ontology [Sale2010a].

5.1.1 Design of Identification Subsystem Ontology

The Intrusion Identification Subsystem (IIS) is capable of identifying attacks against a host and network (Hybrid system) by mirroring traffics and performing the action of identification which identifies malicious attacks on host and network (Hybrid) resources.

It should also be noted that this is a temporary defence mechanism; it is not a permanent prevention of attacks like legacy IPSs. The identification subsystems would use various methods for identifying malicious attacks based on previous research and detailed comparison of selected IHSs shown in Table 3.1.

Figure.5.1 is illustrating the methods for Identification and Response subsystems based on previous research and case studies.

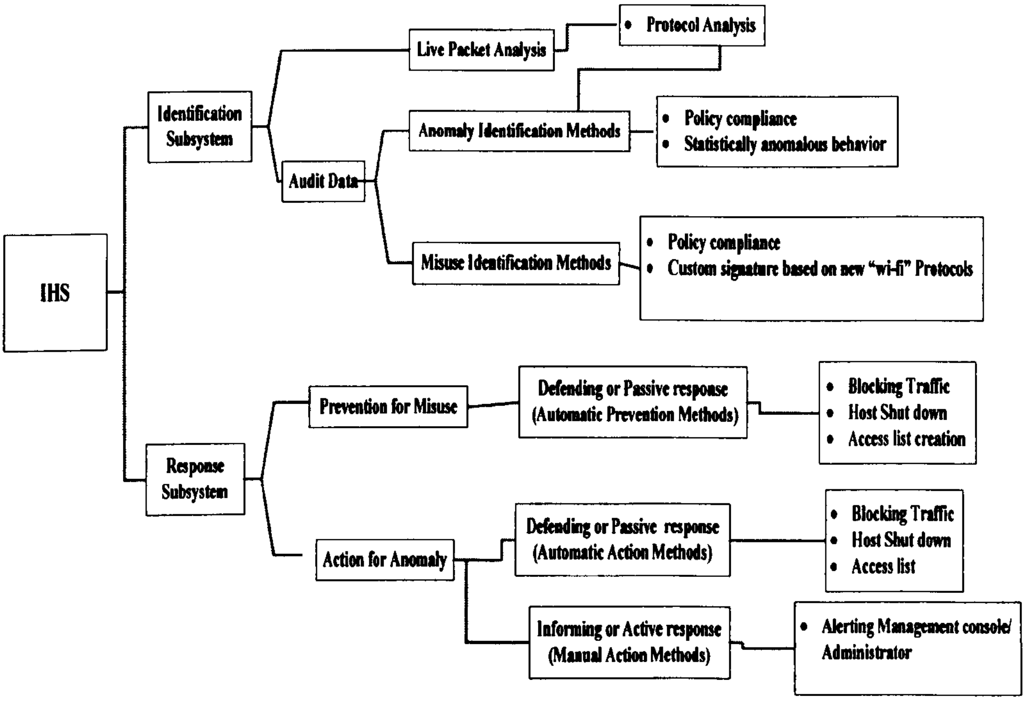

IHS
Identification Subsystem
Live Packet Analysis
• Protocol Analysis
Audit Data
Anomaly Identification Methods
• Policy compliance
• Statistically anomalous behavior
Misuse Identification Methods
• Policy compliance
• Custom signature based on new "wi-fi" Protocols
Response Subsystem
Prevention for Misuse
Defending or Passive response (Automatic Prevention Methods)
• Blocking Traffic
• Host Shut down
• Access list creation
Action for Anomaly
Defending or Passive response (Automatic Action Methods)
• Blocking Traffic
• Host Shut down
• Access list
Informing or Active response (Manual Action Methods)
• Alerting Management console/ Administrator

Figure 5.1. Structure, definitions and methods of IHS model

It should be noted that the design of Identification Subsystem Ontology (ISO) will be based on Figures 2.8, 2.9, 5.1 and analysis of previous case studies discusses earlier.

The slot, type, cardinality and other constraints/facets of the identification ontology are shown in Table 5.1. The complete ISO will be presented later, when all the ontology components for identification subsystem accumulate in order to compose the ISO [Sale2010a].

Table 5.1. The concepts for Identification Ontology and their facets

Ontology: Identification Subsystem:				
Slot	**Type**	**Cardinality**	**Other Facets**	**Allowed Value**
Hybrid	**E**	**Single**	**Class=Identification Method**	**Signatures,**
Misuse	**E**	**Single**	**Class=Identification Method**	**Various Methods**
Anomaly	**E**	**Single**	**Class=Identification Method**	

A value-type facet shown below describes what types of value can fill in the slot or concept. The most common value types are alphanumeric, string, number and enumerated. Some systems distinguish only between single cardinality by allowing at most one value and multiple cardinalities by allowing any number of values. For simplicity the following symbols are used to represent data/value types.

A: Alphanumeric, E: Enumerated, N: Number and S: String

The approach used in Table 5.1 to represent the composite types and other facets are also used in [Noy2001, Sale2010a].

Figure 5.2 represents the Identification ontology with concepts and Table 5.2 shows the corresponding axioms. Similar approach to represent axioms in Table 5.2 has also been used in [Mian2003].The ontology in Figure 5.2 follows the axioms as partOf [Sale2010a].

As explained in chapter two, attributes/slots are the properties of each concept describing various features and attributes of the concepts therefore any new added attributes will not change the concept itself and so the ontology as a whole. The ontology may change if one or more concepts completely changes or a new concept is added to the existing ontology.

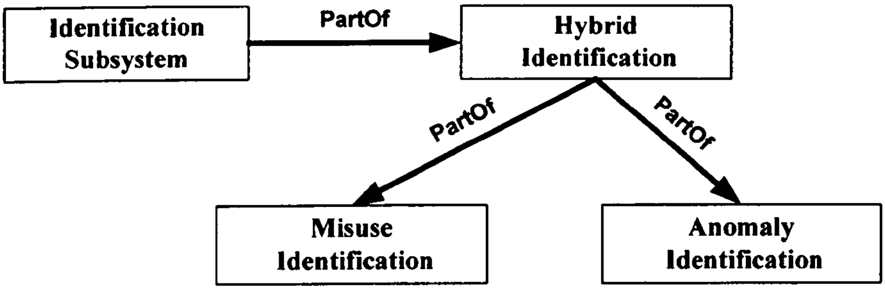


Figure 5.2. Basic view of identification ontology

Since ontology is not a formal mechanism such as set theory therefore different shorthands may be used for different concepts/classes and attributes/slots in order to develope different ontologies.

For simplicity, following letters are used to represent different concepts for identification ontology [Sale2010a]. The same approach will be used for the remaining ontologies in this chapter. The followings are the only concepts available for the identification ontology:

A: Identification subsystem

A1: Hybrid identification method

B: Misuse identification method

C: Anomaly identification method

It can be noted that B and C are both subset of A1, therefore: $B,C \subseteq A1$, therefore $A1=B\cup C$. However, A1 is also a subset of A; hence concepts B and C are as well a subset of concept A. This means that: $A1 \subseteq A$ where $A1=B\cup C$ therefore $B\cup C\subseteq A$. In conclusion: $A= B\cup C$ or $A=A1$.This means that identification ontology is formally a hybrid identification ontology which consists of Misuse and Anomaly identification ontologies.

In the identification ontology, there must be a Hybrid, Misuse and Anomaly identification methods that can be represented by axioms 7, 8, 9. The concepts A, A1, B and C all together define the whole identification ontology represented by axioms shown in the Table 5.2.

a. Hybrid Identification

As shown in Table 5.2, the Hybrid identification method consists of both misuse and anomaly methods [Sale2010a].

b. Misuse Identification

The Misuse identification method can be expressed by Rule Set Database (RSDB) concept. The RSDB concept belongs to the class Misuse as shown in Figure 5.3 [Sale2010a].

Table 5.2. Axioms for Identification Ontology

ID	**Axioms**	**ID**	**Axioms**
1	partOf(Hybrid, Identification)	**2**	Composite(Hybrid)
3	partOf(Anomaly ,Hybrid)	**4**	partOf(Misuse ,Hybrid)
5	Composite (Anomaly)	**6**	Composite (Misuse)
7	($\forall$x, $\exists$y) Identification (x) $\wedge$ Hybrid (y) $\wedge$ partOf(Hybrid, Identification) $\vee$ True		
8	($\forall$x, $\exists$y) Hybrid (x) $\wedge$ Anomaly (y) $\wedge$ partOf(Anomaly ,Hybrid) $\vee$ True		
9	($\forall$x, $\exists$y) Hybrid (x) $\wedge$ Misuse (y) $\wedge$ partOf(Misuse ,Hybrid) $\vee$ True		
10	($\forall$x, $\exists$,b,c) Hybrid (x) $\wedge$ Misuse(b) $\wedge$ Anomaly(c) $\wedge$ partOf(b, x) $\wedge$ partOf(c, x)		
11	($\forall$x, $\exists$a) Identification (x) $\wedge$ Hybrid (a) $\wedge$ partOf(a, x)		

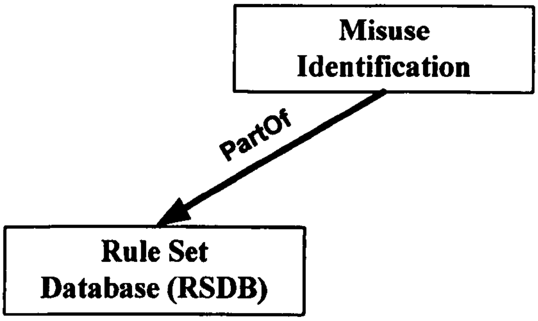


Figure 5.3. Misuse Ontology

For simplicity notation B1 is used to represent the RSDB concept, therefore B1 is the Rule Set Database (RSDB). B1 is a subset of B, hence: B1 $\subseteq$ B. Table 5.3 shows the axioms that are followed by Misuse ontology. The RSDB concept in turn can be expressed by the attributes/slots as Methods A1 to A*n*. The different methods will be described later in the next sections. These attributes belong to the class RSDB as shown in Figure 5.4 [Sale2010a].

Table 5.3. Axioms for Misuse Ontology

ID	Axioms	ID	Axioms
1	partOf(RSDB, Misuse)	2	Composite (RSDB)

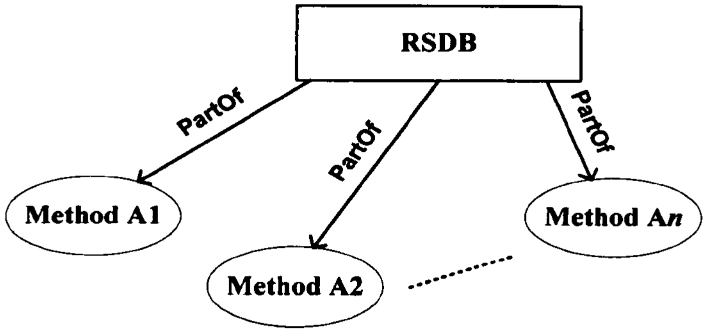


Figure 5.4. RSDB Concept

The following notations are used to represent attributes/slots of RSDB concept:

b1: Method A1

b2: Method A2

⋮ ⋮

b*n*: Method A*n*

It should be reminded that attributes/slots are always part of their concept/class therefore attributes/slots b1 to b*n* are all subsets of B1. If in future, a new attribute becomes available, it will not affect its concept within the ontology. Here it is assumed that currently there are only the attributes/slots b1 to b*n* available for the RSDB concept. The existing methods will be explained in the next section.

Formally using set theory: b1$\subseteq$B1, b2 $\subseteq$ B1,...., b*n* $\subseteq$ B1. Since all attributes b1 to b*n* are part of concept B1, it can be noted that: B1=b1$\cup$b2.....$\cup$b*n*. The same logic, concept and approach have been used to define the remaining ontologies in this chapter. Table 5.4 shows the axioms abided by the RSDB concept.

Table 5.4. Axioms for RSDB concept

ID	Axioms	ID	Axioms
1	partOf (MethodA1, RSDB)	2	partOf (MethodA2, RSDB)
3	partOf (MethodA*n*, RSDB)	4	Primitive (Method A1)
5	Primitive (Method A2)	6	Primitive (Method A*n*)

5.1.1.1 Misuse Identification Methods

In general, Misuse identifications can be categorised as following:

- ***Policy Compliance***

  A security policy is a set of rules established by a network administrator to safeguard the network from attacks. In general, rules are normal permitted activities by users. The identification subsystem will react as soon as it identifies a deviation from the rules [Sale2010a].

- ***Signature Recognition***

  These are recorded signatures from previous attacks which system holds these signatures in its database. The system should recognise the intrusion by their defined markings or the vendor-specific fingerprints and identify these previously documented attacks by looking at its recorded database [Sale2010a].

- ***Custom Signature Based on New "wi-fi" Protocols***

  These are recorded signatures for new "wi-fi" Protocols.

c. Anomaly Identification

The anomaly identification is based on a profile that defines normal user activity. Therefore, an anomaly based IHS should generate alarms for previously unknown attacks, as long as the new attack deviates from normal user activity. This makes the anomaly-based IHS being capable of identifying novel attacks when they used for the first time. The Anomaly concept can be expressed by the attributes/slots:

Method L1, MethodL2 and Method Ln. The different methods will be described later in the next sections. These attributes belong to the class anomaly as shown in Figure 5.5 [Sale2010a].

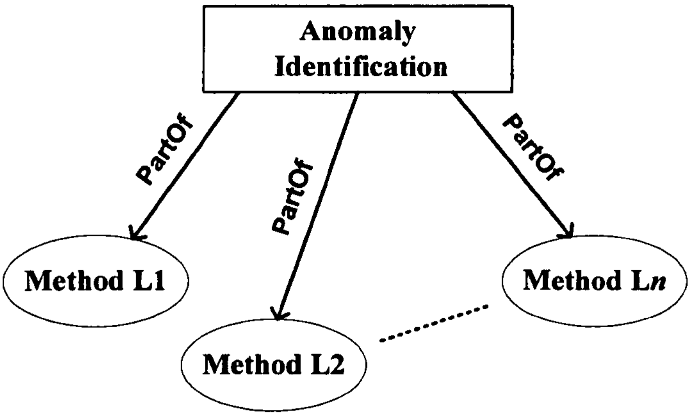


Figure 5.5. Anomaly concept

The following notations are used to represent slots of Anomaly concept:

c1: Method L1

c2: Method L2

: :

cn: Method Ln

It can be noted that these slots are subsets of concept C, that is:

$c1 \subseteq C$, $c2 \subseteq C$,...., $cn \subseteq C$. Hence: $C=c1 \cup c2.....\cup cn$.

Table 5.5 shows the axioms abide by the Anomaly concept.

Table 5.5. Axioms for Anomaly concept

ID	Axioms	ID	Axioms
1	partOf (Method L1, Anomaly)	2	partOf (Method L2, Anomaly)
3	partOf (Method L*n*, Anomaly)	4	Primitive (Method L1)
5	Primitive (Method L2)	6	Primitive (Method L*n*)

5.1.1.2 Anomaly Identification categories

In general, anomaly identifications can be categorised as following:

- *Protocol Specification and Analysis (Live Packets Analysis).* Identification of novel attacks, risks and threats that have not been previously recorded or documented. This is mainly based on the intruder's approach and the way the intruder breaks 802.11 protocols of communication [Sale2010a].
- *Policy Compliance/Script.* Policy compliance deviation is achieved by monitoring the normal behaviour of a device. It is determined on a customised basis for acceptable behaviour for each device. This program contains set of rules that describe what types of activities are potentially considered as a vulnerability to the system. They would monitor and analyse the network events and would initiate identification actions based on these analyses [Sale2010a].
- *Statistically Anomalous Behaviour / Unusual activities.*

  Dynamically alert the operator in real-time to abnormal behaviour of network devices, such as repeatedly failed login access to a system or transformation of 5 MB file from one wireless host to another host at 5 am.

5.1.1.3 Identification Methods

The anomaly and misuse identifications use the following methods to identify the intrusions. These are stored in an identification database [Sale2010a].

A. Misuse Identification Database, Methods A1 to An

A.1. Decoding based (layer 2 Identifications):

- ARP decoding,
- Layer 2 header decoding.

A.2. Protocol based (layer 3 Identification):

- TCP, UDP,ICMP and IP
- Test the IP header's TTL field value,
- Test the ICMP type field against a specific value,
- Test the ICMP ECHO ID field against a specific value,
- Test the ICMP ECHO sequence number against a specific value,
- Test the ICMP code field against a specific value,
- Filter for SYN/FIN/RST TCP packets,
- Test the TCP sequence number field for a specific value,
- Test the packet's payload size against a value,
- Watch the IP option fields for specific codes,
- Test the TCP acknowledgement field for a specific value,
- Test the TCP flags for certain values,

- Process TCP fragments,
- Test the fragmentation bits of the IP header,
- Test the IP header's fragment ID field for a specific value,
- Flag HTTP-based worm sources such as Code Red,
- Detailed analysis of http requests,
- Detailed analysis of http replies.

A.3. Frame structure based (layer 2 Identification)

- Tests the entire frame control field,
- Tests the 802.11 frame's type,
- Tests the 802.11 frame's subtype,
- Tests the distribution system frame control flag,
- Tests more fragments frame control flag,
- Tests the retry frame control flag,
- Tests the power management frame control flag,
- Tests more data frame control flag,
- Tests the WEP frame control flag,
- Tests the order frame control flag,
- Tests the frame's duration/id field,
- Tests the frame's BSSID,
- Tests the frame's sequence number,
- Tests the frame's fragment number,
- Tests the frame's 4th address field.

B. Live Anomaly Identification, Methods L1 to Ln

B.1. Alert based:

- Send alerts to syslog,
- Send WinPopup alert messages to Windows machines,
- Send a record log in a cvs file alerts to a *NIX socket that another program can listen on,
- Log alerts to event logs,
- Write the alert in a simple format,
- Packets logged to a decoded ASCII format or a binary log file,
- Record log in a database,
- Sending alert to administrator's email or pager.

B.2. Protocol based (layer 3 Identifications)

- FTP analysis,
- Identify and log TFTP sessions,
- ICMP analysis, DNS analysis,
- Search for a set of patterns in the packet's payload,
- Logging based, detects password scans,
- Analysing rlogin/telnet, access and record connection events.

B.3.Frame structure based (layer 2 Identification)

- Networks with known WEP keys can be decrypted in real-time,
- Tracks probe requests and responses, Fuzzy encryption detection ,
- BSSID (MAC address) of the network ,
- Display Last-advertised channel for network,

- Display Number of clients (unique MACs) seen on network,
- Display Network status flags,
- Display extra AP information included by some manufacturers,
- Display number of LLC packets,
- Identifies manufacturer and model of AP and client ,
- Display maximum supported rate as advertised by AP ,
- Display name of the network or group,
- Display last seen noise level,
- Display last seen signal level,
- Display network type (Probe, *Ad hoc*, Infra, etc),
- Display WEP status ,
- Identifies unusual probes,
- Identifies excessive disassociation,
- Identifies Netstumbler probe requests,
- Identifies deauthenticate/disassociate Flood,
- Identifies Lucent link test,
- Identifies SSID brute force attempt,
- Identifies AP changing to a new channel,
- Identifies broadcast disconnect/deauthenticate ,
- Identifies invalid BSS timestamps,
- Identifies weak-iv packet,
- Identifies MITM attacks,
- Identifies MAC spoofing attacks.

B.4. Other unclassified Identifications

- Identifies vulnerability scans,
- Identifies port scans,
- Identifies incoming and outgoing connections that are ssh, identifies backdoors,
- Watch & record RPC services for specific, Application/procedure calls,
- Record and analyse email traffic,
- Track software versions,
- Identifies for blaster worm,
- Identifies synflood attacks,
- SSL analyser,
- Identifies clear text passwords,
- Promiscuous mode,
- Search for a pattern in the URI portion of a packet.

In summary, the identification ontology now can be composed by collaboration of Misuse and Anomaly ontologies that are part of the Hybrid ontology and can be formally described as follows:

A=A1 where A1= $B \cup C$

Figure 5.6 shows a final view of identification ontology.

5.1.2 Design of Response Subsystem Ontology

A. Features of Response Subsystem

One of the key important factors in designing a new ontology is efficiency in design. This can be achieved by splitting the ontology into several components of ontologies. As mentioned earlier, this is a "collaborative design". In collaborative design each component of ontologies will be built first, and then they all are compiled into a unified ontology. To accomplish this, it is necessary that every component of Response Subsystem Ontology (RSO) identified separately according to their domain or conceptual level before they are all compiled to realise a single unified RSO.

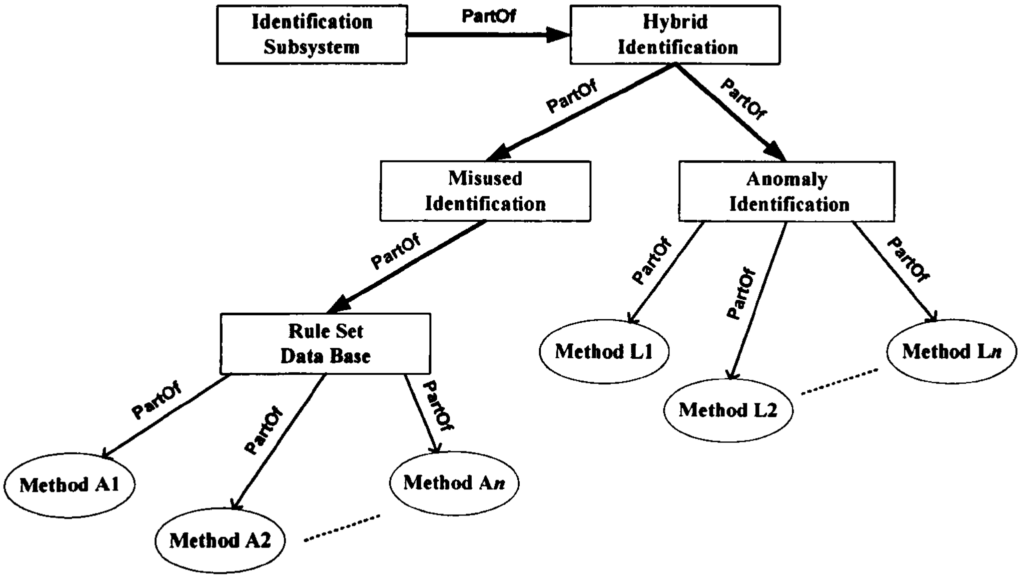


Figure 5.6. Final view of Identification Subsystem Ontology

The intrusion response subsystem is capable of stopping attacks against a given network and provides the following real-time defence mechanism:

- Prevention: it permanently stops detected misuse attacks from execution. In the case of an anomaly based attacks, it temporarily stops the detected attack from execution either automatically or via true management console. In such scenarios, anomaly and misuse, the IHS would send logs to a management console through the response subsystem.
- Reaction in anomaly based attacks: it immunises the system from future attacks from the same malicious source by storing the rule-set created by administrator so next time the same attack will be considered as a misuse attack.

For misuse identification, that is whenever a signature matches the anomaly identification (defending response), the response subsystem will automatically take direct action to prevent its execution. In addition, the IHS will establish an automatic prevention method. For anomaly identifications where response systems cannot automatically function, the IHS will establish a manual action method. This is referred to as informing or passive response [Sale2010b].

B. Concepts and Axioms of RSO

The attributes/slots, concepts/classes, possible constraints and values related to the response ontology are shown in Table 5.6.

Table 5.6. The concepts for response ontology and their facets

Ontology: Response Subsystem:				
Slot	**Type**	**Cardinality**	**Other Facets**	**Allowed Value**
Active	S	**Multiple**	**Class=Response**	**Various Methods**
Passive	S	**Multiple**	**Class=Response**	

The response subsystem may use different methods for each passive and active response. The following symbols are used for simplicity:

E: Response, F: Active, G: Passive

Using set theory this can be expressed as:

$F, G \subseteq E$ therefore:

$E=F\cup G$

The concepts F & G are subsets of response concepts. The axioms which response ontology follows are shown in Table 5.7.

Table 5.7. Axioms for response concept

ID	**Axioms**	**ID**	**Axioms**
1	partOf(Active, Response)	**2**	partOf(Passive, Response)
3	Composite (Active)	**4**	Composite (Passive)
5	$(\forall x, \exists y)$ Response (x) $\wedge$ Active (y) $\wedge$ partOf (Active, Response) $\vee$ True		
6	$(\forall x, \exists y)$ Response (x) $\wedge$ Passive (y) $\wedge$ partOf (Passive, Response) $\vee$ True		

The active concept can be expressed by the attributes/slots Method R1, Method R2 and Method R*n*. The different methods will be described later in this section. These attributes belong to the class active response as shown in Figure 5.7 [Sale2010b].

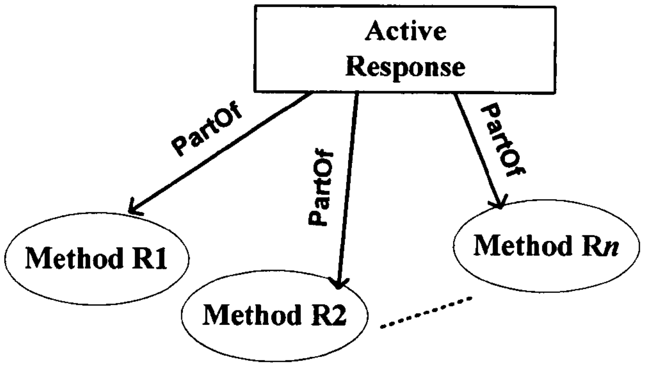


Figure 5.7. Active response concept

The active ontology can be decomposed into several slots. These slots are representing different methods. For simplicity the attributes are shown as follows:

f1: Method R1

f2: Method R2

⋮ ⋮

f*n*: Method R*n*

It can be noted that these slots are subsets of concept F, therefore:

f1, f2,....f*n* ⊆ F hence F = f1∪f2.... ∪f*n*

The axioms of active concepts are shown in Table 5.8 below.

Table 5.8. Axioms for active concept

ID	Axioms	ID	Axioms
1	partOf (Method R1, Active)	2	partOf (Method R2, Active)
3	partOf (Method R*n*, Active)	4	Primitive (Method R1)
5	Primitive (Method R2)	6	Primitive (Method R*n*)

5.1.2.1 Response Methods

In general possible reactions of active response methods in order to prevent intruder can be summarised as:

- Blocking Traffic: Denies traffic from the source address of the attack;
- Host shut down;
- Policy: Creation of Access Control List (ACL) Policy.

The passive concept can be expressed by the attributes/slots Method R'1, Method R'2 and Method R'*n*. The different method will be described later in this section. These attributes belong to the class passive response as shown in Figure 5.8 [Sale2010b].

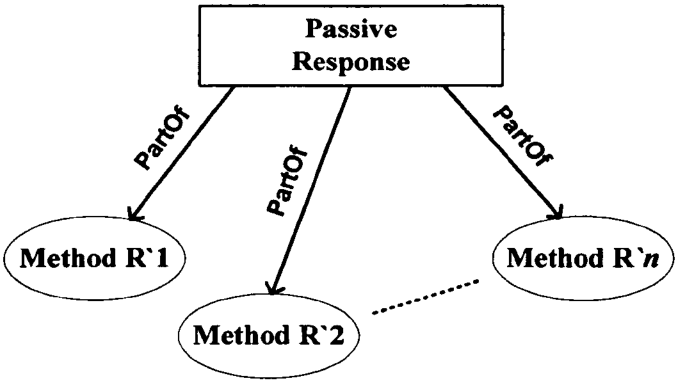


Figure 5.8. Passive response concept

The following shows the decomposition of passive response ontology into several slots. These slots are representing different methods. For simplicity the attributes are shown as follows:

g1: Method R'1
g2: Method R'2
⋮ ⋮
g*n*: Method R'*n*

It can be noted that these slots are subsets of concept G (passive response). Therefore: $g1, g2,....gn \subseteq G$ hence $G = g1 \cup g2.... \cup gn$.

The axioms of passive concepts are shown in Table 5.9 below.

Table 5.9. Axioms for passive concept

ID	Axioms	ID	Axioms
1	partOf (Method R`1, Passive)	2	partOf (Method R`2, Passive)
3	partOf (Method R`*n*, Passive)	4	Primitive (Method R`1)
5	Primitive (Method R`2)	6	Primitive (Method R`*n*)

In general the followings are the reaction of passive response methods (manual action methods) in order to prevent intrusion:

- Alerting administrator: Sends alarms to management console via response subsystem;
- Logging management console: Sends alarms to management console via response subsystem;
- Blocking traffic : Denies traffic from the source address of the attack;
- Host shut down;
- Policy: Creation of Access Control List (ACL) Policy.

In summary, the response ontology now can be composed by the collaboration of Active and Passive response ontologies and can be formally described as follows: $E=F\cup G$. Figure 5.9 shows a final view of the RSO.

5.1.3 Administrator/Manager Console Ontology

In chapter two, Figure 2.9 has illustrated the IHS entities and their relationships where an administrator has been defined as a person who sets up the network and is responsible for management, security and maintenance. An administrator is also responsible for managing operators. Although Figure 2.9 shows a separate block for the manager console and administrator but in fact, the administrator has full control over the manager/management console. Therefore for the sake of the ontology definition it has been assumed that the administrator and manager consoles are in the same block and they have the same responsibilities.

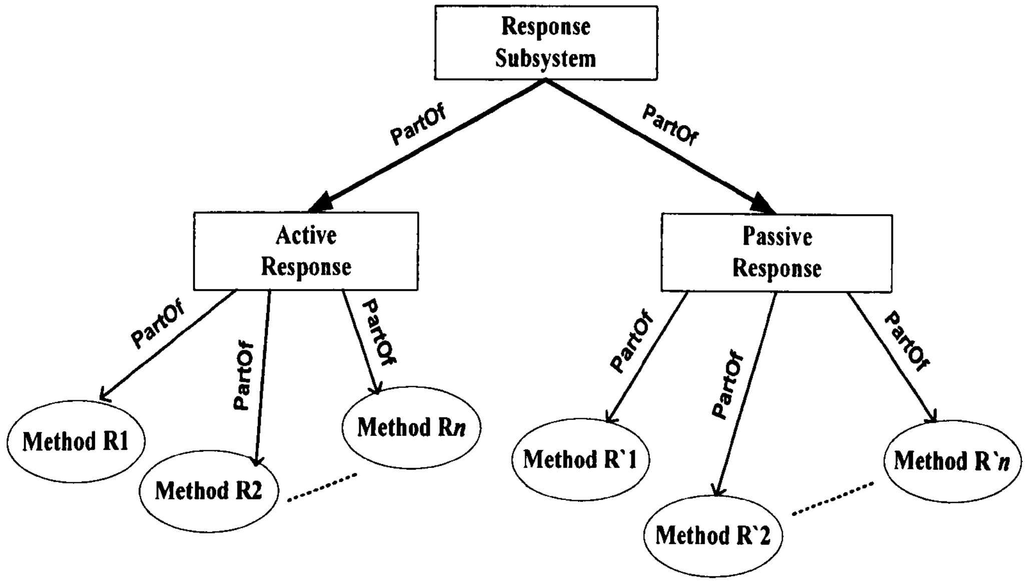


Figure 5.9. Final view of the RSO

The attributes/slots, concepts/classes, possible constraints and values related to Management Console Ontology (MCO) are shown in Table 5.10 [Sale2010b].

For simplicity the concepts and attributes/slots of MCO are denoted below. In future, there might be new attributes/slots become available but these will not affect the concept itself. For the MCO ontology, here it is assumed that there are currently only attributes/slots h1 to h4 are available for the concept. The same approach was applied to all ontologies and will be applied to the remaining ontologies in this chapter, hence:

H: Administrator/Manager Console

h1: Security Policy Database

h2: Updating Rule set Database

h3: Manual Prevention

h4: Network Monitoring

Table 5.10. The slots for ontology of administrator/ manager entity and facets for these slots

Ontology: Administrator/Manager Console				
Slot	**Type**	**Cardinality**	**Other Facets**	**Allowed Value**
Security Policy Data Base	E	Multiple	Class=Administrator	New Policy
Updating Rule Set Data Base	S	Multiple	Class=Administrator	New Signatures
Manual Prevention	E	Multiple	Class=Administrator	SNMP or Other similar Protocols
Network Monitoring	E	Single	Class=Administrator	

Based on the logic behind the previous ontologies, it can be noted that these attributes/slots are all subsets of H. Formally using set theory: $h1, h2, h3, h4 \subseteq H$
Since all attributes h1 to h4 are parts of concept H, it can be noted that:
$H = h1 \cup h2 \cup h3 \cup h4$.

The axioms of the administrator/manager console ontology are shown in Table 5.11 below.

Table 5.11. Axioms for administrator/manager console concept

ID	**Axioms**	**ID**	**Axioms**
1	PartOf (h1, H)	**2**	PartOf (h2, H)
3	PartOf (h3, H)	**4**	PartOf (h4, H)
5	Primitive (h1)	**6**	Primitive (h2)
7	Primitive (h3)	**8**	Primitive (h4)
9	$(\forall x, \exists y)$ H(x) $\wedge$ h1(y) $\wedge$ partOf(h1, H) $\vee$ True		
10	$(\forall x, \exists y)$ H(x) $\wedge$ h2(y) $\wedge$ partOf(h2, H) $\vee$ True		
11	$(\forall x, \exists y)$ H(x) $\wedge$ h3(y) $\wedge$ partOf(h3, H) $\vee$ True		
12	$(\forall x, \exists y)$ H(x) $\wedge$ h4(y) $\wedge$ partOf(h4, H) $\vee$ True		

Figure 5.10 shows final view of administrator/manager console ontology.

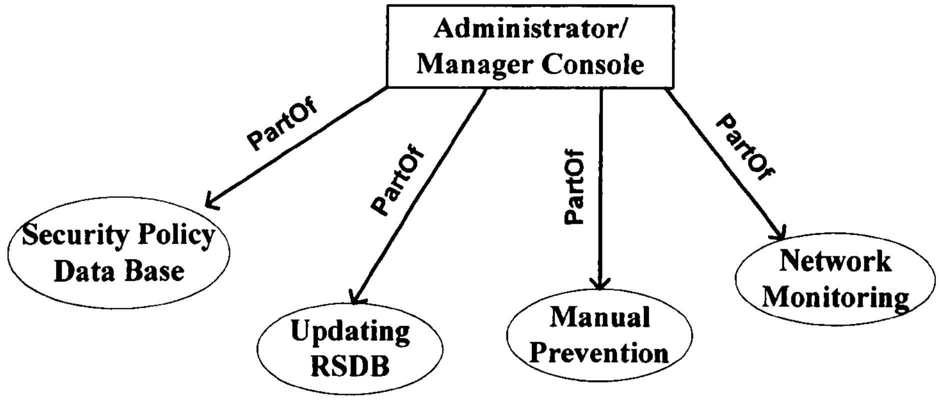


Figure 5.10. Final view of administrator/manager console ontology

5.1.4 Inter IHS Communication Subsystem Ontology

This ontology deals with communication between different IHS systems which are residing in different hosts. It is assumed that such systems are produced by different vendors and use different methods for identifications of and responses to intrusions. Therefore a Unified Intrusion Handling Report Format (UIHRF) should be considered to translate the non standard intrusion report formats from different type of IHSs to a UIHRF using IIHSM.

To exchange alerts from multiple IHS products with different output formats, first the identification subsystem needs to convert the diversified output format of these multiple products into a unified standard format. These unified messages are stored in unified Intrusion Handling Message Database (IHMD) within inter IHS communication module, as described before (Figure 4.1).

The IIHSM should be considered as a standard data format that automated intrusion handling systems can use to report alerts about events that they deem suspicious. The main reason for this is because of its compatibility with the developed new IHS reference model which should also run as efficient as possible in a host based *ad hoc* WLANs. The development of IIHSM standard format will enable interoperability among any types of wireless IHS systems including commercial, open source, and research systems. It will allow users to mix-and-match the deployment of their systems according to their strong and weak points to obtain an optimal implementation. The attributes/slots, concepts/classes, possible constraints and values related to Inter IHS Communication Subsystem (IICS) Ontology are shown in Table 5.12.

Table 5.12. the slots and concept for IICS ontology and their facets

Ontology: IICS				
Slot	**Type**	**Cardinality**	**Other Facets**	**Allowed Value**
IIHSM	**S**	**Single**	**Class=IICS**	**Wireless IDXP, etc**
IHMD	**S**	**Single**	**Class=IICS**	**N/A**
UIHRF	**S**	**Single**	**Class=IICS**	**N/A**

The concepts and slots are denoted as follows:

M: Inter IHS Communication Subsystem (IICS)

U: IIHSM

m1: IHMD

m2: UIHRF

It can be noted that these slots and concept U are all subsets of M. Formally using the set theory: U, m1, m2 $\subseteq$ M. Hence: M = U$\cup$m1$\cup$m2.

The axioms of IICS ontology is shown in Table 5.13 below. Figure 5.11 shows a basic view of IICS ontology without IIHSM attributes.

Table 5.13. Axioms for IICS ontology

ID	**Axioms**	**ID**	**Axioms**
1	PartOf (IIHSM, IICS)	**2**	PartOf (IHMD, IICS)
3	PartOf (UIHRF, IICS)	**4**	Composite (IIHSM)
5	Primitive (IHMD)	**6**	Primitive (UIHRF)
7	($\forall$x, $\exists$y) IICS(x) $\wedge$ IIHSM(y) $\wedge$ partOf(IIHSM, IICS) $\vee$ True		
8	($\forall$x, $\exists$y) IICS(x) $\wedge$ IHMD(y) $\wedge$ partOf(IHMD, IICS) $\vee$ True		
9	($\forall$x, $\exists$y) IICS(x) $\wedge$ UIHRF(y) $\wedge$ partOf(UIHRF, IICS) $\vee$ True		

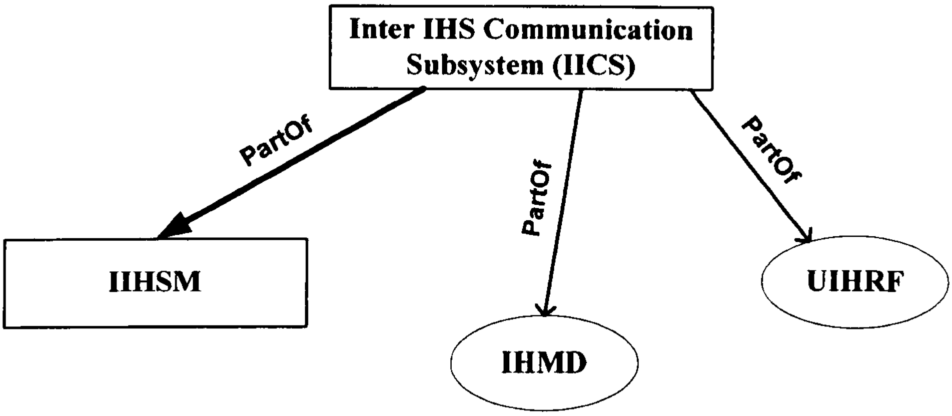


Figure 5.11. Basic view of IICS ontology

The IIHSM concept in turn can be expressed by the attributes/slots of Methods T1, Method T2 and Method T*n*. The various methods or in this case the attributes will be described later in the next section .These attributes belong to the class IIHSM as shown in Figure 5.12 [Sale2010b].

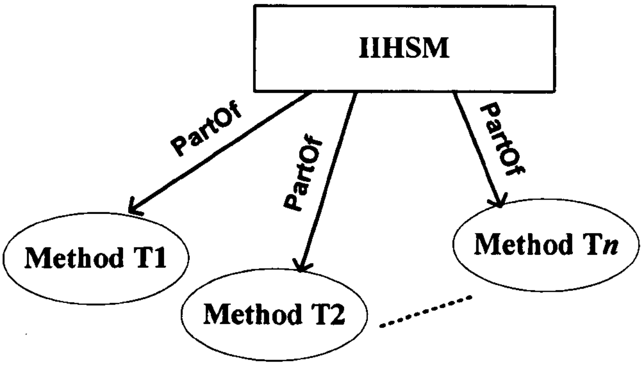


Figure 5.12. IIHSM concept

The following notations are used to represent slots of IIHSM concept:

u1: Method T1

u2: Method T2

⋮ ⋮

u*n*: Method T*n*

It can be noted that these slots are subsets of concept U, that is:

u1 $\subseteq$U, u2 $\subseteq$ U,…., u*n* $\subseteq$ U. Hence formally : U= u1$\cup$u2…..$\cup$u*n*.

Table 5.14 shows the axioms abide by the IIHSM concept.

Table 5.14. Axioms for IIHSM concept

ID	Axioms	ID	Axioms
1	partOf (Method T1, IIHSM)	2	partOf (Method T2, IIHSM)
3	partOf (Method T*n*, IIHSM)	4	Primitive (Method T1)
5	Primitive (Method T2)	6	Primitive (Method T*n*)

In summary, The IICS ontology now can be composed by collaboration of IIHSM ontology and can be formally described as $U \subseteq M$ therefore:

$M = m1 \cup m2 \cup u1 \cup u2 \ldots\ldots \cup un$.

Figure 5.13 shows a final view of IICS ontology with IIHSM attributes.

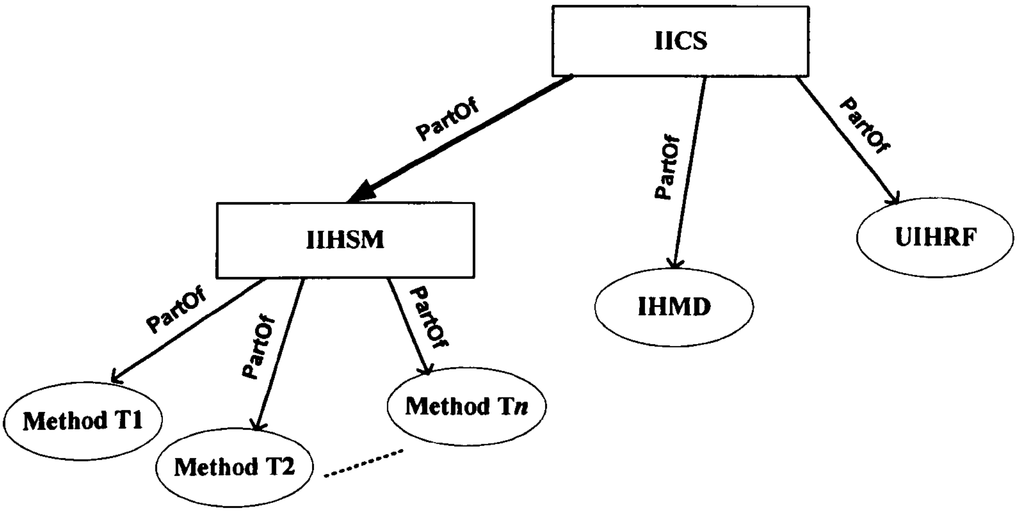


Figure 5.13. Final view of IICS ontology

It should be assumed that IIHSM is protocol independent. This means that methods used in IIHSM ontology can be considered as any type of compatible wireless inter IHS communication protocols including but not limited to Wireless format of Intrusion Detection Exchange Protocol IDXP (RFC4767). This has also been discussed earlier and Figure 4.2 has illustrated the relations between the RFC 4765, 4766 and 4767 with a comparison to IIHSM.

5.1.5 IHS Data/Traffic Source Ontology

The IHS data/traffic source in Figure 2.9 is represented by the IHS data source ontology. This data source refers to the Libpcap packet capture library (Figure 2.4) that resides in the application layer of OSI model. As shown in Figure 2.4 earlier, Libpcap packet capture library was developed by LBNL's Network Research Group. Pcap is an application programming interface for packet capturing. The implementation of Pcap for Unix based systems is known as libpcap. The windows version of libpcap is called WinPcap and it may be used by a program to capture packets travelling over a network. Libpcap and WinPcap are the packet capture and filtering engines of many open source and commercial network tools, including packet sniffers, traffic generators, network testers, protocol analysers, network monitors and network intrusion handling systems. Packet filtering does not occur in Libpcap in the comparison based IHS model in Figure 2.4; instead a packet decoder would filter the packets based on their level of risk. The pre-processor decodes the raw Data Link layer packets. They recognise different protocols, including Ethernet, 802.11, Token Ring, and other higher layer protocols such as IP, TCP, and UDP. In Figure 2.4, the pre-processor arranges or modifies data packets before the identification system carries out some operation to find out if the packet is being used by an intruder. The pre-processor is very important for IHS because it prepares data packets to be analysed against rules in the identification system [Sale2010b].

The attributes/slots, concepts/classes, possible constraints and values related to the Traffic/Data Source ontology are shown in Table 5.15.

Table 5.15. the slots and concept for IHS data source ontology and their facets

Ontology: Traffic/Data Source				
Slot	**Type**	**Cardinality**	**Other Facets**	**Allowed Value**
Libpcap	**E**	**Multiple**	**Class=Data Source**	**N/A**
Winpcap	**E**	**Multiple**	**Class=Data Source**	**Level of risk,**
Filtering System	**S**	**Single**	**Class=Data Source**	**802.11 Protocols, IP,**
Pre-processor	**S**	**Single**	**Class=Data Source**	**TCP,UDP, etc**

For simplicity the concepts and slots are denoted as:

Q: Traffic/Data Source (DS)

R: Libpcap /Winpcap (CAP)

r1: Filtering System (FS)

r2: Pre Processor (PP)

It can be noted that these slots and concept R and slots r1 and r2 are all subset of Q. Formally using the set theory: r1, r2 $\subseteq$ R and R $\subseteq$ Q. Hence: R, r1, r2 $\subseteq$ Q and therefore formally: Q= R$\cup$r1$\cup$r2. The axioms of IHS data source ontology are shown in Table 5.16 below. Figure 5.14 shows IHS data source ontology including its concept and relative attributes.

Table 5.16. Axioms for IHS data source ontology

ID	Axioms	ID	Axioms
1	PartOf (CAP, DS)	2	PartOf (FS,DS)
3	PartOf (PP, DS)	4	Composite (CAP)
5	Primitive (FS)	6	Primitive (PP)
7	$(\forall x, \exists y)$ TS(x) $\wedge$ CAP(y) $\wedge$ partOf(CAP, TS) $\vee$ True		
8	$(\forall x, \exists y)$ TS(x) $\wedge$ FS(y) $\wedge$ partOf(FS, TS) $\vee$ True		
9	$(\forall x, \exists y)$ TS(x) $\wedge$ PP(y) $\wedge$ partOf(PP, TS) $\vee$ True		

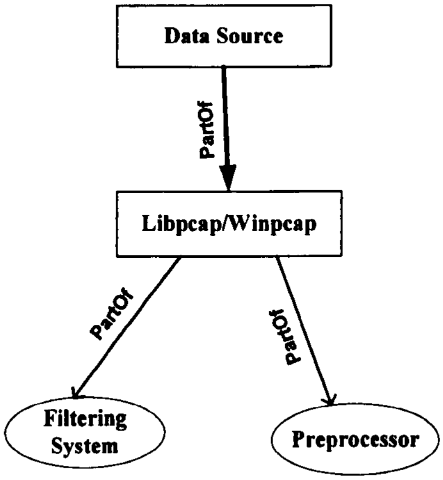


Figure 5.14. IHS data source ontology

5.1.6 IHS Modelling Ontology as a Whole

The new reference model for IHS consists of data source, administrator/manager console, identification and response subsystems and inter IHS communications subsystem and other related components and information. A final ontological representation of this new IHS domain in Figure 5.15 is composed by collaboration of ontologies of all the components described earlier in this chapter. This IHS model can be represented as:

IHS model = $Q \cup H \cup A \cup E \cup M$

Hence an IHS model is:

$(\forall x\text{: IHS})(\exists Q\text{: Traffic Source})(\exists H\text{: Admin/manager console})(\exists A\text{: Identification Subsystem})(\exists E\text{: Response Subsystem})(\exists M\text{: Inter IHS communication Subsystem})(\text{partOf}(Q,x) \wedge \text{partOf}(H,x) \wedge \text{partOf}(A,x) \wedge \text{partOf}(E,x) \wedge \text{partOf}(M,x))$

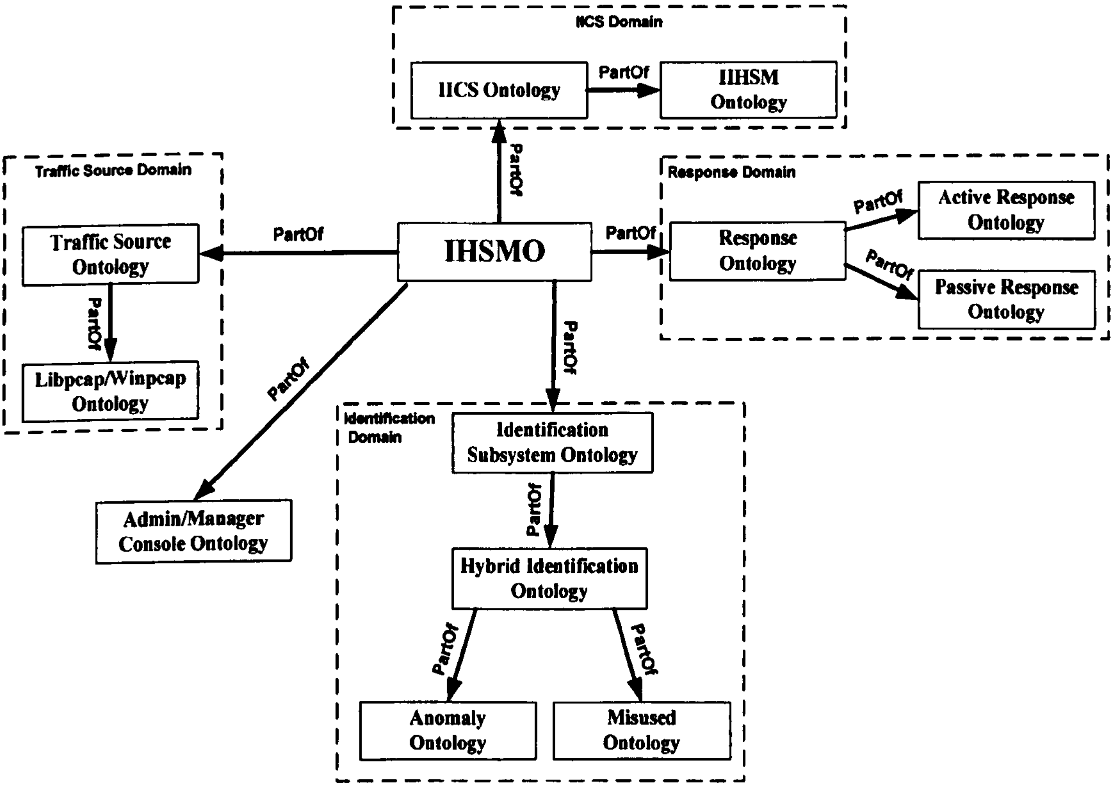


Figure 5.15. Final view of IHSMO

The domains mentioned in Figure 5.15 are defined as follows.

- Traffic Source Domain:

$(\forall Q: \text{Traffic Source})\ (\exists R: \text{Libpcap/Winpcap}) \wedge (\text{partOf}\,(R,Q))$

- Identification Domain:

$(\forall A: \text{Identification Subsystem})(\exists A1: \text{Hybrid})(\exists B: \text{Misuse})(\exists C: \text{Anomaly})\ (\text{partOf}\,(A1,A)) \wedge (\text{partOf}\,(B,A)) \wedge (\text{partOf}\,(C,A))$

- Response Domain:

$(\forall E: \text{Response Subsystem})(\exists F: \text{Active})(\exists G: \text{Passive})\ ((\text{partOf}\,(F,E)) \wedge (\text{partOf}\,(G,E))$

- IICS Domain:

$(\forall M: \text{IICS})\ (\exists N: \text{IIHSM}) \wedge (\text{partOf}\,(N,M))$

5.2 Ontologies Based Research Activities using Ontology Editorial Tools

There are not any records of any research activities similar to our approach in the IHS subject area. However, the following shows some examples of ontology research, in different subject areas:

- "Conceptual Level Programming and Task Ontology" [Mizo1992] the major objectives of MULTIS project is to bridge the gap between domain experts and computers to enable computers to elicit domain experts' ways of problem solving using task ontology. This can be interpreted in another way: MULTIS can help an end user describe how they perform a task at the conceptual level without considering how computer works. This interpretation creates a new idea that "Conceptual level programming supported by task ontology" which is an advancement of utility of task ontology research [Seta1996].

- "Ontologies of Function and Behaviour and Explanation Generation" [Sasa1995]. This research is motivated by a strong desire to know what function is and what behaviour is. In spite of the long history of the research in this subject, there is still no satisfactory model of them that has been obtained. Needless to say, well-established understanding of them is indispensable to qualitative modelling, and hence model-based problem solving [Mizo1997].

- "Ontology of Time and Causality of Fluid for Qualitative Simulation" [Kita1996] Qualitative simulation based on sophisticated ontology of causality is indispensable for model-based problem solving. This research is deeply concerned with ontology of time and causality. Qualitative simulation has a different ontology from the real time but it is not so clear how different they are [Mizo1997]. Other areas of research are "Communication Network Ontology" [Jame2006], which provides a definition of terminology related to the telephone communications system, radio systems, television broadcast, etc. However they seem very general. Basic computing terminology (e.g. computer, CPU, Hard Disk Drive, Computer Process, etc) along with a few communications terms (e.g. LAN, Data Transfer, IP address, etc) Also, a method of developing an ontology for electrical network applications and its reuses have been described in [Bern1996]. Ontology of an electrical distribution network domain has been designed and is described in [Dali1997]. It is vital to mention that there are several ontology development tools and ontology representation languages available for ontology researchers.

5.2.1 Ontology Languages

The most popular existing languages for representing ontologies are the following:

- Knowledge Interchange Format (KIF) [Gene] which was one of the first knowledge representation languages and it is a computer-oriented language for the interchange of knowledge among disparate programs.

- Open Knowledge base Connectivity (OKBC) provides a uniform model of Knowledge Representation Systems (KRSs) based on a common conceptualisation of classes, individuals, slots, facets, and inheritance. The Generic Frame Protocol (GFP) Knowledge Model is the implicit representation formalism underlying OKBC. OKBC is defined in a programming language independent fashion, and has existing implementations in Common Lisp, Java, and C.

- Resource Description Framework (RDF) [Klyn2004] was designed to describe metadata for the resources on the web in means of "statements", "resources" and "properties". Statements in RDF describe resources that can be web pages or real world object like publications, persons or institutions.

- XOL stands for XML-based Ontology exchange Language [Karp1999]. It was designed for the exchange of bioinformatics ontologies but it could be applied to different domains. The language was dedicated to the exchange of ontology definitions among different systems like database systems, ontology development tools or application programs.

- Ontology Interference Layer (OIL) [Fens2000] is a proposal for a joint standard for specifying and exchanging ontologies. Its definition is based on existing frame-based language such as OKBC, XOL and RDF [Ribi2000].

- The DAML project (Darpa Agent Markup Language) proposed a first release of an ontology language called DAML-Ont. After discussing differences between this language and the OIL proposition, two projects merged to propose DAML + OIL which is a language based on RDF and RDF Schema with richer modelling primitives. DAML+OIL provide modelling primitives commonly found in frame-based languages. The language has a clean and well defined semantics based on description logics.

Other ontology languages include: Telos [Mylo1990], LOOM [MacG1991], Operational Concept Modelling Language (OCML) [Shad1993], Concept Modelling Language (CML) [Schr1994], Cycl, Ontolingua, Frame Logic (F-Logic) [Kife1995], Simple HTML Ontology Extensions (SHOE) [Luke2000],CYC [Lena1990, Lena1995], Wordnet [Mill1990], Gene Ontology [Clar2004] and Standard Upper Ontology [Jame2006].

5.2.2 Ontology Tools

Ontology design is a creative process and ontologies designed by different people would be different. The potential applications of the ontology and the designer's understanding and view of the domain will undoubtedly affect ontology design choices [Noy2001].

There are many software tools available to accomplish most aspects of ontology design and development. Many of those are in use today but the most common and widely used tools are: Protégé-2000 [Gross1999], OntoEdit [Maed2000], WebOnto [April2001], WebODE [Domi1999], OilED [Bech2001] and Hozo [Mizo2004]. A detail comparison table of all existing ontology tools can be found in (http://www.xml.com/lpt/a/1061).

The developed ISHMO ontology can be expressed using languages discussed in section 5.2.1 or it can also be represented/edited by the above tools. However, sharing vocabulary is a main step towards unification, but it is not in itself sufficient because of the following reasons:

- Knowledge changes and updates itself.
- Individual ontology researches evaluate data differently.

Therefore, IHSMO is not an absolute and definite way of building a unified IHS ontology model because ontology developers may use their own view and approach in order to define the elements of IHSMO. Because of this, in the next chapter the IHSMO will be evaluated to show the quality of its concepts, attributes, axioms and other elements. A successful evaluation of IHSMO can prove its uniqueness and robustness as a reference model to develop future or to evaluate the existing IHS systems.

To summarise, in this section an ontology-development methodology approach has been used to construct, design and develop the new IHS Modelling Ontology (ISHMO). The development of IHSMO was based on Figure 2.8, the Architecture of IHS Reference Model, Figure 2.9, IHS entities and relationships, some of the existing logics, concepts and functions found in the legacy IDS/IPS software tools (Bro, Snort, Kismet, etc) and using the logics and concepts behind the structure of some ontology tools (Protégé-2000, Ontolingua ,Onto Edit and Hozo). The first and higher-order logics are used for defining the terminology of the IHSMO which gives a precise and unambiguous semantics for each term. The precision helps to avoid possible conflicts and different interpretations by different ontology researchers/designers.

5.2.3 Hozo Ontology editor terminologies

Hozo is an ontology editor tool and it is an environment for Developing/building and using a new ontology. The following is the terminology used in Hozo ontology editor tool.

5.2.3.1 Wholeness and Relational Concepts

The ontology system handles two concepts individually [Mizo2004]:

- Wholeness concept: A concept of a thing considered as the whole, which is composed of multiple concepts, each of which makes up a part of the thing.
- Relational concept: Conceptualised relationship between multiple concepts.

5.2.3.2 Node

In this system, the unit to represent a class/concept is called "node". In [Mizo2004] the class/concept (=node) has been defined as:

- Labels describing the concept.
- Super ordinate concept.
- Definition by natural language (description).
- Partial Concept: Partial concept described by the "part-of" relationship.
- Slot: slots described by the "part-of" relationship.
- Axiom.

5.2.3.3 Slot

In this system, the unit to represent a partial concept or attribute is called "slot". The types of relationships between partial concepts (attributes) which are handled in this system (slot) are shown by "partOf" relationship. The definitions of the partial concept (=slot) given below [Mizo2004] as:

- Role concept: A concept represented by partial concepts from the viewpoint of the whole.

- Class constraint: A constraint of the class to which the entity that bears the role defined by the role concept belongs.
- Role holder: A concept that represents the basic concept of the state bearing the role defined by the role concept.
- Definition by natural language (description of the content).
- Axiom

5.2.3.4 Link

In this system, the unit to represent a relationship between class/concept is called "link". The types of relationships between classes/concepts which are handled in this system (link) are also outlined below [Mizo2004]:

- *is-a* relationship
- Relationship based on the relational concept

5.2.4 Development of IHSMO using Hozo

Because of the above described features, it is concluded that Hozo ontology editor is suitable for building IHSMO which allows both specification and verification of the integrity of the developed model. Hozo ontology editor has three different outputs for the developed IHSMO. One is the graphical layout of the IHSMO consisting concepts, slots, and the relationship between them as shown in Figure 5.16.[2] Second one is the XML source code of IHSMO in text format. The third one is the user friendly text format of converted XML code. Appendix E contains the second and third outputs.

The Figure 5.16 also shows that the Libpcap/Winpcap is used as capturing system for incoming traffics both for *NIX and Windows based operating systems respectively. The filtering system uses the pre-processor mechanism to filter the traffic before entering the IHS system. In this model, there is no packet filtering occurs in Libpcap/Winpcap; instead packet decoder filters the packets based on their level of risk. The pre-processor decodes the raw Data Link layer packets as well as higher layer protocols such as IP, TCP, and UDP. The pre-processor

[2] The electronic version of Figure 5.16 consists of 6 separate pages because of the size of Hozo output. However, the printed version additionally contains a foldable full version of the all-in-one chart.

arranges or modifies data packets before the identification subsystem execute some operation to find out if the packet is being used by an intruder. The pre-processor is very important for IHS to prepare data packets to be analysed against rules in the identification subsystem. Figures for IHSMO without slots can be found in Appendix E1 and E2.

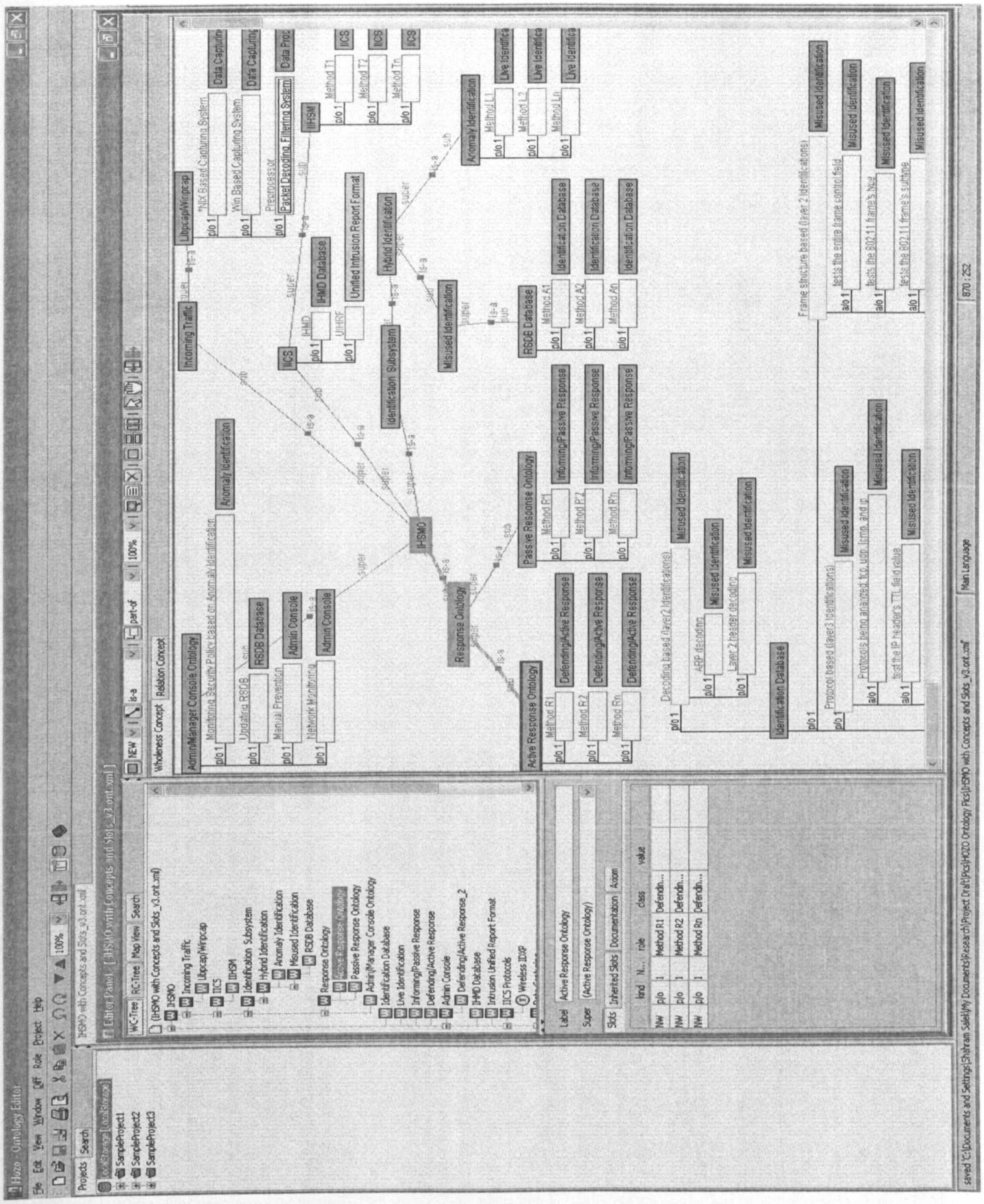

Hozo - Ontology Editor
File Edit View Window Diff Role Project Help
Projects Search
SampleProject1
SampleProject2
SampleProject3
Editor Panel - [IHSMO with Concepts and Slots_v3.ont.xml]
WC-Tree RC-Tree Map View Search
IHSMO
Incoming Traffic
Libpcap/Winpcap
IICS
IIHSM
Identification Subsystem
Hybrid Identification
Anomaly Identification
Misused Identification
RSDB Database
Response Ontology
Active Response Ontology
Passive Response Ontology
Admin/Manager Console Ontology
Identification Database
Live Identification
Informing/Passive Response
Defending/Active Response
Admin Console
IHMD Database
Intrusion Unified Report Format
IICS Protocols
Wireless IDXP
Label Active Response Ontology
Super (Active Response Ontology)
Slots Inherited Slots Documentation Axiom
kind N... role class value
p/o 1 Method R1 Defendin...
p/o 1 Method R2 Defendin...
p/o 1 Method Rn Defendin...
Wholeness Concept Relation Concept
NEW is-a part-of 100%
Admin/Manager Console Ontology
Monitoring Security Policy based on Anomaly Identification
Anomaly Identification
Updating RSDB
RSDB Database
Manual Prevention
Admin Console
Network Monitoring
Admin Console
Incoming Traffic
Libpcap/Winpcap
*NIX Based Capturing System
Data Capturing
Win Based Capturing System
Data Capturing
Preprocessor
Packet Decoding, Filtering System
Data Proc
IICS
IHMD
IHMD Database
UIHRF
Unified Intrusion Report Format
IIHSM
Method T1
Method T2
Method Tn
IICS
Identification Subsystem
Hybrid Identification
Anomaly Identification
Method L1
Method L2
Method Ln
Live Identification
Misused Identification
RSDB Database
Method A1
Method A2
Method An
Identification Database
IHSMO
Response Ontology
Passive Response Ontology
Method R1
Method R2
Method Rn
Informing/Passive Response
Active Response Ontology
Defending/Active Response
Identification Database
Decoding based (layer2 identifications)
ARP decoding
Layer 2 header decoding
Misused Identification
Protocol based (layer3 identifications)
Protocols being analyzed: tcp, udp, icmp, and ip
tests the IP header's TTL field value
Frame structure based (layer 2 identifications)
tests the entire frame control field
tests the 802.11 frame's type
tests the 802.11 frame's subtype
Misused Identification
is-a
super
sub
p/o 1
a/o 1
saved 'C:\Documents and Settings\Shahram Salek\My Documents\Research\Project Draft\Pics\HOZO Ontology Pics\IHSMO with Concepts and Slots_v3.ont.xml'
Main Language
870 : 252

Figure 5.16. IHSMO using Hozo editor (1/6)

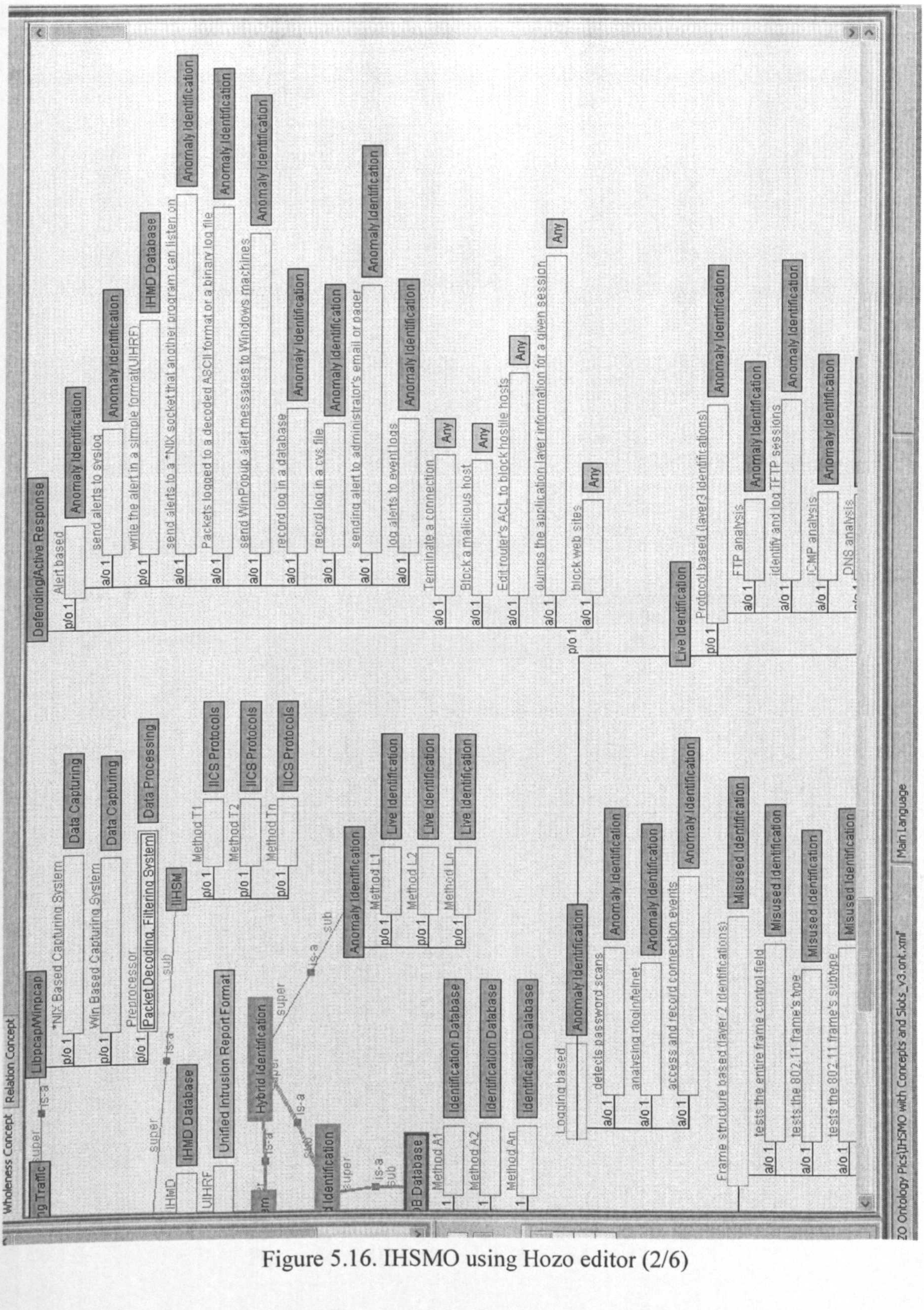


Figure 5.16. IHSMO using Hozo editor (2/6)

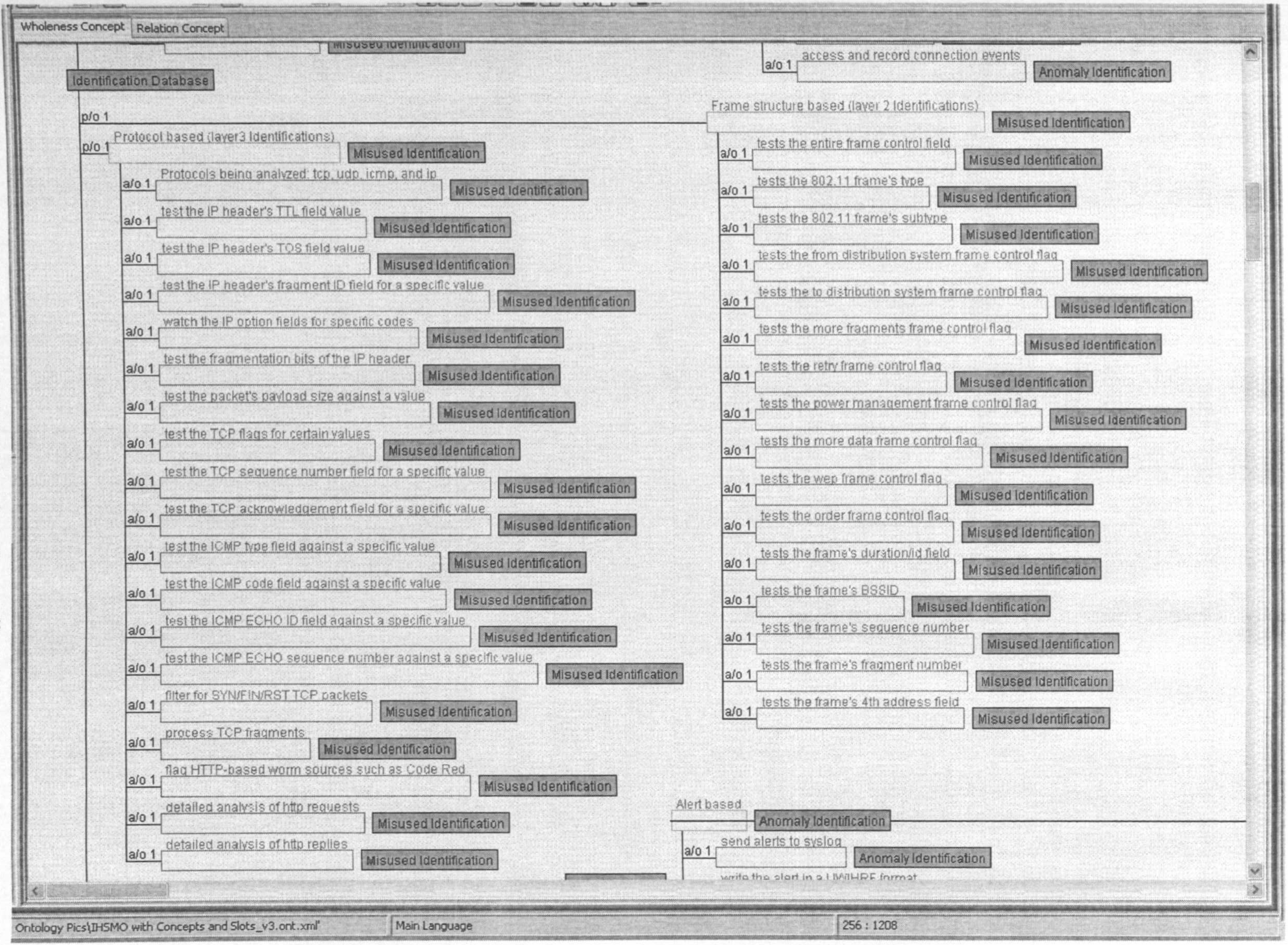
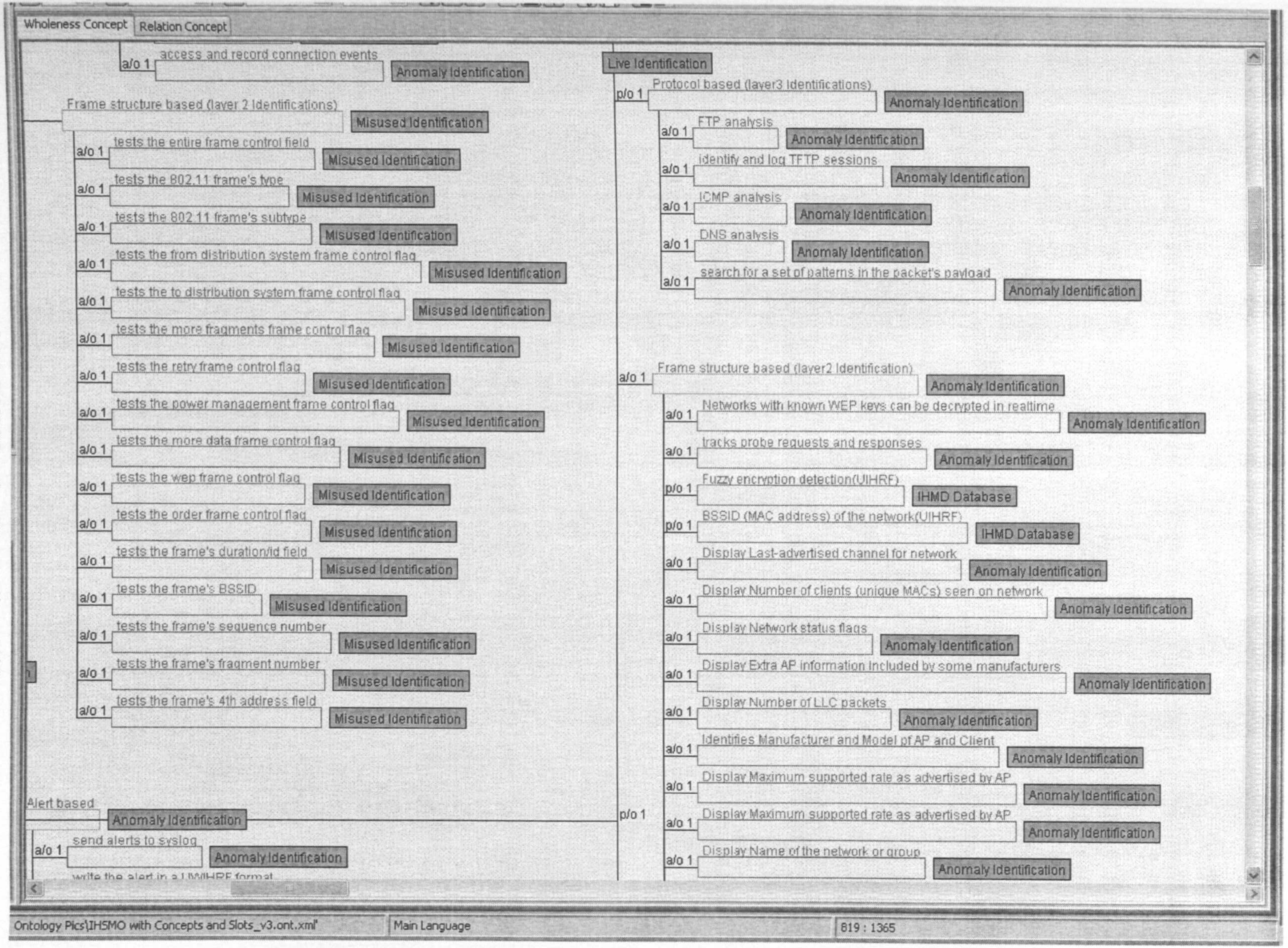

Wholeness Concept
Relation Concept
Misused Identification
Identification Database
p/o 1
p/o 1
Protocol based (layer3 Identifications)
Misused Identification
a/o 1
Protocols being analyzed: tcp, udp, icmp, and ip
Misused Identification
a/o 1
test the IP header's TTL field value
Misused Identification
a/o 1
test the IP header's TOS field value
Misused Identification
a/o 1
test the IP header's fragment ID field for a specific value
Misused Identification
a/o 1
watch the IP option fields for specific codes
Misused Identification
a/o 1
test the fragmentation bits of the IP header
Misused Identification
a/o 1
test the packet's payload size against a value
Misused Identification
a/o 1
test the TCP flags for certain values
Misused Identification
a/o 1
test the TCP sequence number field for a specific value
Misused Identification
a/o 1
test the TCP acknowledgement field for a specific value
Misused Identification
a/o 1
test the ICMP type field against a specific value
Misused Identification
a/o 1
test the ICMP code field against a specific value
Misused Identification
a/o 1
test the ICMP ECHO ID field against a specific value
Misused Identification
a/o 1
test the ICMP ECHO sequence number against a specific value
Misused Identification
a/o 1
filter for SYN/FIN/RST TCP packets
Misused Identification
a/o 1
process TCP fragments
Misused Identification
a/o 1
flag HTTP-based worm sources such as Code Red
Misused Identification
a/o 1
detailed analysis of http requests
Misused Identification
a/o 1
detailed analysis of http replies
Misused Identification
a/o 1
access and record connection events
Anomaly Identification
Frame structure based (layer 2 Identifications)
Misused Identification
a/o 1
tests the entire frame control field
Misused Identification
a/o 1
tests the 802.11 frame's type
Misused Identification
a/o 1
tests the 802.11 frame's subtype
Misused Identification
a/o 1
tests the from distribution system frame control flag
Misused Identification
a/o 1
tests the to distribution system frame control flag
Misused Identification
a/o 1
tests the more fragments frame control flag
Misused Identification
a/o 1
tests the retry frame control flag
Misused Identification
a/o 1
tests the power management frame control flag
Misused Identification
a/o 1
tests the more data frame control flag
Misused Identification
a/o 1
tests the wep frame control flag
Misused Identification
a/o 1
tests the order frame control flag
Misused Identification
a/o 1
tests the frame's duration/id field
Misused Identification
a/o 1
tests the frame's BSSID
Misused Identification
a/o 1
tests the frame's sequence number
Misused Identification
a/o 1
tests the frame's fragment number
Misused Identification
a/o 1
tests the frame's 4th address field
Misused Identification
Alert based
Anomaly Identification
a/o 1
send alerts to syslog
Anomaly Identification
write the alert in a UWIHRF format
Ontology Pics\IHSMO with Concepts and Slots_v3.ont.xml'
Main Language
256 : 1208

Figure 5.16. IHSMO using Hozo editor (3/6)



Wholeness Concept | Relation Concept

- a/o 1 access and record connection events — Anomaly Identification
- Frame structure based (layer 2 Identifications) — Misused Identification
  - a/o 1 tests the entire frame control field — Misused Identification
  - a/o 1 tests the 802.11 frame's type — Misused Identification
  - a/o 1 tests the 802.11 frame's subtype — Misused Identification
  - a/o 1 tests the from distribution system frame control flag — Misused Identification
  - a/o 1 tests the to distribution system frame control flag — Misused Identification
  - a/o 1 tests the more fragments frame control flag — Misused Identification
  - a/o 1 tests the retry frame control flag — Misused Identification
  - a/o 1 tests the power management frame control flag — Misused Identification
  - a/o 1 tests the more data frame control flag — Misused Identification
  - a/o 1 tests the wep frame control flag — Misused Identification
  - a/o 1 tests the order frame control flag — Misused Identification
  - a/o 1 tests the frame's duration/id field — Misused Identification
  - a/o 1 tests the frame's BSSID — Misused Identification
  - a/o 1 tests the frame's sequence number — Misused Identification
  - a/o 1 tests the frame's fragment number — Misused Identification
  - a/o 1 tests the frame's 4th address field — Misused Identification
- Alert based — Anomaly Identification
  - a/o 1 send alerts to syslog — Anomaly Identification
  - write the alert in a UIWIHRF format

Live Identification

- p/o 1 Protocol based (layer3 identifications) — Anomaly Identification
  - a/o 1 FTP analysis — Anomaly Identification
  - a/o 1 identify and log TFTP sessions — Anomaly Identification
  - a/o 1 ICMP analysis — Anomaly Identification
  - a/o 1 DNS analysis — Anomaly Identification
  - a/o 1 search for a set of patterns in the packet's payload — Anomaly Identification
- a/o 1 Frame structure based (layer2 Identification) — Anomaly Identification
  - a/o 1 Networks with known WEP keys can be decrypted in realtime — Anomaly Identification
  - a/o 1 tracks probe requests and responses — Anomaly Identification
  - p/o 1 Fuzzy encryption detection(UIHRF) — IHMD Database
  - p/o 1 BSSID (MAC address) of the network(UIHRF) — IHMD Database
  - a/o 1 Display Last-advertised channel for network — Anomaly Identification
  - a/o 1 Display Number of clients (unique MACs) seen on network — Anomaly Identification
  - a/o 1 Display Network status flags — Anomaly Identification
  - a/o 1 Display Extra AP information Included by some manufacturers — Anomaly Identification
  - a/o 1 Display Number of LLC packets — Anomaly Identification
  - a/o 1 Identifies Manufacturer and Model pf AP and Client — Anomaly Identification
  - a/o 1 Display Maximum supported rate as advertised by AP — Anomaly Identification
  - a/o 1 Display Maximum supported rate as advertised by AP — Anomaly Identification
  - a/o 1 Display Name of the network or group — Anomaly Identification
- p/o 1

Ontology Pics\IHSMO with Concepts and Slots_v3.ont.xml' | Main Language | 819 : 1365

Figure 5.16. IHSMO using Hozo editor (4/6)

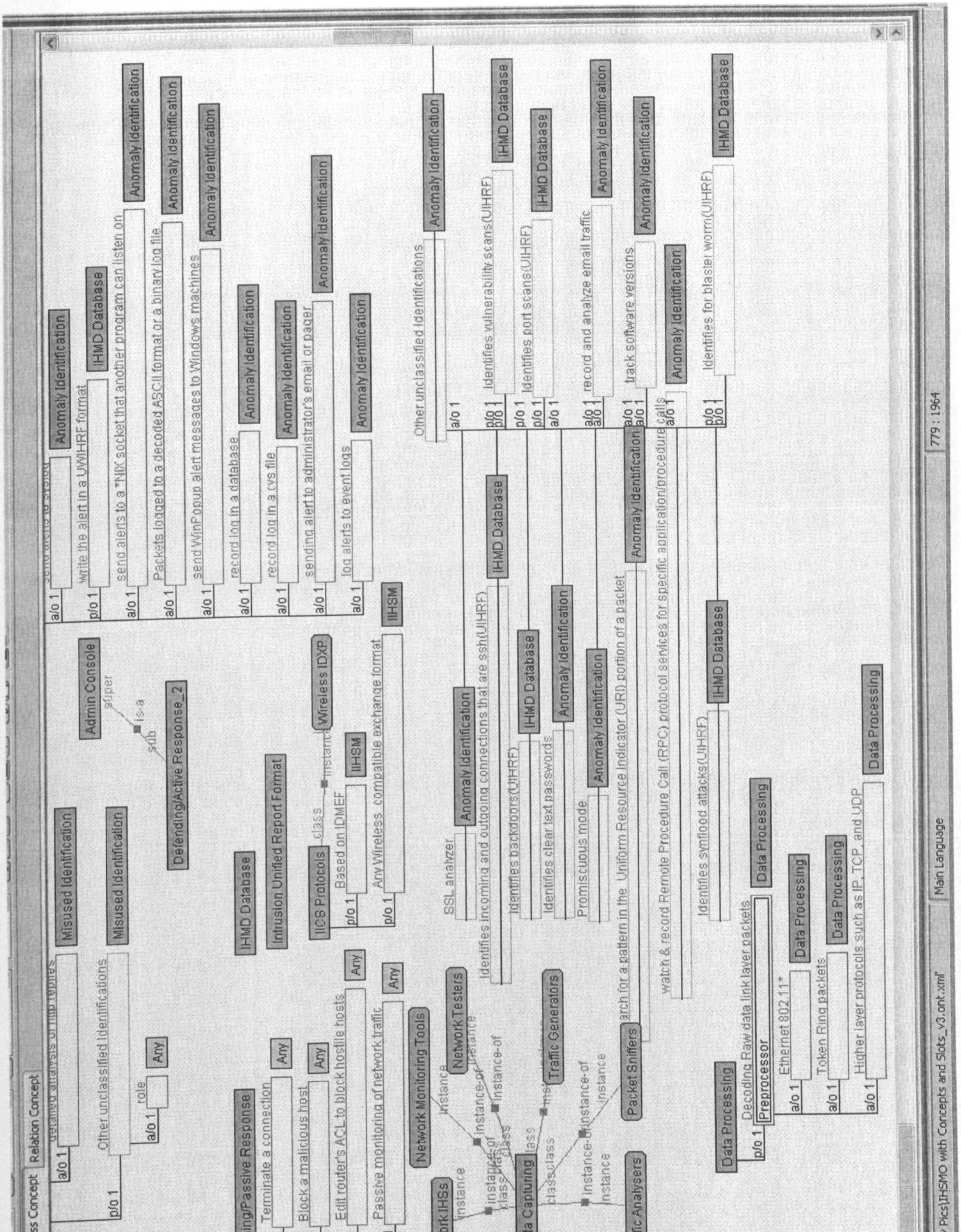


Figure 5.16. IHSMO using Hozo editor (5/6)

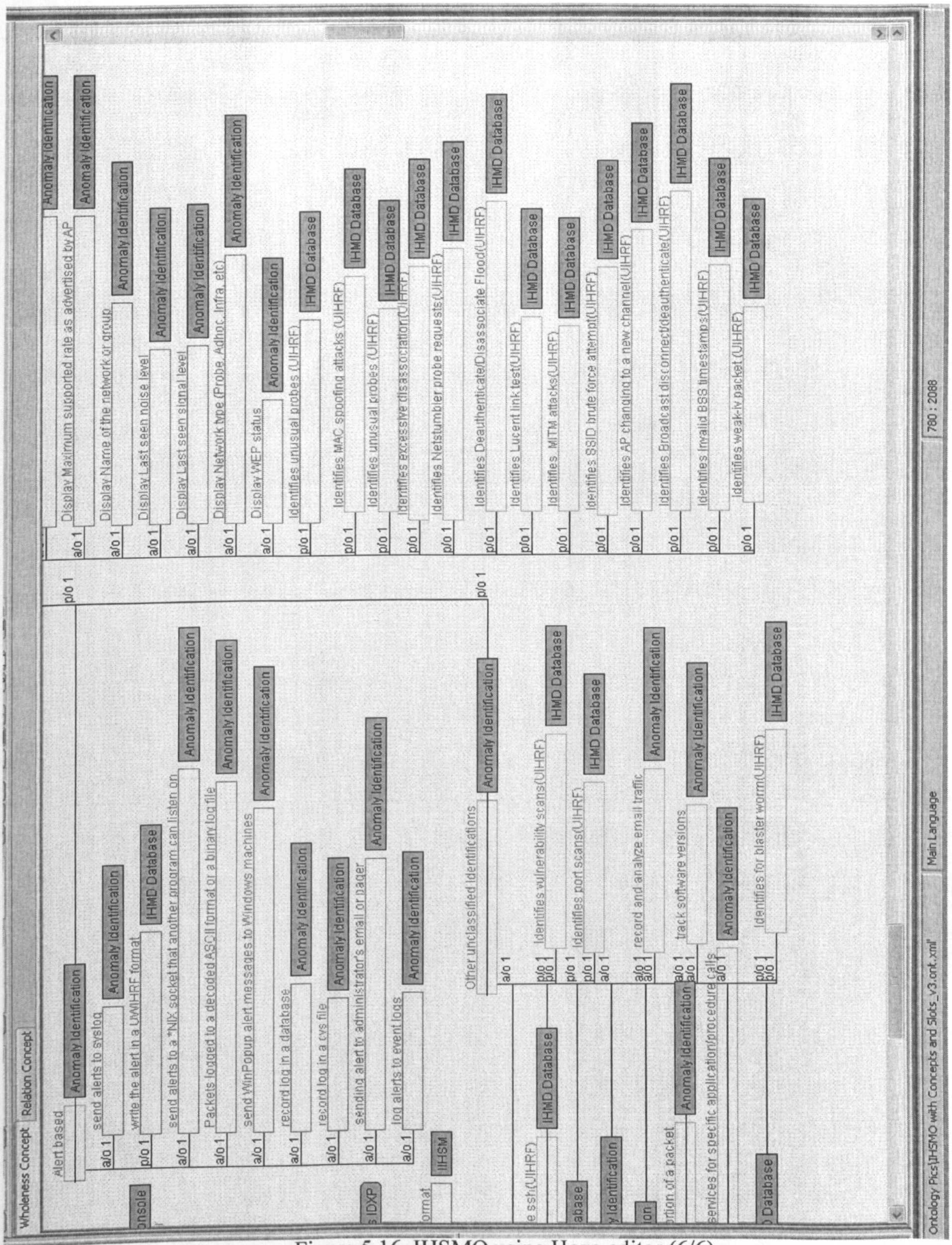

Wholeness Concept
Relation Concept
onsole
s IDXP
ormat
IHSM
Alert based
Anomaly Identification
a/o 1
send alerts to syslog
Anomaly Identification
p/o 1
write the alert in a UWIHRF format
IHMD Database
a/o 1
send alerts to a *NIX socket that another program can listen on
Anomaly Identification
a/o 1
Packets logged to a decoded ASCII format or a binary log file
Anomaly Identification
a/o 1
send WinPopup alert messages to Windows machines
Anomaly Identification
a/o 1
record log in a database
Anomaly Identification
a/o 1
record log in a cvs file
Anomaly Identification
a/o 1
sending alert to administrator's email or pager
Anomaly Identification
a/o 1
log alerts to event logs
Anomaly Identification
e ssh(UIHRF)
IHMD Database
abase
y Identification
ion
rtion of a packet
Anomaly Identification
services for specific application/procedure calls
D Database
a/o 1
Other unclassified Identifications
Anomaly Identification
p/o 1
Identifies vulnerability scans(UIHRF)
IHMD Database
p/o 1
Identifies port scans(UIHRF)
IHMD Database
a/o 1
record and analyze email traffic
Anomaly Identification
a/o 1
track software versions
Anomaly Identification
a/o 1
Anomaly Identification
p/o 1
Identifies for blaster worm(UIHRF)
IHMD Database
p/o 1
a/o 1
Display Maximum supported rate as advertised by AP
Anomaly Identification
a/o 1
Display Name of the network or group
Anomaly Identification
a/o 1
Display Last seen noise level
Anomaly Identification
a/o 1
Display Last seen signal level
Anomaly Identification
a/o 1
Display Network type (Probe, Adhoc, Infra, etc)
Anomaly Identification
a/o 1
Display WEP status
Anomaly Identification
p/o 1
Identifies unusual probes (UIHRF)
IHMD Database
p/o 1
Identifies MAC spoofing attacks (UIHRF)
IHMD Database
p/o 1
Identifies unusual probes (UIHRF)
IHMD Database
p/o 1
Identifies excessive disassociation(UIHRF)
IHMD Database
p/o 1
Identifies Netstumbler probe requests(UIHRF)
IHMD Database
p/o 1
Identifies Deauthenticate/Disassociate Flood(UIHRF)
IHMD Database
p/o 1
Identifies Lucent link test(UIHRF)
IHMD Database
p/o 1
Identifies MITM attacks(UIHRF)
IHMD Database
p/o 1
Identifies SSID brute force attempt(UIHRF)
IHMD Database
p/o 1
Identifies AP changing to a new channel(UIHRF)
IHMD Database
p/o 1
Identifies Broadcast disconnect/deauthenticate(UIHRF)
IHMD Database
p/o 1
Identifies Invalid BSS timestamps(UIHRF)
IHMD Database
p/o 1
Identifies weak-iv packet (UIHRF)
IHMD Database
Ontology Pics\[IHSMO with Concepts and Slots_v3.ont.xml"
Main Language
780 : 2088

Figure 5.16. IHSMO using Hozo editor (6/6)

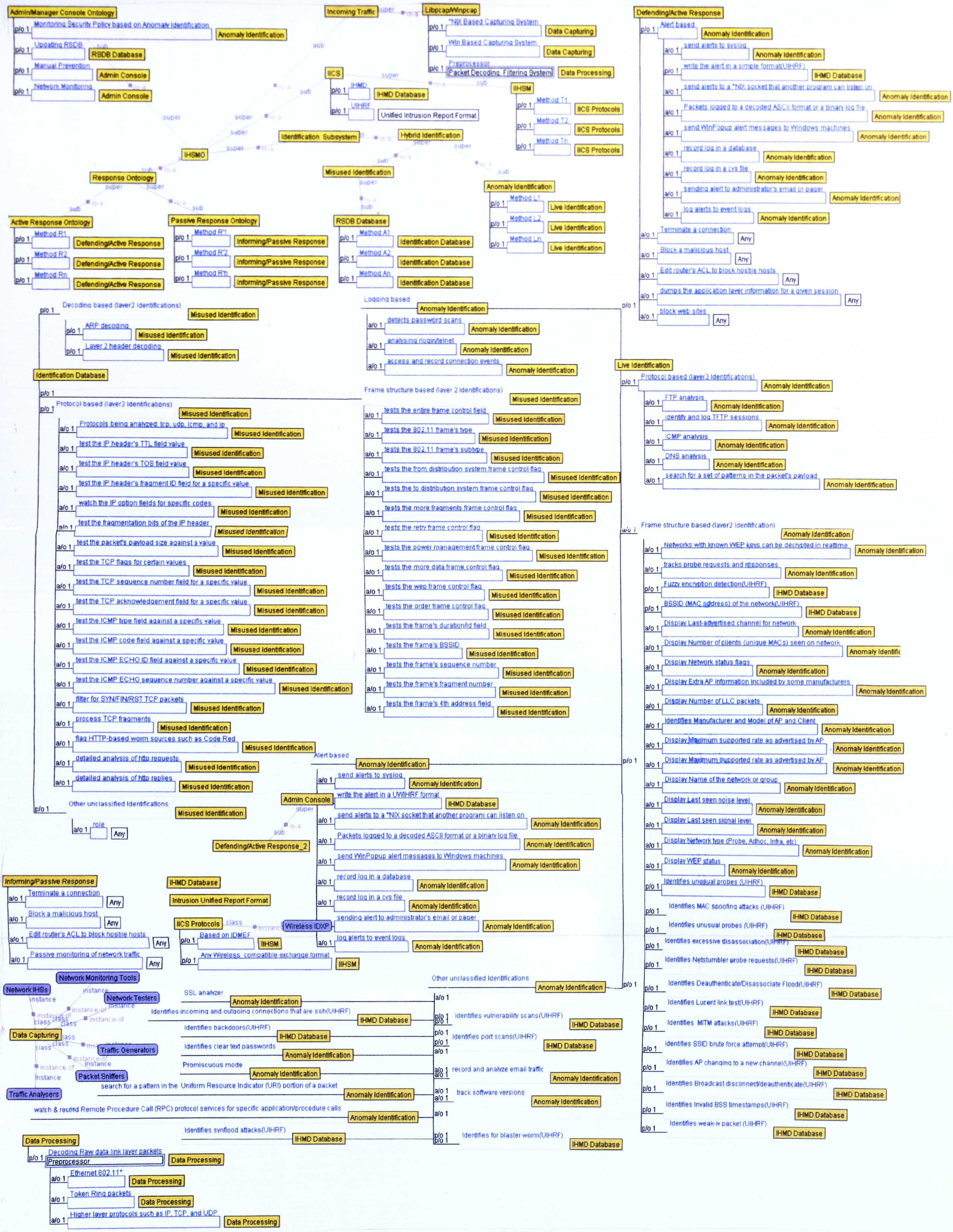

Admin/Manager Console Ontology
Monitoring Security Policy based on Anomaly Identification
Anomaly Identification
Updating RSDB
RSDB Database
Manual Prevention
Admin Console
Network Monitoring
Admin Console
IHSMO
Response Ontology
Active Response Ontology
Method R1
Method R2
Method Rn
Defending/Active Response
Passive Response Ontology
Method R'1
Method R'2
Method R'n
Informing/Passive Response
Incoming Traffic
Libpcap/Winpcap
*NIX Based Capturing System
Win Based Capturing System
Preprocessor
Packet Decoding, Filtering System
Data Capturing
Data Processing
IICS
IHMD
UIHRF
IHMD Database
Unified Intrusion Report Format
IIHSM
Method T1
Method T2
Method Tn
IICS Protocols
Identification Subsystem
Hybrid Identification
Misused Identification
RSDB Database
Method A1
Method A2
Method An
Identification Database
Anomaly Identification
Method L1
Method L2
Method Ln
Live Identification
super
sub
is-a
p/o 1
a/o 1
Defending/Active Response
Alert based
send alerts to syslog
write the alert in a simple format(UIHRF)
send alerts to a *NIX socket that another program can listen on
Packets logged to a decoded ASCII format or a binary log file
send WinPopup alert messages to Windows machines
record log in a database
record log in a cvs file
sending alert to administrator's email or pager
log alerts to event logs
Terminate a connection
Block a malicious host
Edit router's ACL to block hostile hosts
dumps the application layer information for a given session
block web sites
Any
Decoding based (layer2 Identifications)
ARP decoding
Layer 2 header decoding
Logging based
detects password scans
analysing rlogin/telnet
access and record connection events
Protocol based (layer3 Identifications)
Protocols being analyzed: tcp, udp, icmp, and ip
test the IP header's TTL field value
test the IP header's TOS field value
test the IP header's fragment ID field for a specific value
watch the IP option fields for specific codes
test the fragmentation bits of the IP header
test the packet's payload size against a value
test the TCP flags for certain values
test the TCP sequence number field for a specific value
test the TCP acknowledgement field for a specific value
test the ICMP type field against a specific value
test the ICMP code field against a specific value
test the ICMP ECHO ID field against a specific value
test the ICMP ECHO sequence number against a specific value
filter for SYN/FIN/RST TCP packets
process TCP fragments
flag HTTP-based worm sources such as Code Red
detailed analysis of http requests
detailed analysis of http replies
Other unclassified Identifications
role
Frame structure based (layer 2 identifications)
tests the entire frame control field
tests the 802.11 frame's type
tests the 802.11 frame's subtype
tests the from distribution system frame control flag
tests the to distribution system frame control flag
tests the more fragments frame control flag
tests the retry frame control flag
tests the power management frame control flag
tests the more data frame control flag
tests the wep frame control flag
tests the order frame control flag
tests the frame's duration/id field
tests the frame's BSSID
tests the frame's sequence number
tests the frame's fragment number
tests the frame's 4th address field
Live Identification
Protocol based (layer3 Identifications)
FTP analysis
identify and log TFTP sessions
ICMP analysis
DNS analysis
search for a set of patterns in the packet's payload
Frame structure based (layer2 Identification)
Networks with known WEP keys can be decrypted in realtime
tracks probe requests and responses
Fuzzy encryption detection(UIHRF)
BSSID (MAC address) of the network(UIHRF)
Display Last-advertised channel for network
Display Number of clients (unique MACs) seen on network
Display Network status flags
Display Extra AP information included by some manufacturers
Display Number of LLC packets
Identifies Manufacturer and Model of AP and Client
Display Maximum supported rate as advertised by AP
Display Maximum supported rate as advertised by AP
Display Name of the network or group
Display Last seen noise level
Display Last seen signal level
Display Network type (Probe, Adhoc, Infra, etc)
Display WEP status
Identifies unusual probes (UIHRF)
Identifies MAC spoofing attacks (UIHRF)
Identifies unusual probes (UIHRF)
Identifies excessive disassociation(UIHRF)
Identifies Netstumbler probe requests(UIHRF)
Identifies Deauthenticate/Disassociate Flood(UIHRF)
Identifies Lucent link test(UIHRF)
Identifies MITM attacks(UIHRF)
Identifies SSID brute force attempt(UIHRF)
Identifies AP changing to a new channel(UIHRF)
Identifies Broadcast disconnect/deauthenticate(UIHRF)
Identifies Invalid BSS timestamps(UIHRF)
Identifies weak-iv packet (UIHRF)
Admin Console
Alert based
send alerts to syslog
write the alert in a UWIHRF format
send alerts to a *NIX socket that another program can listen on
Packets logged to a decoded ASCII format or a binary log file
send WinPopup alert messages to Windows machines
record log in a database
record log in a cvs file
sending alert to administrator's email or pager
log alerts to event logs
Defending/Active Response_2
Informing/Passive Response
Terminate a connection
Block a malicious host
Edit router's ACL to block hostile hosts
Passive monitoring of network traffic
IHMD Database
Intrusion Unified Report Format
IICS Protocols
class
instance
Wireless IDXP
Based on IDMEF
Any Wireless compatible exchange format
IIHSM
Network Monitoring Tools
Network IHSs
Network Testers
Data Capturing
Traffic Generators
Packet Sniffers
Traffic Analysers
instance-of
SSL analyzer
Identifies incoming and outgoing connections that are ssh(UIHRF)
Identifies backdoors(UIHRF)
Identifies clear text passwords
Promiscuous mode
search for a pattern in the Uniform Resource Indicator (URI) portion of a packet
watch & record Remote Procedure Call (RPC) protocol services for specific application/procedure calls
Identifies synflood attacks(UIHRF)
Other unclassified Identifications
Identifies vulnerability scans(UIHRF)
Identifies port scans(UIHRF)
record and analyze email traffic
track software versions
Identifies for blaster worm(UIHRF)
Data Processing
Decoding Raw data link layer packets
Preprocessor
Ethernet 802.11*
Token Ring packets
Higher layer protocols such as IP, TCP, and UDP

Fig. 5.16: IHSMO Ontology using HOZO editor

5.2.4.1 Methods for Misuse and Anomaly Identifications

The misuse identification uses the following methods to identify the intrusions. The following methods are stored in identification database.

Misuse Identification Database, Methods A1 to A*n*

Decoding based (layer 2 Identifications):

ARP decoding

Layer 2 header decoding

Protocol based (layer 3 Identification):

Protocols: tcp, udp, icmp, and ip

Test the IP header's TTL field value

Test the IP header's TOS field value

Test the IP header's fragment ID field for a specific value

Watch the IP option fields for specific codes

Test the fragmentation bits of the IP header

Test the packet's payload size against a value

Test the TCP flags for certain values

Test the TCP sequence number field for a specific value

Test the TCP acknowledgement field for a specific value

Test the ICMP type field against a specific value

Test the ICMP code field against a specific value

Test the ICMP ECHO ID field against a specific value

Test the ICMP ECHO sequence number against a specific value

Filter for SYN/FIN/RST TCP packets

Process TCP fragments

Flag HTTP-based worm sources such as Code Red

Detailed analysis of http requests

Detailed analysis of http replies

Frame structure based (layer 2 Identification):

Tests the entire frame control field

Tests the 802.11 frame's type

Tests the 802.11 frame's subtype

Tests the from distribution system frame control flag

Tests the to distribution system frame control flag

Tests the more fragments frame control flag

Tests the retry frame control flag

Tests the power management frame control flag

Tests the more data frame control flag

Tests the wep frame control flag

Tests the order frame control flag

Tests the frame's duration/id field

Tests the frame's BSSID

Tests the frame's sequence number

Tests the frame's fragment number

Tests the frame's 4th address field

Other unclassified Identifications

The anomaly identification uses the following methods to identify the intrusions. The following are live identification methods.

<u>Live Anomaly Identification, Methods L1 to L*n*</u>

Alert based:

Send alerts to syslog

Write the alert in a simple format

Send alerts to a *NIX socket that another program can listen on

Packets logged to a decoded ASCII format or a binary log file

Send WinPopup alert messages to Windows machines

Record log in a database

Record log in a cvs file

Sending alert to administrator's email or pager

Log alerts to event logs

Protocol based (layer 3 Identifications):

FTP analysis

Identify and log TFTP sessions

ICMP analysis

DNS analysis

Search for a set of patterns in the packet's payload

Logging based

Detects password scans

Analysing rlogin/telnet

Access and record connection events

Frame structure based (layer 2 Identification):

Networks with known WEP keys can be decrypted in real-time

Tracks probe requests and responses

Fuzzy encryption detection

BSSID (MAC address) of the network

Display Last-advertised channel for network

Display Number of clients (unique MACs) seen on network

Display Network status flags

Display Extra AP information included by some manufacturers

Display Number of LLC packets

Identifies Manufacturer and Model of AP and Client

Display Maximum supported rate as advertised by AP

Display Name of the network or group

Display Last seen noise level

Display Last seen signal level

Display Network type (Probe, ad hoc, Infra, etc)

Display WEP status

Identifies unusual probes

Identifies excessive disassociation

Identifies Netstumbler probe requests

Identifies Deauthenticate/Disassociate Flood

Identifies Lucent link test

Identifies SSID brute force attempt

Identifies AP changing to a new channel

Identifies Broadcast disconnect/deauthenticate

Identifies Invalid BSS timestamps

Identifies weak-iv packet

Identifies MITM attacks

Identifies MAC spoofing attacks

Other unclassified Identifications:

Identifies vulnerability scans

Identifies port scans

Watch & record RPC services for specific application/procedure calls

Record and analyse email traffic

Track software versions

Identifies for blaster worm

Identifies synflood attacks

SSL analyser

Identifies incoming and outgoing connections that are ssh

Identifies backdoors

Identifies clear text passwords

Promiscuous mode

Search for a pattern in the URI portion of a packet

5.2.4.2 Methods for Passive and Active Responses

Informing/Passive Response, Methods R`1 to R`*n*

Terminate a connection

Block a malicious host

Edit router's ACL to block hostile hosts

Passive monitoring of network traffic

Defending/Active Response, Methods R1 to R*n*

Alert based:

Send alerts to syslog

Write the alert in a simple format

Send alerts to a *NIX socket that another program can listen on

Packets logged to a decoded ASCII format or a binary log file

Send WinPopup alert messages to Windows machines

Record log in a database

Record log in a cvs file

Sending alert to administrator's email or pager

Log alerts to event logs

Terminate a connection

Block a malicious host

Edit router's ACL to block hostile hosts

Dumps the application layer information for a given session

Block web sites .

5.2.4.3 Methods T1 to Tn in IIHSM and IHMD Database Contents

Below "p/o" stands for *part-of.*

p/o 1 Fuzzy encryption detection(UIHRF)

p/o 1 BSSID (MAC address) of the network(UIHRF)

p/o 1 Identifies unusual probes (UIHRF)

p/o 1 Identifies excessive disassociation(UIHRF)

p/o 1 Identifies Netstumbler probe requests(UIHRF)

p/o 1 Identifies Deauthenticate/Disassociate Flood(UIHRF)

p/o 1 Identifies Lucent link test(UIHRF)

p/o 1 Identifies SSID brute force attempt(UIHRF)

p/o 1 Identifies AP changing to a new channel(UIHRF)

p/o 1 Identifies Broadcast disconnect/deauthenticate(UIHRF)

p/o 1 Identifies Invalid BSS timestamps(UIHRF)

p/o 1 Identifies weak-iv packet (UIHRF)

p/o 1 Identifies MITM attacks(UIHRF)

p/o 1 Identifies MAC spoofing attacks (UIHRF)

p/o 1 Identifies vulnerability scans(UIHRF)

p/o 1 Identifies port scans(UIHRF)

p/o 1 Identifies for blaster worm(UIHRF)

p/o 1 Identifies synflood attacks(UIHRF) p/o 1 Identifies incoming and outgoing connections that are ssh(UIHRF)

p/o 1 Identifies backdoors(UIHRF)

p/o 1 write the alert in a UIHRF format

5.2.4.4 XML Source Codes of IHSMO with Slots

The source code of IHSMO extracted from the Hozo ontology editor environment showing values of concepts, slots and defined axioms for IHSMO can be found in E.3 of Appendix E.

5.2.4.5 Ontology without Slots

Figure 5.17 shows the development of IHSMO with concepts only and elimination of slots. The output of Hozo for this ontology can also be found in Appendix E.

5.2.4.6 XML Source Codes of IHSMO without Slots

The source code of IHSMO extracted from the Hozo ontology editor environment showing values of concepts and defined axioms for IHSMO can be found in E.4 of Appendix E.

5.3 Conclusion

An IHSMO has been designed using Hozo ontology editor in order to act as a "prototype" to represent a unified IHS reference model and to standardise the building blocks of existing and new IHS systems. The IHS reference model for WLANs is vital for evaluation of current and development of future IHS systems and it further contributes in enhancing security level of layer 2 and 3 of the OSI model in WLANs, which have always been vulnerable to intruders. The design has listed steps in the ontology-development process and defining class hierarchies and properties of classes, attributes and instances. An advanced element of ontology representation has been introduced by definitions and constraints of axioms, rules and functions on the terminologies of IHSMO. Chapter 6 is focusing on evaluation of existing IHSs using IHSMO.

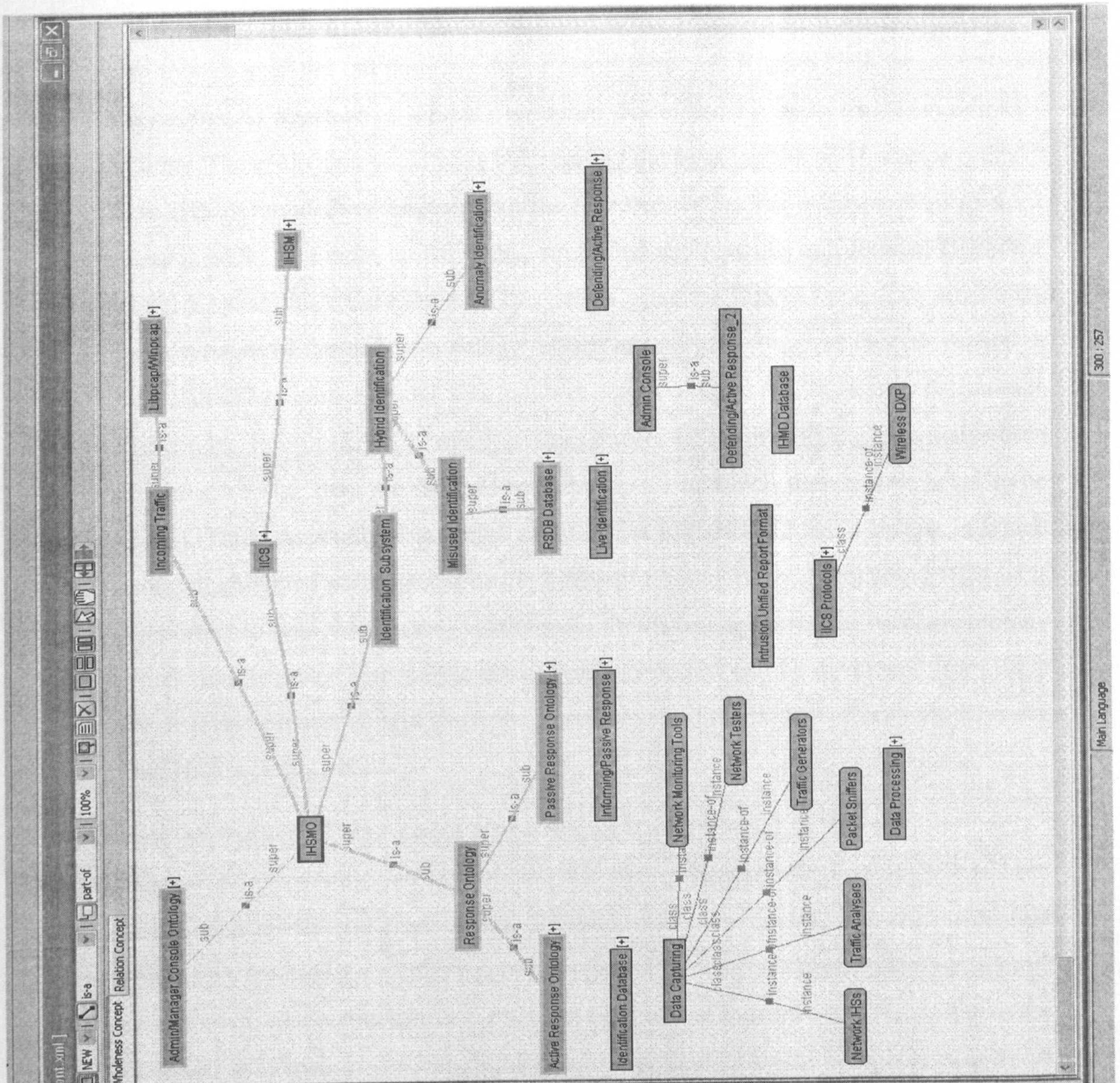


Figure 5.17. IHSMO without slots using Hozo editor

6. Using IHSMO Reference Model to Evaluate Existing IHSs

In this chapter the ontology model of Snort, Bro and Kismet IHSs are developed according to structure of IHSMO reference model and by using the Hozo ontology editor. The evaluation could have been carried out by using the XML source codes of the IHSMO modelling language but the Hozo editor has the ability to convert these source codes to a more user friendly, simplified and readable text format. Therefore, in order to evaluate the existing IHS systems using the IHSMO reference model, the text versions of these three ontology models are compared against the text version of the IHSMO reference model. The full text output of these ontologies can be found in Appendix E in which the missing components of these IHS systems have been highlighted (i.e. they are inserted into the places in which they are expected to be seen). Thus, Appendix E contains all identical full IHSMO text versions in which some highlighted parts are indicating different missing parts of evaluated IHSs. It is important to note that the text versions of outputs can be converted back to ontology modelling languages hence the actual ontology model can be developed from these texts. The next section will show the development of the Snort ontology model using the Hozo ontology editor.

6.1 An Ontology Model of Snort Wireless IHS system

Figure 6.1 shows an ontology model for Snort wireless. This ontology has been developed using structure of the IHSMO reference model. For the evaluation purposes, the model structure has been exported from Hozo ontology language into a text format. The output of this format can be found in appendix E.4. According to the IHSMO reference model, the highlighted lines, depicts major components of Snort wireless system lacking from the full IHSMO. These missing components have significant contribution towards the IHS operation in terms of accuracy, integrity of the identification system, reliability of response system and effectiveness and transparent operation of Snort wireless system.

The full text version of IHSMO has been shown in Appendix E.5. The IHSMO has been used as a reference model to prove the weaknesses of Snort wireless IHS system and its effectiveness in securing a wireless network.

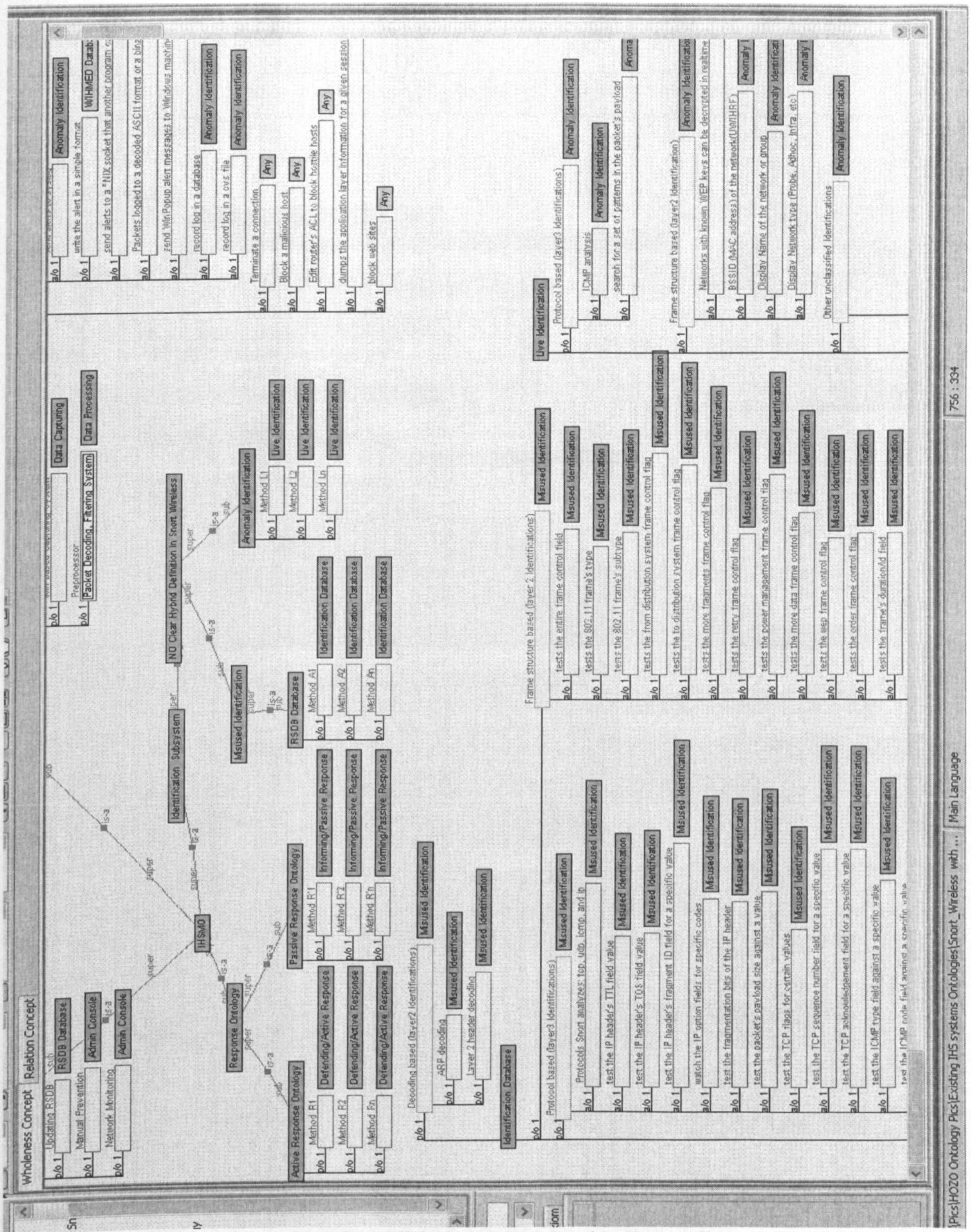


Figure 6.1. Snort Wireless IHSMO using Hozo editor

Following is the text version of IHSMO reference model ontology, which presents the lacking parts of major components of Snort wireless IHS. The text format below is derived from Hozo output according to the IHSMO hierarchical structure.

```
(Ontology Name)
IHSMO with Concepts and Slots_v2.ont.xml

  - IICS
    p/o 1 IHMD : IHMD Database
    p/o 1 UIHRF : Intrusion Unified Report Format
   - IIHSM
     p/o 1 Method T1 : IICS Protocols
     p/o 1 Method T2 : IICS Protocols
     p/o 1 Method Tn : IICS Protocols
- Identification Subsystem
 -Hybrid Identification
  -Anomaly Identification (AI)
   -Live Identification
    a/o 1 FTP analysis: AI
    a/o 1 identifies and log TFTP sessions: AI
    a/o 1 DNS analysis: AI
    p/o 1 Logging based : AI
    a/o 1 detects password scans: AI
    a/o 1 analysing rlogin/telnet: AI
    a/o 1 access and record connection events: AI
    a/o 1 tracks probe requests and responses: AI
    a/o 1 Display Last-advertised channel for network: AI
    a/o 1 Display Number of clients (unique MACs) seen on network:
        AI
    a/o 1 Display Network status flags: AI
    a/o 1 Display Extra AP info included by some manufacturers: AI
    a/o 1 Display Number of LLC packets: AI
    a/o 1 Identifies Manufacturer and Model of AP and Client: AI
    a/o 1 Display Maximum supported rate as advertised by AP: AI
    a/o 1 Display Maximum supported rate as advertised by AP: AI
    a/o 1 Display Last seen noise level: AI
    a/o 1 Display Last seen signal level: AI
    a/o 1 Display WEP status: AI
    p/o 1 Other unclassified Identifications : AI
    p/o 1 Identifies vulnerability scans(UIHRF) : AI
    p/o 1 Identifies port scans(UIHRF) : AI
    a/o 1 record and analyse email traffic: AI
    a/o 1 track software versions: AI
    a/o 1 identifies clear text passwords: AI
    p/o 1 Fuzzy encryption detection(UIHRF) : IHMD Database (WD)
    p/o 1 Identifies unusual probes (UIHRF) : WD
    p/o 1 Identifies unusual probes (UIHRF) : WD
    p/o 1 Identifies excessive disassociation(UIHRF) : WD
    p/o 1 Identifies Netstumbler probe requests(UIHRF) : WD
    p/o 1 Identifies Deauthenticate/Disassociate Flood(UIHRF) : WD
    p/o 1 Identifies Lucent link test(UIHRF) : WD
    p/o 1 Identifies SSID brute force attempt(UIHRF) : WD
    p/o 1 Identifies AP changing to a new channel(UIHRF) : WD
    p/o 1 Identifies Broadcast discon/deauthenticate(UIHRF) : WD
    p/o 1 Identifies Invalid BSS timestamps(UIHRF) : WD
    p/o 1 Identifies weak-iv packet (UIHRF) : WD
    p/o 1 Identifies MITM attacks(UIHRF) : WD
    p/o 1 Identifies MAC spoofing attacks (UIHRF) : WD
    p/o 1 Identifies for blaster worm(UIHRF) : WD
    p/o 1 Identifies synflood attacks(UIHRF) : WD
    p/o 1 Identifies in and out connections that are ssh(UIHRF):WD
    p/o 1 Identifies backdoors(UIHRF) : WD
    a/o 1 SSL analyser: AI
- Informing/Passive Response
    a/o 1 Edit router's ACL to block hostile hosts: Any
- Defending/Active Response
    a/o 1 sending alert to administrator's email or pager: AI
    a/o 1 log alerts to event logs : AI
- Admin Console
  - Defending/Active Response_2
- IHMD Database (WD)
- Intrusion Unified Report Format
- IICS Protocols
    p/o 1 Based on IDMEF : IIHSM
    p/o 1 Any Wireless  compatible exchange format : IIHSM
  - Wireless IDXP
```

The above output clearly indicates that Snort wireless does not support the IICS protocol and its dependent attributes. It also shows that there are many important parts are missing from anomaly identification subsystem that is part of live identification attributes. The Snort wireless does not have understanding of UIHRF or its equivalent. Hence there is no IHMD or its equivalent exists in Snort wireless IHS.

6.2 An Ontology Model of Bro Wireless IHS system

Figure 6.2 shows an ontology model for Bro IHS. This ontology has been developed using the IHSMO reference model's structure. The IHSMO has been used as a reference model to prove the weaknesses of Bro IHS system and its effectiveness and operation in securing a wireless network.

The ontology model has been converted into a text format and can be found in Appendix E.6. The highlighted lines clearly show lack of critical components in Bro IHS compare to IHSMO reference model. These components have a significant impact on the IHS operation in terms of accuracy and integrity of identification, its reliability in tackling the intruders, effectiveness in securing the network and its smooth, transparent and low power operation. The full text version of IHSMO has been shown in Appendix E.7.

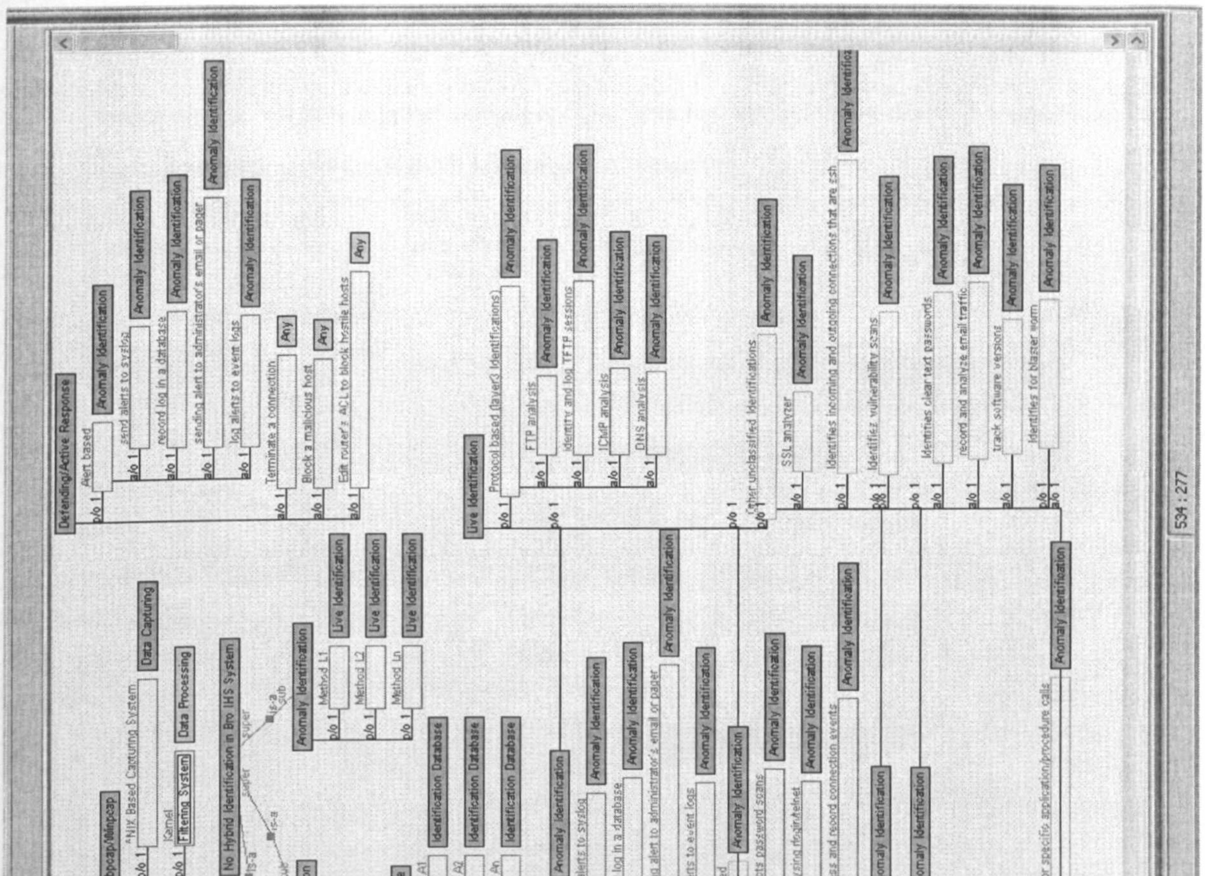

Wholeness Concept
Relation Concept
Admin/Manager Console Ontology
Monitoring Security Policy based on Anomaly Identification
Anomaly Identification
Incoming Traffic
Updating RSDB
RSDB Database
Manual Prevention
Admin Console
Network Monitoring
Admin Console
Libpcap/Winpcap
*NIX Based Capturing System
Data Capturing
Kernel
Filtering System
Data Processing
IHSMO
Identification Subsystem
No Hybrid Identification in Bro IHS System
Misused Identification
Anomaly Identification
Method L1
Method L2
Method Ln
Live Identification
RSDB Database
Method A1
Method A2
Method An
Identification Database
Response Ontology
Active Response Ontology
Method R1
Method R2
Method Rn
Defending/Active Response
Passive Response Ontology
Method R'1
Method R'2
Method R'n
Informing/Passive Response
Identification Database
Protocol based (layer3 Identifications)
filter for SYN/FIN/RST TCP packets
process TCP fragments
flag HTTP-based worm sources such as Code Red
detailed analysis of http requests
detailed analysis of http replies
Misused Identification
Alert based
send alerts to syslog
record log in a database
sending alert to administrator's email or pager
log alerts to event logs
Logging based
detects password scans
analysing rlogin/telnet
access and record connection events
Identifies backdoors
Identifies port scans
Informing/Passive Response
Terminate a connection
Block a malicious host
Edit router's ACL to block hostile hosts
Passive monitoring of network traffic
Any
Admin Console
watch & record Remote Procedure Call (RPC) protocol services for specific application/procedure calls
Defending/Active Response_2
Defending/Active Response
Alert based
send alerts to syslog
record log in a database
sending alert to administrator's email or pager
log alerts to event logs
Terminate a connection
Block a malicious host
Edit router's ACL to block hostile hosts
Live Identification
Protocol based (layer3 Identifications)
FTP analysis
identify and log TFTP sessions
ICMP analysis
DNS analysis
Other unclassified Identifications
SSL analyzer
Identifies incoming and outgoing connections that are ssh
Identifies vulnerability scans
Identifies clear text passwords
record and analyze email traffic
track software versions
Identifies for blaster worm
is-a
super
sub
p/o 1
a/o 1
534 : 277
Main Language
cs\HOZO Ontology Pics\Existing IHS systems Ontologies\Bro_IHS with Concep...

Figure 6.2. Bro Wireless IHSMO using Hozo editor

Following is the text version of IHSMO reference model ontology, which shows missing major components of Bro IHS, hence making it to be an unreliable IHS in protecting a WLAN against intruders. The text format below is derived from Hozo output according to the IHSMO hierarchical structure.

```
(Ontology Name)
IHSMO with Concepts and Slots_v2.ont.xml
 (Wholeness Concept)
-Incoming Traffic
  - Libpcap/Winpcap
    p/o 1 Pre-processor [Packet Decoding, Filtering System] : DP
 - IICS
    p/o 1 IHMD : IHMD Database
    p/o 1 UIHRF : Intrusion Unified Report Format
  - IIHSM
     p/o 1 Method T1 : IICS Protocols
     p/o 1 Method T2 : IICS Protocols
     p/o 1 Method Tn : IICS Protocols
 - Identification Subsystem
  -Hybrid Identification
   -Misuse Identification (MI)
    -Identification Database
    p/o 1 ARP decoding : MI
    p/o 1 Decoding based (layer2 Identifications) : MI
    p/o 1 Layer 2 header decoding : MI
    a/o 1 Protocols being analysed: tcp, udp, icmp, and ip: MI
    a/o 1 test the IP header's TOS field value: MI
    a/o 1 test the IP header's TTL field value: MI
    a/o 1 test the IP header's TTL field value: MI
    a/o 1 test the IP header's TTL field value: MI
    a/o 1 test the IP header's fragment ID field for a specific
          value: MI
    a/o 1 watch the IP option fields for specific codes: MI
    a/o 1 test the fragmentation bits of the IP header: MI
    a/o 1 test the packet's payload size against a value: MI
    a/o 1 test the TCP flags for certain values: MI
    a/o 1 test the TCP sequence number for a specific value: MI
    a/o 1 test the TCP acknowledgement for a specific value: MI
    a/o 1 test the ICMP code field against a specific value: MI
    a/o 1 test the ICMP type field against a specific value: MI
    a/o 1 test the ICMP ECHO ID field against a specific value: MI
    a/o 1 test the ICMP type field against a specific value: MI
    a/o 1 test the ICMP ECHO sequence number against a specific
          value: MI
    p/o 1 Frame structure based (layer 2 Identifications) : MI
    a/o 1 tests the entire frame control field: MI
    a/o 1 tests the 802.11 frame's type: MI
    a/o 1 tests the 802.11 frame's subtype: MI
    a/o 1 tests the from distribution system frame control flag: MI
    a/o 1 tests the to distribution system frame control flag: MI
    a/o 1 tests the more fragments frame control flag: MI
    a/o 1 tests the retry frame control flag: MI
    a/o 1 tests the power management frame control flag: MI
    a/o 1 tests the more data frame control flag: MI
    a/o 1 tests the wep frame control flag: MI
    a/o 1 tests the order frame control flag: MI
    a/o 1 tests the frame's duration/id field: MI
    a/o 1 tests the frame's BSSID: MI
    a/o 1 tests the frame's sequence number: MI
    a/o 1 tests the frame's fragment number: MI
    a/o 1 tests the frame's 4th address field: MI
    p/o 1 Other unclassified Identifications : MI
    a/o 1 role: Any
 - Anomaly Identification
   -Live Identification
    a/o 1 search for a set of patterns in the packet's payload: AI
    a/o 1 Frame structure based (layer2 Identification): AI
    a/o 1 Networks with known WEP keys can be decrypted in
          Real-time: AI
    a/o 1 tracks probe requests and responses: AI
    a/o 1 Display Last-advertised channel for network: AI
    a/o 1 Display Number of clients (unique MACs) seen on network:
          AI
    a/o 1 Display Network status flags: AI
    a/o 1 Display Extra AP information included by some
          Manufacturers: AI
    a/o 1 Display Number of LLC packets: AI
    a/o 1 Identifies Manufacturer and Model pf AP and Client: AI
    a/o 1 Display Maximum supported rate as advertised by AP: AI
    a/o 1 Display Maximum supported rate as advertised by AP: AI
    a/o 1 Display Name of the network or group: AI
    a/o 1 Display Last seen noise level: AI
    a/o 1 Display Last seen signal level: AI
    a/o 1 Display Network type (Probe, Adhoc, Infra, etc): AI
    a/o 1 Display WEP status: AI
    a/o 1 search for a pattern in the URI portion of a packet: AI
    a/o 1 send alerts to a *NIX socket that another program can
          listen on: AI
    a/o 1 Packets logged to a decoded ASCII format or a binary log
          file: AI
    a/o 1 send WinPopup alert messages to Windows machines: AI
    a/o 1 record log in a cvs file: AI
    a/o 1 send alerts to a *NIX socket that another program can
          listen on: AI
    a/o 1 Packets logged to a decoded ASCII format or a binary log
          File: AI
    a/o 1 send WinPopup alert messages to Windows machines: AI
    a/o 1 record log in a cvs file: AI
    a/o 1 Watch Remote Procedure Call (RPC) protocol services for
          Specific application/procedure calls: AI
    a/o 1 Search for a pattern in the Uniform Resource Indicator
         (URI) portion of packet: AI
    p/o 1 Fuzzy encryption detection(UIHRF) : WD
    p/o 1 BSSID (MAC address) of the network(UIHRF):WD
    p/o 1 Identifies unusual probes (UIHRF) : WD
    p/o 1 Identifies unusual probes (UIHRF) : WD
    p/o 1 Identifies excessive disassociation(UIHRF):WD
    p/o 1 Identifies Netstumbler probe requests(UIHRF):WD
    p/o 1 Identifies Deauthenticate/Disassociate Flood(UIHRF):WD
    p/o 1 Identifies SSID brute force attempt(UIHRF):WD
    p/o 1 Identifies Lucent link test(UIHRF):WD
    p/o 1 Identifies AP changing to a new channel(UIHRF):WD
    p/o 1 Identifies Broadcast disconnect/deauthenticate(UIHRF):WD
    p/o 1 Identifies Invalid BSS timestamps(UIHRF):WD
    p/o 1 Identifies weak-iv packet (UIHRF):WD
    p/o 1 Identifies  MITM attacks(UIHRF):WD
    p/o 1 Identifies MAC spoofing attacks (UIHRF):WD
    p/o 1 Identifies synflood attacks(UIHRF):WD
    a/o 1 Promiscuous mode: Anomaly Identification
    p/o 1 write the alert in a simple format(UIHRF):WD
    a/o 1 dumps the application layer information for a given
          Session: Any
    a/o 1 block web sites: Any
- IHMD Database
- Intrusion Unified Report Format
- IICS Protocols
   p/o 1 Based on IDMEF : IIHSM
   p/o 1 Any Wireless  compatible exchange format:IIHSM
  - Wireless IDXP
- Data Processing (DP)
   p/o 1 Decoding Raw data link layer packets [Pre-processor]:DP
   a/o 1 Ethernet 802.11*: DP
   a/o 1 Token Ring packets: DP
   a/o 1 Higher layer protocols such as IP, TCP, and UDP:DP
```

It can be noted that there are significant issues with identification system, including anomaly identification subsystem. There are not robust misuse identification subsystems in place and major attributes are missing from live identification subsystem. Similar to the Snort wireless IHS, the Bro IHS does not have a clear understanding of IICS, UIHRF and IIHSM systems that makes it to be an unreliable IHS in protecting a wireless network against intruders.

6.3 An Ontology Model of Kismet IHS system

Figure 6.3 shows an ontology model for Kismet IHS. This ontology has been developed based on IHSMO reference model's structure. The ontology model has been converted to a text format and can be found in appendix E.8. Like previous IHS systems and according to IHSMO reference model, the highlighted lines show lack of vital components in Kismet IHS.

These missing components would have significant effect on the IHS operation in terms of accuracy and integrity of the identification system, etc. The full text version of IHSMO can be found in Appendix E.9.

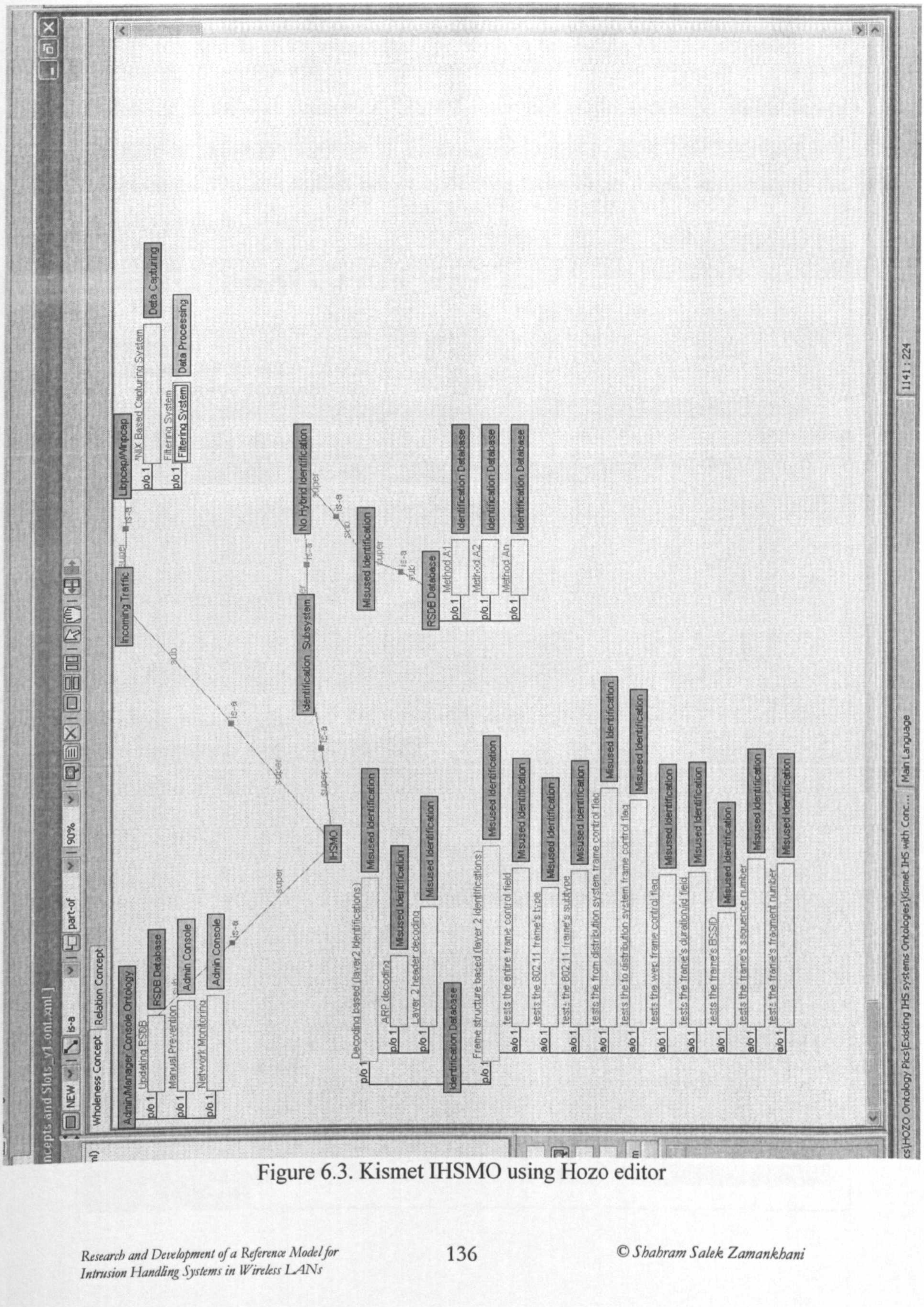


Figure 6.3. Kismet IHSMO using Hozo editor

The IHSMO has been used as a reference model to prove the weaknesses of Kismet IHS system and its effectiveness and operation in securing a wireless network.

Following is the text version of IHSMO reference model ontology, which shows the missing major components of Kismet IHS compared to IHSMO structure and components. The text format below is derived from Hozo output according to the IHSMO hierarchical structure.

```
(Ontology Name)
IHSMO with Concepts and Slots_v2.ont.xml

(Wholeness Concept)
- IHSMO
  - Incoming Traffic
    - Libpcap/Winpcap
    p/o 1 Pre-processor [Packet Decoding, Filtering System] : DP
  - IICS
    p/o 1 IHMD : IHMD Database
    p/o 1 UIHRF : Intrusion Unified Report Format
   - IIHSM
     p/o 1 Method T1 : IICS Protocols
     p/o 1 Method T2 : IICS Protocols
     p/o 1 Method Tn : IICS Protocols
  - Identification Subsystem
   - Hybrid Identification
    - Anomaly Identification
      p/o 1 Method L1 : Live Identification
      p/o 1 Method L2 : Live Identification
      p/o 1 Method Ln : Live Identification

  - Response Ontology
    - Active Response Ontology
      p/o 1 Method R1 : Defending/Active Response
      p/o 1 Method R2 : Defending/Active Response
      p/o 1 Method Rn : Defending/Active Response
    - Passive Response Ontology
      p/o 1 Method R'1 : Informing/Passive Response
      p/o 1 Method R'2 : Informing/Passive Response
      p/o 1 Method R'n : Informing/Passive Response
  -Misuse Identification (MI)
   - Identification Database
     p/o 1 Protocol based (layer3 Identifications) : MI
     a/o 1 Protocols being analysed: tcp, udp, icmp, and ip: MI
     a/o 1 test the IP header's TTL field value: MI
     a/o 1 test the IP header's TOS field value: MI
     a/o 1 test the IP header's fragment ID field for a specific
           value: MI
     a/o 1 watch the IP option fields for specific codes: MI
     a/o 1 test the fragmentation bits of the IP header: MI
     a/o 1 test the packet's payload size against a value: MI
     a/o 1 test the TCP flags for certain values: MI
     a/o 1 test the TCP sequence number field for a specific value
           : MI
     a/o 1 test the TCP acknowledgement field for a specific value
           : MI
    a/o 1 test the ICMP type field against a specific value: MI
    a/o 1 test the ICMP code field against a specific value: MI
    a/o 1 test the ICMP ECHO ID field against a specific value: MI
    a/o 1 test the ICMP ECHO sequence number against a specific
          value: MI
    a/o 1 filter for SYN/FIN/RST TCP packets: MI
    a/o 1 process TCP fragments : MI
    a/o 1 flag HTTP-based worm sources such as Code Red: MI
    a/o 1 detailed analysis of http requests: MI
    a/o 1 detailed analysis of http replies: MI
    a/o 1 tests the more fragments frame control flag: MI
    a/o 1 tests the retry frame control flag: Misuse
          Identification
    a/o 1 tests the power management frame control flag: MI
    a/o 1 tests the more data frame control flag: MI
    a/o 1 tests the wep frame control flag: MI
    a/o 1 tests the order frame control flag: MI
    a/o 1 tests the frame's 4th address field: MI
    p/o 1 Other unclassified Identifications : MI
    a/o 1 role: Any
   - Live Identification
    p/o 1 Protocol based (layer3 Identifications) : AI
    a/o 1 FTP analysis: AI
    a/o 1 identifies and log TFTP sessions: AI
    a/o 1 ICMP analysis: AI
    a/o 1 DNS analysis: AI
    a/o 1 search for a set of patterns in the packet's payload: AI
    p/o 1 Logging based : AI
    a/o 1 detects password scans: AI
    a/o 1 analysing rlogin/telnet: AI
    a/o 1 access and record connection events: AI
    a/o 1 Frame structure based (layer2 Identification): AI
    a/o 1  Networks with known WEP keys can be decrypted in real-
time
          : AI
    a/o 1 tracks probe requests and responses: AI
    p/o 1 Fuzzy encryption detection(UIHRF) : WD
    p/o 1 BSSID (MAC address) of the network(UIHRF) : WD
    a/o 1 Display Last-advertised channel for network: AI
    a/o 1 Display Number of clients (unique MACs) seen on network:
          AI
    a/o 1 Display Network status flags: AI
    a/o 1 Display Extra AP information included by some
          manufacturers: AI
    a/o 1 Display Number of LLC packets: Anomaly Identification
    a/o 1 Identifies Manufacturer and Model pf AP and Client: AI
    a/o 1 Display Maximum supported rate as advertised by AP: AI
    a/o 1 Display Maximum supported rate as advertised by AP: AI
    a/o 1 Display Name of the network or group: AI
    a/o 1 Display Last seen noise level: AI
    a/o 1 Display Last seen signal level: AI
    a/o 1 Display Network type (Probe, Adhoc, Infra, etc): AI
    a/o 1 Display WEP status: AI
    p/o 1 Identifies unusual probes (UIHRF) : WD
    p/o 1 Identifies unusual probes (UIHRF) : WD
    p/o 1 Identifies excessive disassociation(UIHRF) : WD
    p/o 1 Identifies Netstumbler probe requests(UIHRF) : WD
    p/o 1 Identifies Deauthenticate/Disassociate Flood(UIHRF) : WD
    p/o 1 Identifies Lucent link test(UIHRF) : WD
    p/o 1 Identifies SSID brute force attempt(UIHRF) : WD
    p/o 1 Identifies AP changing to a new channel(UIHRF) : WD
    p/o 1 Identifies Broadcast disconnect/deauthenticate(UIHRF)
          WD
    p/o 1 Identifies Invalid BSS timestamps(UIHRF) : WD
    p/o 1 Identifies weak-iv packet (UIHRF) : WD
    p/o 1 Identifies  MITM attacks(UIHRF) : WD
    p/o 1 Identifies MAC spoofing attacks (UIHRF) : WD
    p/o 1 Other unclassified Identifications : AI
    p/o 1 Identifies vulnerability scans(UIHRF) : WD
    p/o 1 Identifies port scans(UIHRF) : WD
    a/o 1 watch & record RPC services for specific
          application/procedure calls : AI
    a/o 1 record and analyse email traffic: AI
    a/o 1 track software versions: AI
    p/o 1 Identifies for blaster worm(UIHRF) : WD
    p/o 1 Identifies synflood attacks(UIHRF) : WD
    a/o 1 SSL analyser: AI
    p/o 1 Identifies incoming and outgoing connections that are
          ssh (UIHRF) : WD
    p/o 1 Identifies backdoors(UIHRF) : WD
    a/o 1 Identifies clear text passwords: AI
    a/o 1 Promiscuous mode: AI
    a/o 1 search for a pattern in the URI portion of a packet: AI
    p/o 1 Alert based : AI
    a/o 1 send alerts to syslog: AI
    p/o 1 write the alert in a UIHRF format : WD
    a/o 1 send alerts to a *NIX socket that another program can
          listen on : AI
    a/o 1 Packets logged to a decoded ASCII format or a binary log
          file: AI
    a/o 1 send WinPopup alert messages to Windows machines: AI
    a/o 1 record log in a database: AI
    a/o 1 record log in a cvs file: AI
    a/o 1 sending alert to administrator's email or pager: AI
    a/o 1 log alerts to event logs: AI
  - Informing/Passive Response
    a/o 1 Terminate a connection: Any
    a/o 1 Block a malicious host: Any
    a/o 1 Edit router's ACL to block hostile hosts: Any
    a/o 1 Passive monitoring of network traffic: Any
  - Defending/Active Response
    p/o 1 Alert based : AI
    a/o 1 send alerts to syslog: AI
    p/o 1 write the alert in a simple format(UIHRF) : WD
    a/o 1 send alerts to a *NIX socket that another program can
          listen on : AI
    a/o 1 Packets logged to a decoded ASCII format or a binary log
          file : AI
    a/o 1 send WinPopup alert messages to Windows machines: AI
    a/o 1 record log in a database: AI
    a/o 1 record log in a cvs file: AI
    a/o 1 sending alert to administrator's email or pager: AI
    a/o 1 log alerts to event logs: AI
    a/o 1 Watch Remote Procedure Call (RPC) protocol services for
          specific application/procedure calls : AI
    a/o 1 Search for a pattern in the Uniform Resource Indicator
          (URI) portion of packet: AI
    a/o 1 Terminate a connection: Any
    a/o 1 Block a malicious host: Any
    a/o 1 Edit router's ACL to block hostile hosts: Any
    a/o 1 dumps the application layer information for a given
          session: Any
    a/o 1 block web sites: Any
- Admin Console
  - Defending/Active Response_2
- IHMD Database
- Intrusion Unified Report Format
- IICS Protocols
    p/o 1 Based on IDMEF : IIHSM
    p/o 1 Any Wireless  compatible exchange format : IIHSM
  - Wireless IDXP
- Data Processing
    p/o 1 Decoding Raw data link layer packets [Pre-processor] : DP
    a/o 1 Ethernet 802.11*: DP
    a/o 1 Token Ring packets: DP
    a/o 1 Higher layer protocols such as IP, TCP, and UDP: DP
```

The above shows that Kismet IHS system has neither IICS nor any response systems to prevent the wireless network from possible intruders. Moreover, it lacks the most important subsystems in IHSMO, which are UIHRF and IIHSM. Other missing parts are, identification systems, including anomaly identification subsystem, response ontology, including active and passive response ontologies, attributes of live identification and attributes of identification database, which is part of misuse identification subsystem.

6.4 Conclusion

The ontology model of three most popular IHS systems, including Snort Wireless, Bro and Kismet IHSs was evaluated using IHSMO reference model. The above analysis clearly shows that none of the above wireless IHS systems are capable of securing a Wireless network against intruders or possible attacks. The evaluation of the ontology model of these systems precisely proves the lack of important concepts and slots which are already existed in the IHSMO reference model. These missing concepts and slots are extremely important to secure a WLAN against intruders.

An affective IHS system should be structured by following the intensive, comprehensive and effective structure of IHSMO reference model. Hence the IHSMO should be used as a comprehensive reference model to evaluate existing and development of future Wireless IHSs.

7. Conclusions and Future Work

7.1 Conclusions

This thesis has presented a novel IHS reference model for WLANs. Use of such reference model should allow the characterisation of different IHSs in a standardised and efficient manner. Thus, this approach is helping to build collaborating system of communicating IHSs which may also be viewed as a distributed super-IHS system. Such system will also be scalable and adaptable to changing conditions. Distributed systems have better traffic processing and handling capability, especially when presented with variable and increased traffic profiles. Additionally, there are less (or no) single points of failure.

An effective IHS system should be structured by following the intensive and comprehensive while effective structure of IHSMO reference model to achieve the most accurate, reliable, effective and efficient IHS system without compatibility issues. This new system will be used in a distributed *ad hoc* network since it is capable of communicating with other IHSs thorough IICS subsystem using UIHRF, IIHSM and IIHSCP. Hence the IHSMO should be used as a comprehensive reference model to evaluate existing and development of future Wireless IHSs. The earlier chapters have already established the hypothesis and provided the necessary qualitative and quantitative merits of the ontological approach to the development of the IHSMO. A substantial part of the thesis has been devoted to establishing facts and evaluation where appropriate. It is therefore important to succinctly summarise the findings of each relevant chapter in order to re-establish the original hypothesis and the context in which the research was carried out. Below is the summary of the conclusions derived from the thesis in a holistic manner.

In summary the following was achieved in this thesis:

- An extensive literature survey of recent research works on IHSs has been carried out. It identified inconsistency of the use of terminology. This justified the approach and the need to retain the same level of consistency in developing the model proposed in this thesis.

- An extensive literature survey of existing IHSs for WLANs have been carried out and their architecture is presented in a unified way. This established a common framework for comparison and benchmarking.
- Detailed comparison of selected IHSs was produced which further informed the evaluation of the proposed model.
- The thesis identified a distinct absence of a standardised reference model for IHSs, which justified the effort invested in the identification of such a model.
- As a result of the above, a Novel reference model for a scalable distributed IHS architecture is formally introduced which specifies the need for identification and response systems (with associated sets of methods) as well as inter IHS communications protocol.
- A mathematical model of IHS description for *ad hoc* networks with present intruder has been developed and a need for Unified Intrusion Handling Report Format (UIHRF), Inter IHS Message and Inter IHS Communication Subsystem (IICS) for inter IHS communications was identified.
- Ontology-based approach for modelling of IHS was suggested and formally adopted throughout.
- Intrusion Handling System Modelling Ontology (IHSMO) has been developed and their associated terminologies and formal axiomatic definitions have been defined.
- Ontologies of internal components such as Manager Console Ontology (MCO), Identification Subsystem Ontology (ISO), Response Subsystem Ontology (RSO), Inter IHS Communication Subsystem (IICS) Ontology and IHS Traffic Source Ontology (TSO) have been defined.
- It is concluded that using the Hozo ontology editor to build ontology for the IHSMO was the right choice that allowed both specification and verification of the integrity of the model developed.
- For the evaluation purposes the ontology-based models of three most popular IHSs (Snort Wireless, Bro and Kismet) have been built using Hozo ontology editor in order to compare IHSs in a unified way using IHSMO as a reference.

This enabled a functionality gap analysis of the important concepts and slots which are lacking in the three IHSs.

The main contributions of this research have been published in [Sale2005, Pakš2006, Sale2007, Sale2010a, Sale2010b] and it is available in appendix F.

7.2 Future Work

It can be noted that the concepts described in this research are focusing on architecture and ontological issues but not specifying implementation oriented issues. Therefore future work needs to be done on the following aspects:

1. Throughout this research it was assumed that IIHSM is protocol independent and can be used by any type of IIHSCP i.e. wireless versions of IDMEF communications protocol [Deba2002, Wood2007] (which does not exist yet) or Intrusion Detection Exchange Protocol [Fein2007]. Alternatively, UDP can be used as a message carrier, after IHSs form some kind of association and recognise each other. This would lead to a less energy overhead wireless communication protocol for IICS.
2. Obviously, definition of the IIHSMs' format is required. Generic format may look as shown in Figure 7.1.

Header: Sender IHS Identifier	Payload: Report data according to UIHRF

Figure 7.1: Proposed generic IIHSM format

3. Structure and functionality of the IHMD also has to be specified. This is important to efficiently store all the UIHRFs within RSDB subsystems.

References

[Aird2006] Airdefence IDS, Airdefence Inc, 2006. (http://www.airdefense.net/products/index.php)

[Amit2005] D. Amitava, “Fault-tolerant and energy-efficient permutation routing protocols for wireless networks”, *IEEE Transactions on Computers,* vol. 54, no. 11, pp. 1409-0421, Nov., 2005.

[Ande1980] J. Anderson, *"Computer Security Threat Monitoring and Surveillance".* James P. Anderson Co., Fort Washington, Pa., 1980.

[Arpi2001] J. C. Arpírez, O. Corcho, M. Fernåndez-López, A. Gómez-Pérez, “Web ODE: a Scalable Workbench for Ontological Engineering*",* *Proc. of the 1st international conference on Knowledge capture*, Victoria, British Columbia, Canada, October 22-23, 2001, pp6-13.

[Bake2005] Z. Baker, V. Prasanna, “High-throughput linked-pattern matching for intrusion detection systems”, *Proc. of the 2005 symposium on Architecture for networking and communications systems*, Princeton NJ, 2005, pp. 193 – 202.

[Bala1998] J.S. Balasubramaniyan, J.O. Garcia-Fernandez, D.Isacoff, E. Spafford, D. Zamboni, “An Architecture for intrusion detection using autonomous agents”, CERIAS Technical Report 98/05, June 11,1998.

[Basi2005] I. Basicevic, M. Popovic, V.Kovacevic, “The use of distributed network-based IDS systems in detection of evasion attacks”, *Proc. of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources (AICT/SAPIR/ELETE'05),* 2005, pp. 78-82.

[Bech2001] S. Bechhofer, I. Horrocks, C. Globe, R. Steven. “Oiled: a reasonable ontology editor for the semantic web”, *In Working Notes of 2001 Int, Description Logics Workshop (DL-2001)*, 2001, pp.1-9.

[Bern1996] A. Bernaras, I. Laresgoiti, J. Corera, ”Building and reusing ontologies for electrical network applications”, *Proc. of the 12th European Conference on Artificial Intelligence (ECAI),* 1996, pp.298-302.

[Blue2007] Bluetooth SIG Inc, "Bluetooth technology benefits", 2007. (http://www.bluetooth.com/Bluetooth/Learn/Benefits/)

[Bono2007] L. Bononi and S. Nikoletseas, *"Resources and mobility management in wireless networks"*,Wiley online library for Wireless Communications and Mobile Computing, Volume 8, Issue 3,2007,pp.277–405, (http://onlinelibrary.wiley.com/doi/10.1002/wcm.576/pdf)

[Cans2002] A. Cansian, A. Silva, M. Souza, "An attack signature model to computer security intrusion detection", *Proc. of IEEE Military Communications Conference, (MILCOM 2002),* Vol. 1, pp. 1368 – 1373.

[Chef2006] A.G. Chefranov, T.A. Mazurova, "Pseudo-random number generator RC4 period improvement", *Proc. IEEE International Conference on Automation, Quality and Testing and Robotics*, Cluj-Napoca, 2006, Vol.2, pp. 38-41.

[Cisc] Cisco Systems Inc, "Cisco Intrusion Prevention System Command Reference 5.0" (http://www.cisco.com/en/US/products/hw/vpndevc/ps4077/products_command_reference_chapter09186a00803a01c5.html)

[Clar2004] J. I. Clark, A. Ireland, J. Lomax, M.A. Harris and the Gene Ontology Consortium, "*Go on, Tell Us What You Think: Community Input into the Gene Ontology"*, The Gene Ontology Consortium. (http://www.geneontology.org/, 2008).

[Clau2002] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, "Optimised Link State Routing Protocol", *IETF Internet Draft*, July 2002.

[Cobb2004] C. Cobb, *"Cryptography for Dummies",* Hungry Minds Inc, U.S, 2004.pp. 12, 14-15, 39, 40, 43, 226-227

[Crac] Aircrack-ng Organisation, Aircrack documentation. (http://www.reng-edv.de/hotfixes/aircrack.html)

[Dali1997] H. Dalianis, F. Persson, "Reuse of an Ontology in an electrical Network Domain", *In AAA197 Spring Symposium Series, workshop on Ontological Engineering. California*,24-26 March,1997, pp.25-32.

[Deba2007] H. Debar, D. Curry, and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)", *RFC4765*, March 2007.

[Domi1999] J. Domingue, E. Motta, O.C. Garcia, "Knowledge Modelling in WebOnto and OCML", *A User Guide, Version 2.4*, The Open University, 1999, (http://kmi.open.ac.uk/projects/ocml/, 2005).

[Dona2002] S. Donald, R. McMillen, D. Ford, J. McEachen, "Therminator 2: A thermodynamics-based method for real-time patternless intrusion detection", *Proc. IEEE Military Communications Conference, (MILCOM 2002)*,Vol. 1, pp. 1498 – 1502.

[Dreg2004] H. Dreger, A. Feldmann, V. Paxson, R. Sommer, "Operational experiences with high-volume network intrusion detection", *Proc. of the 11th ACM Conference on Computer and Communications Security*, Washington DC, 2004. pp. 2 – 11.

[Fari2006] D. Faria, D. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints", *Proc. of the 5th ACM workshop on wireless security*, Los Angeles California, 2006, pp. 43 – 52.

[Fein2007] B. Feinstein, G. Matthews, "*The Intrusion Detection Exchange Protocol (IDXP)*", *RFC4767*, March 2007.

[Fens2000] D. Fensel , I. Horrocks , F.V. Harmelen , S. Decker , M.Erdmann , M. C. A. Klein, "OIL in a Nutshell", *Proc. of the 12th European Workshop on Knowledge Acquisition, Modelling and Management*, France, October 2000,pp. 1-16.

[Free2006] Free RADIUS Wiki Organisation, "IEEE802.11 Documentation", December 2006, (http://wiki.freeradius.org/IEEE_802.11).

[Genn1995] J.H. Gennari, R. B. Altman, M.A. Musen. "Reuse with PROTEGEII: from elevators to ribosomes", *Proc. of the 1995 Symposium on Software Reusability*, Seattle, Washington, U.S. 1995, pp.72-80

[Gaya2003] S. Gayal, S. A. V. Manickam, "*Wireless LAN Security Today and Tomorrow*". Centre for information and network security, Pune University, 2003, (http://www.madchat.org/reseau/wireless/wireless-lan-security.pdf).

[Geie2003] J.Geier, *WPA Security Enhancements*, March 2003. (http://www.wi-fiplanet.com/tutorials/article.php/2148721)

[Gene] M. R. Genesereth, "Knowledge Interchange Format draft proposed American National Standard": http://logic.stanford.edu/kif/dpans.html.

[Gill2006] R. Gill, J. Smith, A. Clark, "Specification-Based Intrusion Detection in WLANs", *22nd Annual Computer Security Applications Conference (ACSAC'06)*, Florida, U.S. 2006, pp.141 -152.

[Gowa2003] A. Gowans, "2.4GHz & 802.11G", March2003. (http://209.85.165.104/search?q=cache:dmsUz0jdqk4J:www.ieee802.org/18/Meeting_documents/2003_March/18-03-017r0_UK_update_on_OFDM_in_2.4GHz.ppt+DSSS+Modulation&hl=en&ct=clnk&cd=8&gl=uk).

[Grif1997] D. Griffin., G. Pavlou, T. Tin, "Implementing TMN-like management services in a TINA compliant architecture: a case study on resource configuration management", *Proc. of the 4th International Conference on Intelligence in Services and Networks*, 1997. pp. 263-274

[Gros1999] W.E. Grosso, H. Eriksson, R. W. Fergerson, J. H. Gennari, S.W. Tu, M. A. Musen, "Knowledge Modelling at the Millennium (The Design and Evolution of Protégé-2000)" *,Proc. of the 12 the International Workshop on Knowledge Acquisition, Modelling and Management* (KAW'99), Banff, Canada, October 1999.

[Grub1993] T.R. Gruber,"A translation approach to portable ontology specifications", *Knowledge Acquisition 5,* 1992. pp. 199-220.

[Grun2002] M. Gruninger, J. Lee, "Ontology Applications and Design", Introductory article to special issue on ontology engineering, Communications of the ACM, Vol.45, No2, February 2002, pp. 39-41.

[Harr1997] J. Harrington, D. Pritchard, 1997, "Concepts and applications of wireless security systems for tactical, portable, and fixed sites", *Proc. of The Institute of Electrical and Electronics Engineers 31st Annual 1997 International Carnahan Conference on Security Technology*, pp.133-139.

[Hass2002] Z.J. Haas, M.R. Pearlman, P. Samar, The Zone Routing Protocol (ZRP) for *Ad hoc* Networks, IETF Internet Draft, July 2002.

[Hils2000] D. Hilson, "IWN Networks - a TINA based systems software architecture to support global transactional initiatives", Tina Conference Paris, 2000, (http://www.tinac.com/conference/proceedings/day2/implementation/Hilson.doc)

[Hobe2008] S.Hoberman, "Ontology and Taxonomy, Design Challenge", *Information Management Magazine,* 2008, (http://www.information-management.com/issues/2007_47/10001177-1.html)

[Holl2004] T. Hollebeek, R. Waltzman, "The role of suspicion in model-based intrusion detection", *Proc. of the 2004 workshop on new security paradigms, New Security Paradigms Workshop,* Nova Scotia Canada, 2004. pp. 87 – 94.

[Huan2003]Y. Huang, W. Lee, "A cooperative intrusion detection system for *ad hoc* networks", *Proc. of the 1st ACM workshop on security of ad hoc and sensor networks,* Fairfax Virginia, 2003, pp. 135 – 147.

[Ihea2005] C. Iheagwara, A. Blyth, M. Bennett, "Architectural and Functional Issues in Systems Requirements Specifications for Wireless Intrusion Detection Systems Implementation", *Proc. of IEEE systems communications (ICW'05),* 2005. pp. 434-441

[Inte2006] Intel and 802.11, Intel Inc, 2006, (http://www.intel.com/standards/case/case_802_11.htm).

[Jame2006] J. Schoening, Standard Upper Ontology Working Group (SUO WG) Home Page, (http://suo.ieee.org/index.html, 2006).

[John2002] D.B. Johnson, D.A. Maltz, Y. Hu, J.G. Jetcheva, The Dynamic Source Routing Protocol for Mobile *Ad Hoc* Networks (DSR), *IETF Internet Draft*, Feb 2002.

[Kach2003] O. Kachirski, R. Guha, "Intrusion Detection Using Mobile Agents in Wireless *Ad hoc* Neworks", *Proc. of IEEE Workshop on Knowledge Media Networking*, 2003, pp. 153- 158.

[Karp1999] P.D. Karp, V.K. Chaudhri, E. Thomere, "XOL: An XML-Based Ontology Exchange Language", July1999,
(http://www.ai.sri.com/pkarp/xol, 2008)

[Kers2007] M. Kershaw, *Kismet Wireless documentation*, 2007.
(http://www.kismetwireless.net/documentation.shtml)

[Khan2006] R.A. Khan. "The Code Project: Comparison of IEEE 802.11a and IEEE 802.11b", 2006,
(http://www.codeproject.com/useritems/IEEE_WLAN_Standards.asp).

[Kife1995] M. Kifer, G. Lausen, J. Wu, "Logical Foundations of Object-Oriented and Frame-Based Languages" Journal of the ACM, Vol 42,No 4, May 1995, pp. 741-843.

[Kim2004] M. Kim, H. Kang, S. Hong, S. Chung, J. Hong, "A flow-based method for abnormal network traffic detection", *IEEE/IFIP Network Operations and Management Symposium, NOMS 2004.* Vol. 1, pp.599–612.

[Kita1996] Y. Kitamura, M. Sasajima, M. Ikeda, and R. Mizoguchi, "A method of qualitative reasoning for mode-based problem solving and its application to a nuclear plant", *International J. of Expert Systems with Applications, Vol. 10, No. 3/4,* 1996, pp.441-448.

[Kita2005] Y. Kitamura, Y. Koji, R. Mizoguchi, "An Ontological Model of Device Function and Its Deployment for Engineering Knowledge Sharing", *In Proc. of the First Workshop FOMI 2005 - Formal Ontologies Meet Industry, Castelnuovo del Garda (VR)*, Italy, June 9-10, 2005, CD-ROM (Reformatted version).

[Klyn2004] G. Klyne, J. J. Carroll, "Resource Description Framework (RDF) Concepts and Abstract Syntax", *W3C Recommendation,* 10 February 2004,(http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/, 2008).

[Kong2003] J. Kong, X. Hong, M. Gerla, "New Set of Passive Routing Attacks in Mobile", *Proc. of IEEE Military Communications Conference, MILCOM 2003,* Vol. 1, pp. 796 – 801.

[Krug2000] C. Krugel, T. Toth, "A survey on Intrusion Detection Systems", Technical Report TUV-1841-00-11, Technical University of Vienna, Information Systems Institute, Distributed Systems Group.

[Lee2000] W. Lee, S. J. Stolfo, "A framework for Constructing Features and Models for Intrusion Detection Systems", *ACM Transactions on Information and System Security,* Vol. 3, No. 4, 2000. pp. 237-261.

[Lena1990] D. B. Lenat, R. V. Guha, K. Pittman, D. Pratt, M. Shepherd, "CYC: Toward Programs With Common Sense", *Communication of the ACM,* Vol. 33, No. 8, 1990, pp. 30-49.

[Lena1995] D. B. Lenat, "CYC: A Large-Scale Investment in Knowledge Infrastructure", *Communications of the ACM,* Vol.38, No.11, Nov. 1995, pp.33-38.

[LBNL2007] Lawrence Berkeley National Laboratory, *Bro Intrusion Detection System,* Berkeley, CA, USA, 2007, (http://www.bro-ids.org/Features.html).

[Lim2003] Y. Lim, T. Schmoyer, J. Levine, and H. L. Owen, "Wireless Intrusion Detection and Response". *Proc. 2003 IEEE Workshop on Information Assurance,* United States Military Academy, 2003, pp 68-75.

[Lin1996] J. Lin, M.S. Fox, T. Bilgic, "A requirement ontology for engineering design", *Concurrent Engineering: Research and Applications,* Vol. 4, No4, Sept1996, pp.279-291.

[Liu2006] Y. Liu, C. Comaniciu, H. Man, "A Bayesian Game Approach for Intrusion Detection in Wireless *Ad hoc* Networks", *Proc. 2006 Workshop on game theory for communications and networks,* ACM

International Conference Proceeding Series, 2006.Vol. 199, Pisa Italy, pp. 1-12.

[Lock2003] A. Lockhart, *Snort Wireless User Guide*, Sourcefire Inc, 2003. (http://snort-wireless.org/docs/usersguide/)

[Luke2000] S. Luke, J. Hein, SHOE 1.01 Proposed Specification, show project, 28 April 2000, (http://www.cs.umd.edu/projects/plus/SHOE/, 2005).

[MacG1991] R. M. MacGregor, "Using a description classifier to enhance deductive inference", In Proc. Seventh IEEE Conference on AI Application, Florida, 1991, pp. 141-147.

[Madw] Madwifi Organisation, Madwifi Project. (http://madwifi.org/wiki/ngFeatures)

[Maed2000] A. Maedche, H.P. Schnurr, S. Staab, R. Studer, "Representation Language-Neutral Modeling of Ontologies", *Proc. of the German Workshop "Modellierung-2000"*, Koblenz, Germany, April, 5-7, 2000.

[Math2005] S. Mathew, D. Britt, R. Giomundo, S. Upadhyaya, M. Sudit, A. Stotz, "Real-time multistage attack awareness through enhanced intrusion alert clustering", *Proc. IEEE Military Communications Conference, MILCOM2005*,vol. 24, pp. 12-24.

[Mart2000] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehaviour in Mobile *Ad hoc* Networks", *Proc. Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking*, 2000, pp. 255-265.

[Mian2003] P. G. Mian, R.A. Falbo, "Building Ontologies in a Domain Oriented Software Engineering Environment", *IXA Argentine Congress on Computer science(CACIC 2003),* La Plata, Argentine, 6-10 October 2003,pp. 930-941.

[Mian2004] P. G. Mian, R.A. Falbo. "Supporting Ontology Development with ODEd", *Journal of the Brazilian Computer Society,* Vol.9, issue2, 2004, pp.57-76.

[Milo2003] T. Milo, S. Abiteboul, B. Amann, O. Benjelloun, F. Dang Ngoc. "Exchange Intensional XML Data", *SIGMOD 2003*, San Diego, CA, 9-12 June 2003, pp.289-300.

[Mizo, 1992] R. Mizoguchi, "Task ontology and its use in a task analysis interview systems - Two-level mediating representation in MULTIS –", *Proc. of the JKAW'92*,1992, pp.185-198.

[Mizo1993] R. Mizoguchi, "Knowledge Acquisition and Ontology", *Proc. of the KB&KS'93*, Tokyo, 1992, pp.121-128.

[Mizo1997] R. Mizoguchi, M. Ikeda, "Towards ontology engineering", *Proc. Joint 1997 Pacific Asian Conference on Expert Systems/Singapore International Conference on Intelligent Systems*, 1997, pp.259-266.

[Mizo2004] R. Mizoguchi, "The Hozo Ontology Editor User Manual", *Mizoguchi Lab., the Institute of Scientific and Industrial Research, Osaka University,http://www.ei.sanken.osakau.ac.jp/hozo/eng/manual/manual_en.html#c7.*

[Mizo2005] R. Mizoguchi, "The Role of Ontological Engineering for AIED Research", *Computer Science and Information Systems*, Vol 2, No 1, 2005, pp. 31-42.

[Mukh1994] B. Mukherjee, L. T. Heberlein, K. N. Levitt, "Network Intrusion Detection", *IEEE Network 1994*, Vol 8, No 3, pp. 26-41.

[Muse1992] M.A. Musen, "Dimensions of knowledge sharing and reuse", *Computers and Biomedical Research 25*, 1992, pp. 435-467.

[Mylo1990] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis, "Telos: Representing knowledge about information systems", *In ACM Trans. on Information Systems*, Vol. 8, No.4, 1990, pp. 325-362.

[Ndis] NDIS Inc, Network, NDIS developer's reference.
(http://www.ndis.com/)

[Ning2002] P. Ning, Y. Cui, D. Reeves, "Constructing Attack Scenarios Through Correlation of Intrusion Alerts", *Proc. 9th ACM Conference on Computer and Communications Security*, Washington DC, 2002, pp. 245 – 254.

[Noy2001] N.F. Noy, D.L. McGuinness, "Ontology Development 101:A Guide to Creating Your First Ontology", *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880,* Stanford University, March 2001.

[Ossm2004] M. Ossmann, "*An Article on WEP Cracking Tools, WEP: Dead Again*", 2004. (http://www.securityfocus.com/infocus/1814)

[Pakš2006] A. Pakštas, S. Salekzamankhani, B.Virdee, "Fighting Intrusions in Wireless LANs: A Need for the Reference Model". *Proc. 2nd IEEE and IFIP International Conference in Central Asia on the Next Generation of Mobile, Wireless and Optical Communications Networks, (ICI 2006)*, Tashkent, Uzbekistan , Sep. 19, 2006, ISBN: 1-4244-0543-2.

[Perk2002] C. Perkins, E.M. Belding-Royer, S. Das ,"*Ad hoc* On-demand Distance Vector (AODV) Routing", *IETF Internet Draft*, Jan 2002.

[Prov2007] IBM Internet Security Systems, "*IBM Proventia Network Anomaly Detection System*", 2007. (http://www.iss.net/products/Proventia_Network_Anomaly_Detection_System/product_main_page.html)

[Rahm2006b] M.A Rahman, A. Pakstas, F. Z. Wang, "Towards Communications Network Modeling Ontology for Designer and Researchers", 10th IEEE International Conference on Intelligent Engineering Systems 2006(INES2006), London, June 26-28, 2006.

[Refe2007] Lexico Publishing Group, LLC, 2007. (http://www.reference.com/browse/all/)

[Ribi2000] M. Ribière and P Charlton, "Ontology Overview from Motorola Labs with a comparison of ontology languages", December, 2000, (http://www.fipa.org/docs/input/f-in-00045/f-in-00045.pdf).

[Sale2005] S. Salekzamankhani, A. Pakštas, B.Virdee, "Towards Development of a Reference Model for Intrusion Detection Systems for Wireless LANs", *IEEE Globecom 2005, Workshop on Adaptive Wireless Networks, AWIN* Nov. 30, 2005, San Louis, USA.

[Sale2007] S. Salekzamankhani, A. Pakštas, "Why we need a reference model for intrusion handling systems for Wireless LANs?" *Telecommunications and Computer Networks, SoftCOM2007, 15th International Conference,* 27-29 Sept, 2007 pp.1– 6. ISBN: 978-953-6114-95-5.

[Sale2010a] S. Salekzamankhani, A. Pakstas, B. Virdee, "An Ontology Approach to Construction of Identification Subsystem for Intrusion Handling Systems in Wireless LANs", World Congress on Engineering (WCE), the 2010 International Conference of Wireless Networks (ICWN'10), 30 June - 2 July 2010, London, UK, ISBN: 978-988-17012-9-9.

[Sale2010b] S. Salekzamankhani, A. Pakstas, B. Virdee, "Ontology Approach to Construction of Response and Management Console Subsystems for Intrusion Handling Systems in Wireless LANs", World Congress on Engineering (WCE), the 2010 International Conference of Wireless Networks (ICWN'10), 30 June - 2 July 2010, London, UK, ISBN: 978-988-17012-9-9.

[Sasa 1995] M. Sasajima, Y. Kitamura, M. Ikeda, and R. Mizoguchi, "FBRL: A Function and behaviour representation language", *Proc. of IJCAI-95, Montreal,* 1995, pp. 1830-1836.

[Scho2002] G. Scholz, "An Architecture for Securing Wireless Networks". *The Internet Protocol Journal,* Cisco Systems, 2002, Vol 5, No 3, pp.17-22,

[Schr1994] A.T. Schreiber 1, B.J. Wielinga, J.M. Akkermans, W. Van De Velde, "CML-The CommonKADS Conceptual Modelling", In Proc. of the EKAW'1994, September 1994, pp. 1-25.

[Shad1993] N. Shadbolt, E. Motta, A. Rouge, "Constructing knowledge-based systems", *IEEE Software,* Vol.10, No. 6, Nov 1993. pp 34 – 38.

[Shan2003] M. Shankar, N. Rao, S. Batsell, "Fusing Intrusion Data for Detection and Containment", *Proc. IEEE Military Communications Conference, MILCOM 2003,* pp. 741 – 746.

[Seka1999] Sekar, R, Guang, Y, Verms, S. Shanbhac, "A High-performance Network Intrusion Detection System", *Proc. 6th ACM conference on*

Computer and communications security, Kent Ridge Digital Labs Singapore, 1999. pp. 8-17.

[Seta1996] K. Seta, M. Ikeda, O. Kakusho, and R. Mizoguchi, "Design of a conceptual level programming environment based on task ontology", *Proc. of the First International Conference on Success and Failures of Knowledge-Based Systems in Real-World Applications, Bangkok*,1996, pp.11-20.

[Somm2003] R. Sommer, V. Paxson, "Enhancing Byte-level Network Intrusion Detection Signatures with Context", *Proc. 10th ACM Conference on Computer and Communications Security*, Washington DC, 2003. pp. 262 – 271.

[Spen2000]W. Spencer, *"Network Security Assessment"*, 2000. (http://www.nsai.net)

[Sper2000] C. Sperberg-McQueen, J. Paoli, E. Maler, T. Bray, "Extensible Markup Language (XML) 1.0 (Second Edition)", *World Wide Web Consortium*, October 2000, (http://www.w3.org/TR/2000/REC-xml-20001006).

[Stam2005] I. Stamouli, P. Argyroudis, and H.Tewari "Real-time Intrusion Detection for *Ad hoc* Networks", *Proc. Sixth IEEE international symposium on World of Wireless Mobile and Multimedia Networks*, 2005, pp. 374-380.

[Step2004] R. Stephan, P. Ray, N. Paramesh, "Knowledge Based Fault Management for OSI Networks". *International Journal of Network Management*, 2004,Vol. 14, No 1, pp.59–73.

[Stum] Netstumber Organisation, Network Stumbler Archives, 2007, (http://www.netstumbler.com/).

[Sun2003] B. Sun , K. Wu, &, U. Pooch, "Alert Aggregation in Mobile *Ad hoc* Networks", *Proc. 2003 ACM Workshop on Wireless Security*, San Diego CA, 2003, pp.69 -78.

[Tane2003] A. S. Tanenbaum, "*Computer Networks*", Prentice Hall, 2003.

[Tanz2002] F. Tanzella, "*Wireless LAN Security – How to Secure WLANs*", November 2002,

(http://airdefense.net/wirelesslansecurity/wlan_security_whitepaper.html)

[Tech2001] Searchnetworking Organisation, *"IEEE 802.11 Modulation Technique"*, 2001, (http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci525721,00.html).

[Tech2006] Microsoft Technet, "Wi-fi Modulation: Configuring Windows XP IEEE 802.11 Wireless Networks for the Home and Small Business", March2006, (www.microsoft.com/technet/prodtechnol/winxppro/maintain/wifisoho.mspx).

[TISN] Trusted Information Sharing Network (TISN) organisation, "Wireless Security – Information for CIOs", 2007. (http://www.tisn.gov.au/agd/WWW/rwpattach.nsf/VAP/(930C12A9101F61D43493D44C70E84EAA)~Wireless+CIO+(FINAL).doc/$file/Wireless+CIO+(FINAL).doc)

[Toh2001] C.K. Toh, *"Ad Hoc Mobile Wireless Networks: Protocols and Systems"*, Prentice Hall, 2001, ISBN 0-13-007817-4

[Unde2003] J. Undercoffer, A. Joshi, J. Pinkston, *"Modeling Computer Attacks: An Ontology for Intrusion Detection"*, G. Vigna, E. Jonsson, and C. Kruegel (Eds.): RAID 2003, LNCS 2820, Springer-Verlag Berlin Heidelberg 2003, pp. 113–135.

[Usch1995] M. Uschold, M. King, "Towards a Methodology for Building Ontologies", *Technical Report, AIAI-TR-183*, University of Edinburg, Edinburg, July 1995, (Presented at the workshop on basic ontological issues in knowledge sharing, IJCAI95, Montreal), (http://www.aiai.ed.ac.uk/~enterprise/enterprise/ ontology.html, 2005).

[Vign2003] G. Vigna, F. Valeur, R. Kemmerer, "Designing and Implementing a Family of Intrusion Detection Systems", *Proc. 9th European Software Engineering Conference 11th ACM SIGSOFT International Symposium* *on Foundations of Software Engineering*, Helsinki, Finland, 2003, pp. 88 – 97.

[Vign2004] G. Vigna, , W. Robertson, D. Balzarotti, "Testing Network-based Intrusion Detection Signatures using Mutant Exploits", *Proc. 11th ACM conference on computer and communications security*, Washington DC, 2004, pp. 21 – 30.

[Vlad2004] A. Vladimirov, K. Gavrilenko,A. Mikhailovsky, "*WI-FOO The Secrets of Wireless Hacking.* " , Addison Wesley, 2004, pp. 26-33.

[Wade1997] V. Wade, "A Methodology for Developing Integrated Multi-domain Service Management Systems", *Proc. 4th International Conference on Intelligence and Services in Networks: Technology for Cooperative Competition*, 1997. pp. 245-254.

[Wagn2002] D. Wagner, P. Soto, "Mimicry Attacks on Host-based Intrusion Detection Systems", *Proc. 11th ACM conference on Computer and communications security,* Washington DC, 2002, pp. 255 – 264

[Watk2004] D. Watkins, C. Scott, "Methodology for Evaluating the Effectiveness of Intrusion Detection in Tactical Mobile *ad hoc* networks", *Proc. IEEE Wireless Communications and Networking Conference,* WCNC 2004 Vol. 1, pp. 623 – 628.

[Wave2001]Wave Report, IEEE 802.11 standard tutorial, November 2001 (http://www.wave-report.com/tutorials/ieee80211.htm)

[Wifi1999] Wi-Fi Alliance organisation, "*Get to know Alliance*", 1999. (http://www.wi-fi.org/about_overview.php)

[Wood2002] M.Wood, M.Erlinger, "Intrusion Detection Messassge Exchange Requirements draft-ietf-idwg-requirements-10", Internet Draft, October 2002, (http://tools.ietf.org/html/draft-ietf-idwg-requirements-10).

[Wood2007] M. Wood, M. Erlinger, *"Intrusion Detection Message Exchange Requirements", RFC4766,* March 2007.

[Yaro2000] F. Yarochkin, "*Snortnet - A Distributed Intrusion Detection System*", Kyrgyz Russian Slavic University, Bishkek, Kyrgyzstan, 2000.

[Yee2003] A. Yee, "The Intelligent IDS: Next Generation Network Intrusion Management Revealed", 2003, pp. 1-11. (http://www.nfr.com/resource/downloads/The_Intelligent_IDS.pdf)

[Yeo2004] J. Yeo, M. Youssef, A. Agrawala, "A Framework for Wireless LAN Monitoring and its applications", *Proc. of the 2004 ACM workshop on wireless security*, Philadelphia PA, 2004, pp. 70 – 79.

[Zhan2000] Y. Zhang, W. Lee, "Intrusion Detection in Wireless *Ad hoc* Networks", *Proc. 6th International Conference on Mobile Computing and Networking*, Boston, Massachusetts, U.S., 2000, pp. 275 – 383.

[Zhan2001] Y. Zhang, Q. Depei, B. Chongming, W. Weiguo, G. Xiaobing, "Multi-agent Based Intrusion Detection Artchitecture", *IEEE Proc. 2001 International Conference on Computer Networks and Mobile Computing, (ICCNMC'01),* 2001, pp. 494.

[Zhan2003] Y. Zhang, W. Lee, "*Intrusion Detection Techniques for Mobile Wireless Networks*", Kluwer Academic publishers, 2003, pp. 545 – 556.

Appendix A. Security in Wireless Networks

The main problem with WLANs is due to its nature because intrusion may happen without intruder no longer gaining physical access to network in order to compromise its security. Here intrusion means any set of actions that attempt to compromise the integrity, confidentiality or availability of a WLAN by an unauthorised access. This would create new vulnerabilities that do not exist in traditional wired networks; hence many of the proven security measures turn out to be ineffective [Sale2007, Sale2005].

A.1 Overview of Wireless Environment

The IEEE 802.11 networking standard was released in 1997 [Wave2001]. This standard defined wireless networks offering data rates of 1 and 2 Mbps, which could be achieved using Infrared or radio communication. Wireless equipment getting access to the network, a medium access control protocol used in IEEE802.11 is a Collision Sense Multiple Access with Collision Avoidance (CSMA/CA) which uses a lot of the channel's raw capacity. Theoretically a maximum raw data rate of 2 Mbps defined by the 802.11 standard is reduced to 1 Mbps throughput due to CSMA/CA protocol. Although this standard was the original wireless networking standard, and manufacturers put their products on the market, there was one main drawback preventing it from proper acceptance and this was the issue of product interoperability. Products manufactured according to IEEE802.11 were not guaranteed to work with products from other manufacturers. What caused very low acceptance by the market.

The wireless fidelity (Wi-Fi) originally called the Wireless Ethernet Compatibility Alliance (WECA) is a trade group formed in 1999 with a purpose of certifying equipment related to wireless networking in order to ensure compatibility between vendors. Once a wireless networking appliance has been given the "Wi-Fi Certified" logo, it is certified by the Wi-Fi Alliance [Wifi1999] to be compatible with equipment from other wireless network equipment manufacturers.

Although there have been many changes to the original IEEE 802.11 standard, the three most important changes are amendments to the original, and these are the IEEE 802.11a, 802.11b and 802.g amendments. These amendments are also referred to as standards, although they are not individual standards themselves. Currently IEEE 802.11 is the most widely used WLAN technology.

Problems with the original IEEE802.11 were resolved with the amendment to 802.11 a, b and g. IEEE 802.11b which was released in 1999 [Wave2001]. The IEEE802.11 defined products that were made under this standard had an indoor range of 30 meters, an outdoor range of 90 meters, which in turn decreases the network data rate to 1 Mbps in the case of outdoor usage. As this amendment also used CSMA/CA, actual network throughput was still affected and reduced data rates to 5.9Mbps over TCP and 7.1 Mbps over UDP [Wave2001]. Another enhancement was Adaptive Rate Selection defined by IEEE802.11b which means that in the event of signal problems, the data rate would reduce appropriately from 11 Mbps, then 5.5, then 2, then 1. Acceptance of products using this specification was much more positive than the original standard, as data rates were better, and the Wi-Fi Alliance [Wifi1999] had given their approval for compliance of interoperability. Similarly to the original standard it also uses the 2.4GHz band, which is also used by microwave ovens, cordless telephones and Bluetooth [Blue2007] and potentially can produce interference when such similar devices are used in close proximity with each other.

The next amendment was the IEEE 802.11a specification, which was released at approximately the same time as the IEEE 802.11b amendment [Wave2001]. This new release came with a maximum data rate of 54Mbps, with throughput about 24.7 Mbps, and Adaptive Rate Selection decreasing data rate to 48 Mbps, 36, 24, 18, 12, 9 and then 6. This specification used 12 wireless network channels to communicate on, which the product could be made to use, giving a lot of spectrum space to use, and also used the 5 GHz frequency band, so interference was minimal as no household products such as microwave ovens used the same band. Products made using this amendment were not compatible with equipment using the IEEE 802.11b amendment as written in the specification, but manufacturers made equipment that was compatible with both amendments and so that would make them work with each other. Although this amendment made products better in performance than the original standard, the IEEE 802.11b amendment had good success with global acceptance and so products under this specification did not have a great amount of success. To combat this low acceptance, manufacturers decided to make products that could use a combination of the IEEE 802.11x standards, meaning IEEE 802.11a, 802.11b or later 802.11g. This was a successful idea and acceptance grew [Tane2003].

The IEEE released 802.11g in June 2003 which was another amendment to the original standard. This specification uses the 2.4 GHz band, the same network access method, as in CSMA/CA, and its Adaptive Rate Selection is from 54 Mbps to 48, then 36, 24, 18, 12, 9 and then 6. As this amendment uses the same frequency band as IEEE 802.11b, devices made under IEEE 802.11g are compatible with those made under IEEE 802.11b [Tane2003].

Thus as with any of the other 802 networking standards (Ethernet, Token Ring, etc.), the 802.11 specification affects the lower layers of the OSI reference model, the Physical and Data Link layers shown in Figure A.1 [Tanz2002]. The IEEE 802.11 MAC is common to all IEEE 802.11 PHY layers and specifies the functions and protocols required for control and access. The MAC layer is responsible for managing data transfer from the upper-level functions to the physical media.

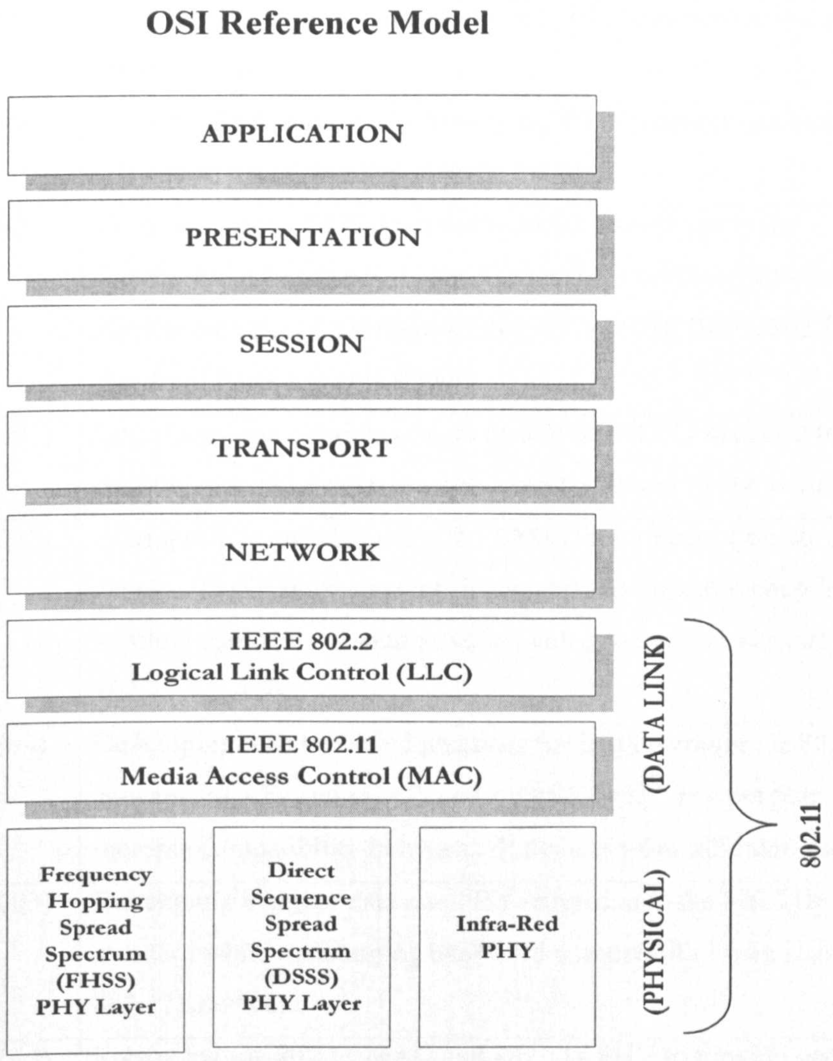

OSI Reference Model
APPLICATION
PRESENTATION
SESSION
TRANSPORT
NETWORK
IEEE 802.2
Logical Link Control (LLC)
IEEE 802.11
Media Access Control (MAC)
Frequency Hopping Spread Spectrum (FHSS) PHY Layer
Direct Sequence Spread Spectrum (DSSS) PHY Layer
Infra-Red PHY
(DATA LINK)
(PHYSICAL)
802.11

Figure A.1. IEEE 802.11 standards mapped to the OSI reference model focuses on the bottom two levels of the OSI model: PHY and MAC [Tanz2002]

Table A.1 lists the variants of 802.11 standards and provides an overview of the document contents [Wave2001, Tane2003, Free2006]. Table A.2 also shows a comparison of the physical layer of 802.11 standards.

Table A.1. Overview and timeline of IEEE 802.11 Variants

802.11 Variant	**Description**
802.11a (1999)	A standard for WLAN operations in 5GHz band with a data rates of up to 54Mbps.
802.11b (1999)	A standard (also known as Wi-Fi) for WLAN operations in the 2.4 GHz band with data rates of up to 11Mbps.
802.11c (2001)	Documentation of 802.11-specific MAC procedures to the *International Organisation for Standardisation/International Electro-technical Commission* (ISO/IEC) 10038 (IEEE 802.1d) standard. Work has completed.
802.11d (2001)	Definitions and requirements to enable the 802.11 standard to operate in countries that are not currently served by the standard.
802.11e (2005)	Attempting to enhance the 802.11 MAC for a better *Quality of Service* (QoS). Improvements in capabilities and efficiency in order to allow applications such as voice, video, or audio transport over 802.11 WLANs.
802.11f (2003)	Developing recommended practices for implementing the 802.11 concepts of APs and *distributed systems* (DSs). The purpose is to increase compatibility between AP devices from different vendors.
802.11g (2003)	Developing a higher data rate PHY extension to the 802.11b standard while maintaining backward compatibility with current 802.11b devices.
802.11h (2003)	Enhancing the 802.11 MAC and 802.11a PHY to provide network management and control extensions for spectrum and transmit power management in the 5GHz band in order to allow regulatory acceptance of the standard in some European countries.
802.11i (2004)	Enhancing the security and authentication mechanisms.
802.11n (2008)	Defining modifications to the Physical and Media Access Control Layer (PHY/MAC) that deliver a minimum of 100 Mbps throughput at the MAC SAP (service access point).

Table A.2. Comparison of IEEE 802.11 PHYs and Throughputs

802.11 Variant	***Frequency/ Wavelength***	***Modulation Encoding***	***Maximum data rate /Throughput (Mb/s)***	***Minimum Throughput at MAC SAP***	***Indoor Range***	***Outdoor Range***
802.11 (1997)	2.4 GHz ISM	DSSS[3]	2	<1	~25	~75
802.11 (1997)	2.4 GHz ISM	FHSS[4]	2	<1	~25	~75
802.11 (1997)	850-950nm (IR)	Diffuse IR[5]	2	<1	~25	~75
802.11a (1999)	5 GHz ISM	OFDM[6]	54	25-27	~30	~100
802.11b (1999)	2.4 GHz ISM	DSSS[7]	11	4-5	~35	~110
802.11g (2003)	2.4 GHz ISM	OFDM	54	25-27	~35	~110
802.11n (2008)	2.4 GHz or 5 GHz	?	540	200	~50	?

The next generation wireless applications will require higher WLAN data throughput with a longer range. To achieve this, the 802.11n task group is set in order to define modifications to the Physical Layer and Media Access Control Layer (PHY/MAC) that deliver a minimum of 100 Mbps throughput at the MAC SAP (service access point). This minimum throughput requirement approximately quadruples WLAN throughput performance compared to existing 802.11a/g networks. Over-the-air throughput is targeted to be in exceed of 200 Mbps to meet at least the 100 Mbps MAC SAP throughput requirements. Other necessary improvements include range at given throughputs, robustness to interference, and an improved and more uniform service within the coverage of an access point (Basic Service Set – BSS). Wider

[3] Direct Sequence Spread Spectrum

[4] Frequency Hopping Spread Spectrum

[5] Diffuse infrared

[6] Orthogonal Frequency-Division Multiplexing

[7] Direct Sequence Spread Spectrum

bandwidth channels and multiple antenna configurations could lead to data rates of 500 Mbps [Inte2006, Khan2006].

A.2 A model of WLAN attack's structure

The following is a breakdown of the different phases in a network based system compromise, categorising the methods attackers use to gain access to a system and to hide their traces. An attack leading to a system compromise is conducted in phases which start out with reconnaissance.

A.2.1 Reconnaissance

The first phase is *reconnaissance*. This comes in two flavours, *passive* and *active*. The first step to attacking a wireless network involves finding a network to attack. The first popular software to identify wireless networks was NetStumbler. NetStumbler is a Windows application that listens for information, such as the Service Set Identifier (SSID), which is a sequence of up to 32 letters or numbers that is the ID, or name, of a WLAN and being broadcasted from WAPs that have not disabled the broadcast feature. When it finds a network, it notifies the person running the scan and adds it to the list of found networks.

- Passive reconnaissance: is watching network traffic to discover hosts, addresses, resources, users, etc. It is very hard to detect passive network sniffing. Luckily the ability to do passive sniffing requires close proximity to the target network, or a network adjacent to the target network.

- Active reconnaissance: is where the attacker actively sends data to your network in an attempt to learn its extent, hosts, users, vulnerabilities, and so on. Tools like nmap have powerful features both for exploring networks and for obscuring the source of the attacks. Such tools however are not always necessary. Ping, Telnet, and finger can be used. E-mail can also be used.

A.2.2 Compromise

After reconnaissance comes the compromise phase, at this point the attacker knows some specific information about your network and the systems on it. He, or she, has selected one or more known vulnerabilities. He will now exploit one or more of these weaknesses to compromise one or more accounts on the system.

A.2.3 Obfuscation

The next phase is obfuscation. Here the attacker will try to eliminate all evidence of the intrusion.

A.2.4 Entrenchment

The next phase is entrenchment. Here the attacker will both enlarge the set of compromised accounts and systems and create additional holes to come through if the initial routes are found and blocked. At this point you are pretty much done for. Your only chance to get the attacker out and keep them out is to disconnect from the Net, wipe all the machines, and install from scratch, being sure this time to apply all of those annoying security updates from your vendors.

Obviously, the earlier you can detect and head off an attack, the better. Once the entrenchment phase has been reached, it is very difficult to get the intruders out, because they generally have all the power your system administrators have, and have probably compromised multiple systems in multiple ways. The only certain way to get rid of them is to disconnect from the outside world and start from scratch, loading everything off read-only media. Even this is problematic if you have data and code you must restore from backups. Unless you know exactly when your system was compromised and have a backup that hasn't been mounted since before that date of compromise, you have no trustworthy copy of that code or data.

A.3 WLANs Security Threats Classification

WLANs devices are not isolated. They trust each other in some fashion. It might be the almost blind faith or through a sophisticated cryptographic verification used by the Kerberos authentication system in which case a particular user will be trusted. It doesn't matter if the intruder can compromise the system, eventually it would be able to attack other systems, by taking over either *root*, and hence the system's identity, or some user account.

A.3.1 Password Related Threats

Authentication is the process of *proving* someone's identity. This is distinct from the assertion of identity known as *identification*, and from deciding what privileges to be credited with that identity, known as *authorization*. While all three are important, authentication is the trickiest from the perspective of network security. We are concerned here with two forms of authentication, that of a user to a machine during an initial login sequence, and machine to machine authentication during operation. Solutions to the first problem are typically categorised as "something you know," "something you have," and "something you are." Machine to machine authentication is generally divided into two types: "cryptographic" and "other."

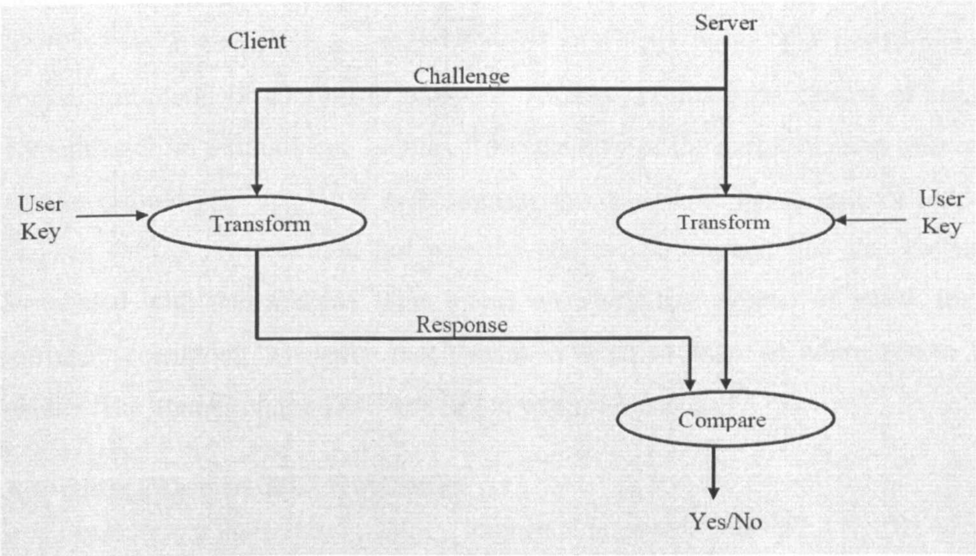


Figure A.2. Challenge Response Authentication

System bugs are the exciting way to crack a system, but they are not the most common attack. Rather it's a more mundane feature, the user passwords, which is most often attacked. A high percentage of system penetrations occur because of the failure of the entire password authentication system. These are called *dictionary attacks*, and they are usually very successful. Make no mistake about it. If your password file falls into enemy hands, there is a very high probability that your network *will* be compromised.

A.3.2 Authentication Related Threats

Computers often trust each other. This is well and good for machines under common control; indeed, it is often necessary to their usability. But the web of trust often spreads far wider than it should. It is a major part of a security administrator's job to ascertain and control which machines trust which, for what, and by what mechanisms. The address based mechanisms used by many of the standard protocols are inadequate in high threat environments.

For better or worse, the dominant form of a host to host authentication today relies on the network itself. That is, the network itself conveys not just the remote user's identity, but is also presumed to be sufficiently accurate that one can use it as an authenticated identity. As we have seen, this is dangerous. Network authentication itself comes in two flavours, address based and name based. For the former, the source's numeric IP address is accepted. Attacks on this form consist of sending something from a fraudulent address. The accuracy of the authentication thus relies on the difficulty of detecting such impersonations. Name based authentication is weaker still. It requires that not just the address be correct, but also the name associated with that address. This opens up a separate avenue of attack for the intruder: corrupting whatever mechanism is used to map IP addresses to host names. The attacks on the DNS attempt to exploit this path.

A.4 Categories of WLAN Attacks

It is necessary to understand why it's important to secure your WLAN. The WLAN is subject to the same network attacks to which any wired network could be vulnerable. However, there are some attacks that are specific to a WLAN that would not be possible if it weren't broadcasted through the air.

A WLAN is susceptible to attacks ranging from passive eavesdropping to active interfering [Zhan2003].

In Passive attack an unauthorised user gains access to important information without any modification to these assets. Eavesdropping and Traffic analysis (sometimes called traffic flow analysis) are examples of a passive attack. These attacks are described below [TISN].

- **Eavesdropping:** The attacker monitors transmissions for message content. An example of this attack is a person listening into the transmissions on a LAN between two workstations or tuning into transmissions between a wireless handset and a base station.
- **Traffic analysis:** The attacker, in a more subtle way, gains intelligence by monitoring the transmissions for patterns of communication. A considerable amount of information is contained in the flow of messages between communicating parties.

In an active attack the unauthorised user would make deletion or modification to data stream, message or file. Active attacks would be in one of the following forms or combination of them:

Masquerading, Replay, message (data) modification, denial of service (DoS) and Repudiation. These attacks are defined below [TISN].

- **Masquerading:** The attacker impersonates an authorised user and thereby gains certain unauthorised privileges.
- **Replay:** The attacker monitors transmissions (passive attack) and retransmits messages as the legitimate user.
- **Message modification:** The attacker alters a legitimate message by deleting, adding to, changing, or reordering it.
- **Denial-of-service (DoS):** The attacker prevents or prohibits the normal use or management of communications facilities.
- **Repudiation:** Repudiation is an active attack on non-repudiation by either the source or the destination where the source entity denies sending a message or the destination entity denies receiving a message. Basic 802.11 securities do not have non-repudiation. Without non-repudiation, the source entity can deny ever having sent a message and the destination entity can deny ever having received the message. Figure A.3 shows an example of this attack.

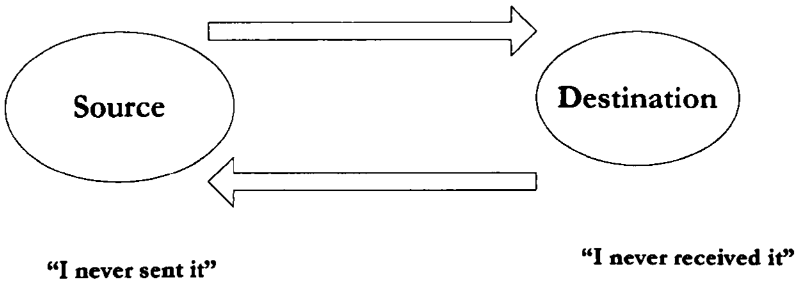


Figure A.3. An example of repudiation

The risks associated with 802.11 are the result of one or more of these attacks. The consequences of these attacks include, but are not limited to, loss of proprietary information, legal and recovery costs, tarnished image, and loss of network service [TISN]. Figure A.4 shows taxonomy of above attacks.

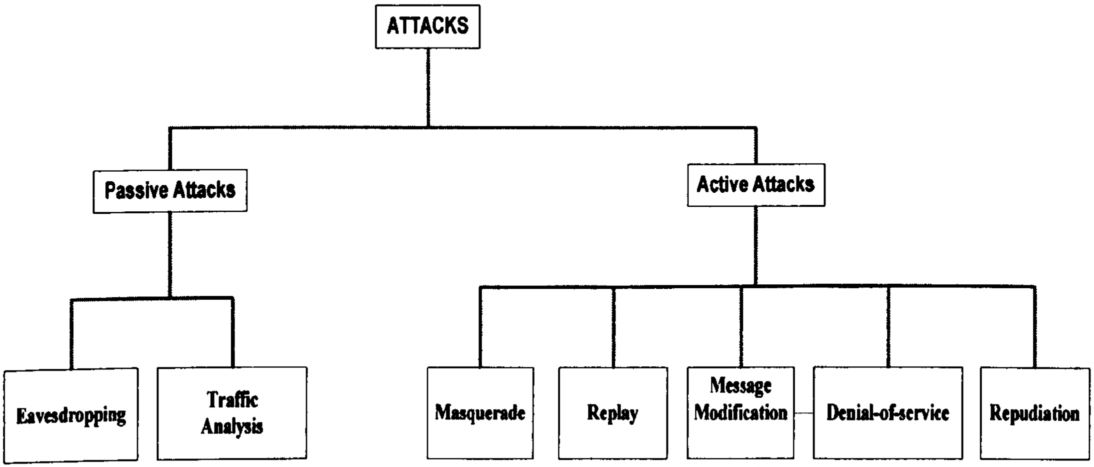


Figure A.4. Taxonomy of Wireless Security Attacks [TISN]

The remaining of this chapter will discuss the available IEEE security features and their drawback. Furthermore in order to show the inherit vulnerabilities of WLANs; there are some experimental approaches to present most popular passive and active attacks.

A.5 Attacks in WLAN

This section would present experiments and exercises on various methods of hacking and protecting a wireless network. It provides all the background information and examines the theory behind attacks on a wireless network. As previously mentioned, this section would discuss and cover two major types of attacks; passive and active, and would show the theory behind them. It would examine the systems in a widespread use of today's WLAN technology to counteract the methods of hackers. Throughout this section the techniques that are explained are in use by actual hackers, and although a talented hacker will have sophistication and complexity to their attacks to achieve maximum impact whilst trying to keep anonymous, the basic underlining theory behind the attack is always the same. Also throughout this section the drivers and programs that are talked about are mainly for the Linux Operating System (OS), unless otherwise stated. This is because most hacking programs available require Linux as the OS as it is open source and this has always been the choice for hacker's, because this adds flexibility to change drivers and programs as necessary. This is something that would be most difficult under Windows, as it is closed source, meaning the source code is not available to the public. All programs and drivers mentioned here are licensed under the GNU GPL (General Public License). The GPL is basically permission to investigate the internal structure of programs, as well as modify the program, and redistribute it [Gowa2003]. A condition of modification and redistribution is access to the source code, which most programs provide. See Appendix B for Chipset and Driver Coverage.

A.5.1 Preparation for attack

What covered in appendix B were the chipsets that are used in wireless cards. The chipset used is important, because available programs are designed specifically for use with certain chipsets, with the Prism chipset being most dominant. Also of importance is the card's ability to support different functions, and these functions are mainly support for RFMON mode and the softAP feature. The card's ability to construct wireless network frames can be important for some special tasks, but software is available that can construct these frames and do not rely on any other special features of the card. Once the correct card has been obtained and has been tested against the required features, then all that is needed is the skill and freely available software to perform the attacks. For the attacks that are about to be covered, a complete walkthrough will be given from a theoretical viewpoint. The wireless network attacks to be covered will be breaking WEP encryption, different flavours of the DoS (Denial of Service) attack, IP spoofing (includes port scanning), MAC address spoofing, wireless traffic injection and the Man In The Middle (MITM) attack. Whilst some of these attacks are simply about information gathering of the target network (passive attacks), some of them are very destructive (active attacks).

A.5.1.2 Denial of Service

One of the most destructive attacks is the Denial of Service attack (DoS), and is especially easy on wireless networks. DoS is an attack that has the intention of making a network resource, including servers, ordinary hosts, or entire network connections unreachable. One method of accomplishing this is by flooding a host's resources so that it becomes too consumed with work and eventually crashes [Vlad2004]. There are three main types of DoS attacks; OSI layer 1, layer 2 and layer 3. Layer 1 attacks aim to fill up the channel's space where an AP's hotspot resides to cause that AP DoS to the network [Vlad2004]. An example of a program capable of this is FakeAP. FakeAP works by generating fake beacons for APs that will eventually cause disturbance to the networking channel.

Layer 2 DoS attacks can be sent by an external host to a wireless network, and as an example can forge De-authenticate packets from the wireless AP to any of its hosts, or to all of its hosts. If this kind of attack takes place constantly, the hosts involved will be disconnected from the AP for as long as the hacker is sending the packets, which effectively denies the host network access. An example of a program capable of this is Aireplay from the Aircrack tools. Using this form of the DoS attack means that the hacker doesn't need to be a part of the target network to cause the intended harm, but can also work from an internal machine on the target wireless network. Also as management frames are not encrypted on wireless networks, bypassing encryption is not necessary and so this attack can have its full effect, encryption or no encryption [Cobb2004].

The third form of the DoS attack is layer 3 based and is commonly referred to as the SYN flood. This attack is based on the idea of sending SYN requests to specific ports of a target system too quickly for the system to process, thereby filling its backlog [Cobb2004]. This in turn means that it is not free to process legitimate network connections, which eventually results is DoS to the target. Using this type of attack means that a layer 3 connection needs to be available, meaning that the hacker needs to be an internal computer-part of the target network. This attack is formed on the basis of the TCP three way handshakes, which is used in all communications using connection-oriented reliable TCP communication. The three way handshake is shown in Figure A.5.

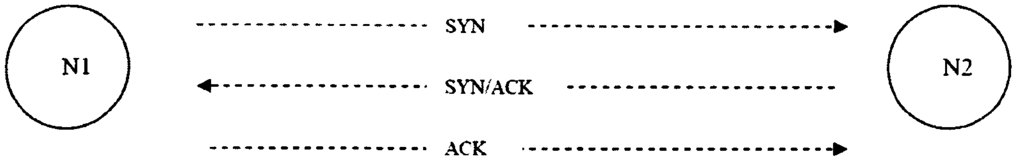


Figure A.5. Three way handshake

During the first step node 1 (N1) sends a SYN packet to node 2 (N2) to "synchronise" a connection on a certain port. If the port is open PC (B) will reply with a SYN/ACK packet to firstly acknowledge that it received N1's SYN packet, and also a synchronising of its own data. The final part is for N1 to send an acknowledgement that it received N2's synchronise packet. After these three steps the "3 Way Handshake" is complete, then communication can occur using the now "open" connection.

All the DoS attacks are similar, because they are only effective for as long as the hacker is flooding the DoS packets. The layer 3-based attack can be improved by sending a larger payload with the SYN packets, and also increase the window size to give the target extra work to do making it even busier with each packet sent, and this gives good results. The window size works by changing the amount of bytes transmitted in each connection that is agreed on by both participants. An example of a program that will perform this attack is Hping, which will be covered later. The standard form of "ping" built into Linux will also work, as it has the option to change the interval between packets, which is important when performing DoS.

A.5.1.3 IP Spoofing

IP spoofing itself is not a destructive attack but rather a way of masking your identity during communication. An example of where IP spoofing can be used is when performing system fingerprinting. This is the collection of information by probing the target system for open ports, services and so forth, and by masking the hacker's IP address for someone who actually exists on the target network, which shifts the blame onto them.

As mentioned, an example situation of using IP spoofing is port scanning. To achieve maximum stealth, port scanning is usually done using a legitimate network user as the victim, and so using their IP address in place of the hacker's. IP spoofing used to port scan is also a layer 3 scan, meaning that the hacker needs to be part of the target to do the scan. Figure A.6 shows the basic concept of IP spoofing put to use by port scanning.

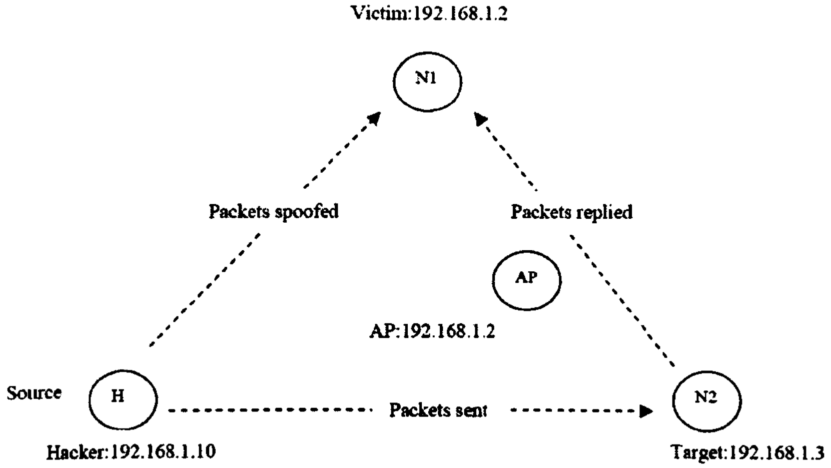


Figure A.6. Basic concept of IP spoofing

In the situation above the hacker has become part of the target network. The hacker is pretending to be the PC (Victim) for the purpose of fingerprinting the Server (Target). This means that any logs produced or any alarms raised will point to the PC user as the offender, when in fact they are not. Another point to note is that any replies sent by the server will go to the PC and not the hacker. This means that the spoofing process is actually replacing the source address in the packet with that of the Victim, and so when the server replies, it replies to the sender of that packet who is the hacker. It is important to realise that the spoofing of the IP address is done on the fly, and should not be done as part of the wireless card's configuration. The reason for this is that the spoofed IP address is part of the network and in use, and if the hacker changes their IP address in the configuration of the wireless card this will flag a duplicate IP address warning on the PC's console and this will draw attention to unusual activity. It is possible to perform a more destructive form of this attack, by performing a DoS attack on the PC and therefore taking it out of the network, and then the hacker could change their IP address in the card's configuration. This is unnecessary though, as it will draw attention to the PC's user and requires two wireless cards; one for the DoS attack, and the other for the direct port scanning. Another important point about this type of scan is the difficulty in knowing the information contained in the replies, because the hacker does not receive any replies. There is a way around this, and involves constant monitoring of the PC to detect changes in its sequence numbers.

Normally, a host will respond in different ways depending on the situation. If the host sent the SYN packet, as in the 3-way handshake then it will reply with a SYN/ACK packet as in the third step of the handshake. This only happens if the service or "port" requested is offered, but will reply with a RST/ACK if the service or port is not offered. If the host did not send the originating SYN packet, then it will send a RST packet to act as a "reset" as it did not initiate a connection. Taking these possibilities into consideration, any response from a host, for example a RST packet, will ultimately affect its sequence numbers from the usual pacing. Also take into consideration when performing this attack that the Victim does need to be relatively inactive, maybe during a quiet period when network usage is very low.

This is required because a difference in its sequence numbers needs to be noticeable, and so if it is busy this will be much more difficult to notice. This means that when monitoring the Victim PC, any changes in its usual sequence numbers will indicate that it is sending out RST packets to the Server, in return to services requested for by the hacker. For example, if the hacker has sent 7 SYN packets requesting service on port 21 (FTP), and at the same time whilst monitoring the Victim PC it notices changes indicating 7 packets sent out, then a conclusion can be made that service on port 21 is offered on the Server. All possible open ports can be discovered this way using this technique, and so acts as a good information gathering technique that can be used for maybe a more sophisticated attack. A program capable of this type of scan is Hping.

A.5.1.4 MAC Spoofing

This type of pretence again is not destructive by itself, but when combined with other techniques can be the basic of a more powerful attack. MAC spoofing, like IP spoofing is about pretending to be someone else usually for the purpose of getting special access to somewhere, maybe bypassing a router's "MAC list" security. MAC spoofing is most easily done using a Linux machine. Although every MAC address is "burnt" onto the card itself and is unique, techniques are being used now to overcome this in the software part of a wireless card's communication. By defining a software based MAC address, all communications from that wireless card will use that newly defined MAC address, over-riding the hardware defined MAC address. As mentioned, this type of pretence is mostly useful in overcoming "MAC list" security provided by routers and wireless APs. Once the changing of the MAC is done on the hacker's machine, their machine is "trusted" by the AP, and given access to the network, providing all other credentials are met including encryption (if any). Another powerful use of this technique is the Man In The Middle attack. By changing the MAC on the wireless card, being used as a softAP, all of its associated hosts will turn to the fake AP when their usual (legitimate) AP gets a DoS attack. To change the MAC of the wireless card under Linux, the ifconfig command can be used.

A.5.1.5 MAN In The Middle

The Man In The Middle (MITM) attack is one of the most destructive and powerful attacks to perform on a wireless network. This attack involves the primary goal of overtaking a legitimate AP and pretending to be it, thereby taking over its usual connected wireless hosts. Once this has been achieved, every packet that passes across the network will go through the fake AP first, and so will see all communication on the network in unencrypted-clear text. This attack does involve other attacks mentioned to this point including a DoS attack, MAC spoofing and breaking encryption if the legitimate AP has it turned on. This MITM attack can be looked at by looking at key stages. The first stage would be network discovery, to look for the AP that will be used and its credentials including channel assignment, encryption and so forth. The second stage would be to perform the DoS on the target AP to effectively disconnect its wireless link to the network. The third and final stage would be to ensure the connectivity of all connected hosts to the fake AP in control by the hacker. After these main stages are complete, the hacker will be able to eavesdrop on all network communication. An important point to note about this attack is that two wireless cards are needed [Vlad2004]. The first performs the DoS attack on the legitimate AP, while the other acts like the softAP and effectively eavesdrops on the network. Figure A.7 will be used for illustration purposes.

The first stage of this attack is use network discovery programs such as Kismet to find the target AP. Kismet is a good program to use, because a lot of credentials can be found out. When taking over an AP in a hostile way it is important to make sure that all of its features are copied. Some wireless cards can be very picky and if the smallest detail is left out of the fake AP then it may refuse to connect to it. The best approach to ensuring a successful attack is to copy the encryption, the AP's MAC address, the network name or "ssid", and a DHCP server needs to be running on the softAP in order for hosts to get their IP addresses.

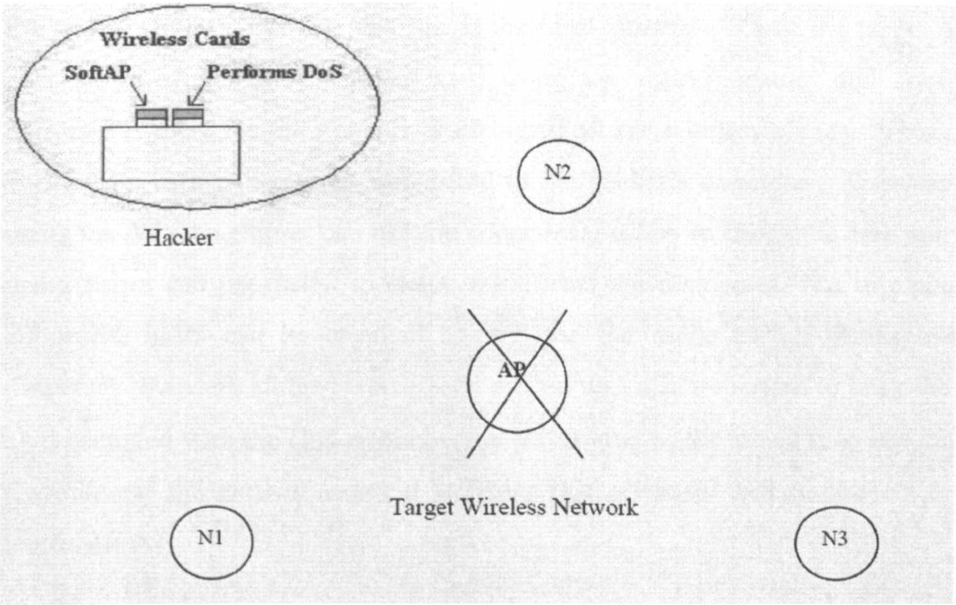


Figure A.7. Man in the Middle (MITM) attack

The information that can be found out from network discovery includes the MAC address of the AP and of the clients connected and the SSID (network name). Other information that can be found out includes the networking channel assignment, whether or not encryption is enabled and whether the network is infrastructure or *ad hoc*. It is worthwhile to note the MAC address of the AP as this will be needed, and of the clients as this can act like a checklist later when verifying connectivity between the hosts. To get the encryption key, it will have to be broken using the method described earlier. A simple DHCP server can be dhcpd on Linux systems. A common IP address of an AP is 192.168.1.1. The alternative is to break the encryption on the network, become a part of the network for enough time to eavesdrop on the communications and establish the IP address used. This can be done using a program like Ethereal. Once all the credentials have been found out, they can be used to configure the softAP. Wireless cards that offer the softAP feature include cards using the chipsets Prism, Hermes, Symbol and Atheros [Vlad2004]. These cards are available to purchase and include good support. The second stage of this attack involves performing DoS on the target AP using the second wireless card. A layer 2-based DoS can be used, as it can be from outside the target wireless network, and this is the ideal situation. Once the target AP has been hijacked, the first wireless card using the softAP feature and configured correctly is there for the immediate taking of all the wireless clients. This can be verified by listing the clients connected to the hacker's computer. Wireless cards using the Atheros chipset can use the wlanconfig utility to check the card and cards using Prism can use iwlist to check the clients are connected. At this point the connected hosts can be checked against the list made earlier during network discovery to ensure all hosts have been picked up. It is important to keep the other card occupied with the DoS continuously for as long as the attack is to last, because the moment the DoS is ended it is likely that all hosts will re-connect to their legitimate AP.

Once the attack has been successful, and all clients have connected to the hacker's machine, connectivity can be tested with ping. This ensures that all hosts have connected to the fake AP and that a communications link exists between the softAP and all the hosts. At this point the attack is a success, and now important information travelling across the network between clients can be leaked to the hacker. This means that the hacker can see and copy all packets in clear text including private and valuable information, otherwise known as eavesdropping. It is important to note during this attack that from the viewpoint of all the hosts, nothing has changed. They might get a "network connection established" like windows XP notifies in the taskbar, but unless they do some investigation they will be unaware of what has taken place. The only changes to a client's machine will be their IP address, and that should be it.

Appendix B. Experiments

B.1 Chipset and Driver Coverage

Given the theory behind any attack, it cannot go ahead with the most important component-the wireless card. Programs that are available for free download off the Internet use a few chipsets but one is very popular to hackers, because of its manufacturer's openness of its specification, and that is the Prism chipset. The Prism chipset was formally made by Intersil, who have always made information freely available from their website that describes the Prism chipset's specification, operation and structure. The Prism chipset is now maintained by Conexant. This has been very popular and has led to a whole list of programs that have the baseline requirement of using this chipset. The Prism chipset has evolved into a family of chipsets from the release of the original 802.11 standard until present. There have been many versions of the chipset, but of the most notable was Prism 1 with relation to the original 802.11 standard. With the release of 802.11b came Prism 2 and Prism 3, then Prism Indigo for 802.11a, Prism GT for 802.11b/g and Prism Duette for 802.11a/b. There are many wireless cards produced by many manufacturers in the form of PCMCIA adaptors, PCI cards and USB adaptors that use the Prism chipset [Vlad2004]. The Linux driver for Prism based cards (Prism 2, 2.5 and 3), named Host AP, can be found at http://hostap.epitest.fi.

There are many more chipsets that are seen as adequate for use in hacking. There are a few key points that make a chipset adequate for hacking. RFMON[8] mode is one of these. RFMON mode simply places the wireless card into "listen" mode so that any packet that crosses the wireless interface is read, and can be captured [Vlad2004]. This means that any packets sent on a single channel can be captured or if the channel is in roaming mode, any packets are "seen" by the wireless card.

[8] "Radio Frequency Monitoring, RFMON is a passive method of WLAN discovery. It is a sniffing mode which allows the card to report drivers from the 802.11 layer. A client with a wireless card that is configured in RFMON mode will be able to capture all RF signals on the channels to which it is configured to listen. RFMON is a receive-only mode. While in RFMON mode, wireless clients are unable to transmit any frames; their cards are only able to receive, and therefore capture traffic" http://www.webopedia.com/TERM/R/RFMON.html

Another important feature of any hacker-friendly wireless card is support for being an Access Point (software based), and also its ability to build wireless frames and transmit them [Vlad2004]. One of the most important chipset is the Cisco Aironet and chipsets made by Hermes, Symbol, and Atheros.

The Aironet chipset is a Cisco proprietary and used in products sold only by Cisco. One unique feature of Cisco Aironet cards that use the Aironet chipset is the ability to hop through all available channels without using a software-based channel hopper. The only drawback to this is that when using an Aironet card in RFMON mode it cannot be locked to one channel, because the firmware would override the software-based option and continue to hop through the channels. Aironet cards are also known for their excellent receiving sensitivity. Aironet cards are well supported across all common Windows and UNIX-based operating systems. The main disadvantage with an Aironet card is that it cannot be placed into software AP mode (softAP) mode. The reason for this is Cisco is the only people allowed to make the drivers for their wireless cards because their cards are proprietary, and so only Cisco drivers are available. However the Aironet cards are popular when it comes to site surveying, and multiple-channel traffic analysis due to their hardware-based channel-hopping feature [Vlad2004]. Cisco Aironet drivers for Linux can be found at http://www.cisco.com/pcgi-bin/tablebuild.pl/aironet-utils-linux.

The Hermes chipset is developed by Lucent. Cards using this chipset have been available for a long time like other makes, and are known for good receiving sensitivity and being user-friendly [Vlad2004]. These cards can be placed into monitor using a patch released by the Shmoo Group, and is available at http://airsnort.shmoo.com/orinocoinfo.html. This patch enables Hermes based cards to be put into RFMON mode. Another project that supports softAP mode for Hermes based cards is HermesAP, available at http://www.hunz.org/hermesap.html. This project will also support RFMON mode.

The Symbol chipset is based on the Prism 2 chipset but with their own MAC layer controller. The MAC layer controller is the part that communicates with the software drivers, and its firmware performs most of the basic 802.11 protocol functions including RFMON mode handling, frame insertion and softAP mode handling [Vlad2004]. Cards using the Symbol chipset can be used under Linux with the Orinoco driver, which is the same as the Hermes driver, HermesAP.

The Atheros chipset is another popular and commonly sought after chipset in wireless cards. The most common Linux driver used for Atheros based cards is the Madwifi project, available at http://madwifi.org. A new addition to this project is Madwifi-NG, which can create Virtual Access-Points (VAP) [Madw]. These VAP are virtual devices that are fully functional interfaces, and can be used in infrastructure mode, RFMON* mode or AP mode, which creates a softAP. Both the old Madwifi project and the NG project are able to put the wireless card into RFMON mode.

There are windows drivers that support RFMON mode for chipsets based on Atheros, Agere, Broadcom and Realtek. These drivers can be used with LAN analysers, for example WildPacket's Airopeek. These drivers are available at WildPacket's site, which is available at http://www.wildpackets.com/support/downloads/drivers.

There are other chipsets that are used in hacking, but the ones covered above are the most commonly sought after for use with Linux. Other chipsets include Broadcom, Intel, Atmel and Realtek [Vlad2004].

For all the chipsets that do not have open-source drivers for Linux, there are alternatives. These alternatives are the NdisWrapper driver from SourceForge, and Driver Loader from Linuxant. NdisWrapper is a "wrapper" that uses the NDIS API to link the wireless card to the Linux kernel. NDIS is short for Network Driver Interface Specification and its main purpose is to define a standard API (Application Programmers Interface) for Network Interface Cards (NICs) [Ndis0000]. This means that NDIS acts as an interface, specifically for the layer 3 network protocol drivers, and layer 2 hardware level MAC drivers [Ndis0000].

NDIS is essentially a library of functions that hides the underlining complexities of the NIC hardware. NdisWrapper acts like a "wrapper" to wrap around these NDIS functions, primarily used for Windows on Intel-based computers, and integrate them into the Linux system kernel. What this means in terms of the end-user, is that they can take the Windows based driver from their installation CD or download it from the appropriate manufacturers website and use it with the NdisWrapper drivers under Linux. What this also means is that practically any wireless card designed for use under Windows can be used under Linux in this way [Vlad2004]. A list is maintained at SourceForge, and is available at http://ndiswrapper.sourceforge.net/mediawiki/index.php/List. This list keeps track of a number of manufacturers whose cards are known to work along with any special instructions and Windows driver locations. Another very important feature of this list is that for the majority of wireless cards the chipset information is shown. This is very important as discussed above, when searching for the right card to use for hacking. The only downside of this is that manufacturers change the versions of their chipsets and it is not part of a product's advertising to show the change in version numbers. This means that it is a trial and error situation when purchasing wireless cards for specific chipsets. The main page for NdisWrapper is found at http://ndiswrapper.sourceforge.net/support.html. An alternative to NdisWrapper is Driver Loader, sold by Linuxant. This driver is an alternative but is only on a trial basis and there is a cost for this driver after the trial period. This driver functions the same as NdisWrapper and uses the same Windows based driver for the wireless card. The only difference between the two is NdisWrapper is free under the GPL and Driverloader is commercial, but with a free trial. Driverloader is available from http://www.linuxant.com/driverloader.

B.2 Installation and Experiments for Kismet

Kismet was obtained from *http://www.kismetwireless.net/code/kismet-2006-04-R1.tar.gz* The "kismet-2005-08-R1" has been used in our test bed. The installation progress was as follows.

```
[root@localhost project] tar -xzvf kismet-2005-08-R1.tar.tar
[root@localhost project] cd kismet-2005-08-R1
[root@localhost project] ./configure
[root@localhost project] make
[root@localhost project] make install
```

Once installed, Kismet needs to be configured. The command in the instructions above "./configurer" is used by a lot of open-source vendors now to creating an automatic checklist of what is on the target system, as well as to make sure all required dependencies are met. These Dependencies are often the common programs and utilities installed with Linux. To configure Kismet, two of the files in "/usr/local/etc" are used. The files are "kismet.conf" which controls the main functionality of Kismet and "kismet_ui.conf", which controls the appearance of the GUI that Kismet uses. We changed the "kismet.conf" file to become:

```
Suiduser=guest
Source=madwifing_g,wifi0,MadwifiNG
Writeinterval=20
Logtemplate=%n-%d-%I-%l
```

Let's look at the definition of parameters. The "Suiduser" parameter is used to set Kismet up for the user who will run the program. In this case the user "guest" will run Kismet, and this is considered the proper secure way of running Kismet, as it will not process harmful whilst using root access, as root is not the user to run Kismet. The "source" parameter is used to configure Kismet to use a certain manufacturer of chipset, as it places the card into RFMON mode, which is a feature controlled by the chipset. We changed the source to use the MadWifi-NG driver which is used for Atheros based chipset cards. Over the time of research implementation these card had to change. These different assignments are shown below:

- Atheros Chipset: madwifing_g,wifi0,MadwifiNG
- Prism Chipset: hostap, wlan0, Prism
- Realtek chipset: rt2500, rausb0, Realtek

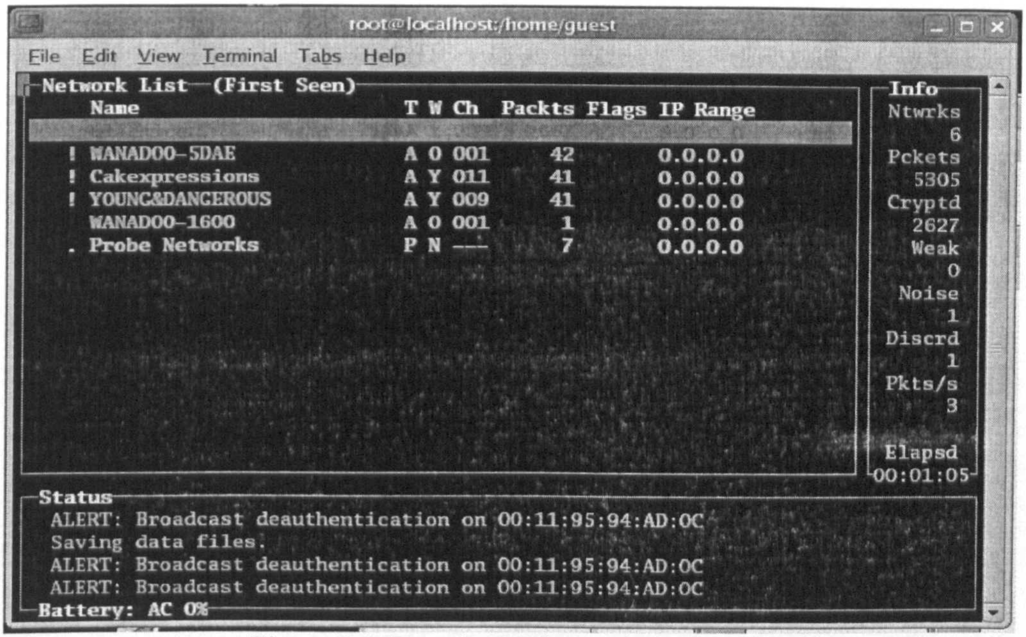
As we used each card for the situation at hand, we changed the parameter manually to use the wireless card we needed. The first parameter, for example "hostap" tells Kismet which chipset the wireless card is using. This is found in the Kismet documentation at *http://www.kismetwireless.net/documentation.shtml.*There it lists the different assignments for different chipsets. The second parameter, for example "wlan0" tells Kismet which interface the wireless card is using. These are simply the interface names used by Linux, and are found out using the "iwconfig" command. In the above assignments, the Atheros chipset is a special case where the interface needs to be the "master" device; the "wifi0" device as opposed to the "ath0" interface. This is only true for the Madwifi-NG driver as stated in the Kismet documentation. The last parameter is user-defined, and is just used to identify what source Kismet is using, for example "MadwifiNG" for the Madwifi-NG driver.

The second part to Kismet's configuration is to configure how the GUI is presented in the file "kismet_ui.conf". We only wanted to change one parameter and that was "showintro" to "false" to skip the introduction every time Kismet loads, which is not important to the functionality of Kismet.

B.2.1 Prevention System Experiments for Kismet

The first program that we used was Kismet. Following is the command and screen-shot I (Figure B.1) used for a Broadcast Disassociation attack.

```
[root@localhost project] aireplay –0 0 –a 00:11:95:94:AD:0C rausb0
```

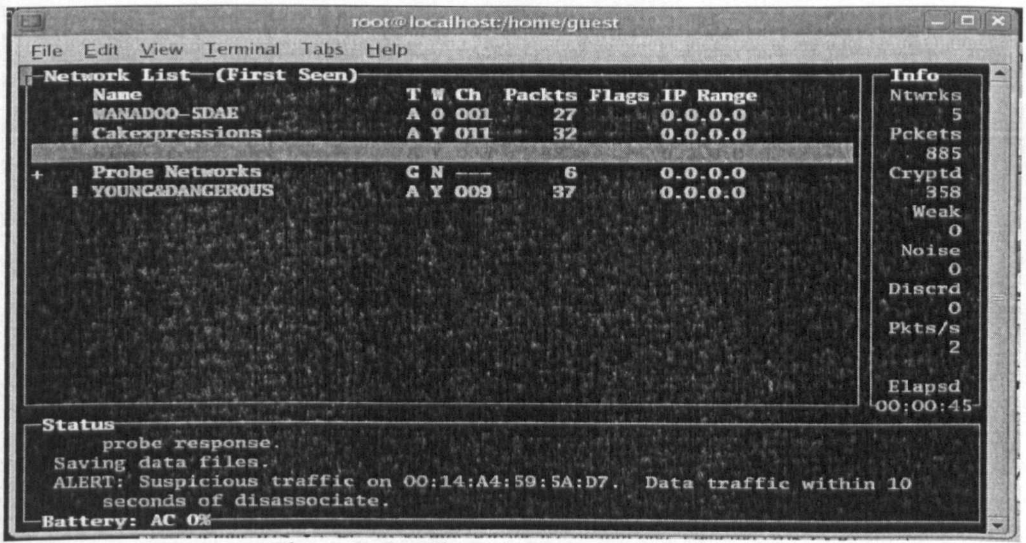


```
Network List—(First Seen)
     Name                       T W Ch  Packts Flags IP Range

   ! WANADOO-5DAE               A O 001     42        0.0.0.0
   ! Cakexpressions             A Y 011     41        0.0.0.0
   ! YOUNG&DANGEROUS            A Y 009     41        0.0.0.0
     WANADOO-1600               A O 001      1        0.0.0.0
   . Probe Networks             P N ---      7        0.0.0.0

Info
Ntwrks
     6
Pckets
  5305
Cryptd
  2627
  Weak
     0
 Noise
     1
Discrd
     1
Pkts/s
     3

Elapsd
00:01:05

Status
 ALERT: Broadcast deauthentication on 00:11:95:94:AD:0C
 Saving data files.
 ALERT: Broadcast deauthentication on 00:11:95:94:AD:0C
 ALERT: Broadcast deauthentication on 00:11:95:94:AD:0C
Battery: AC 0%
```

Figure B.1. Broadcast Disassociation attack

The next command and output (Figure B.2) shows a client de-authenticate attack.

```
[root@localhost project]aireplay –0 0 –a 00:11:95:94:AD:0C –c00:14:A4:59:5A:D7 rausb0
```



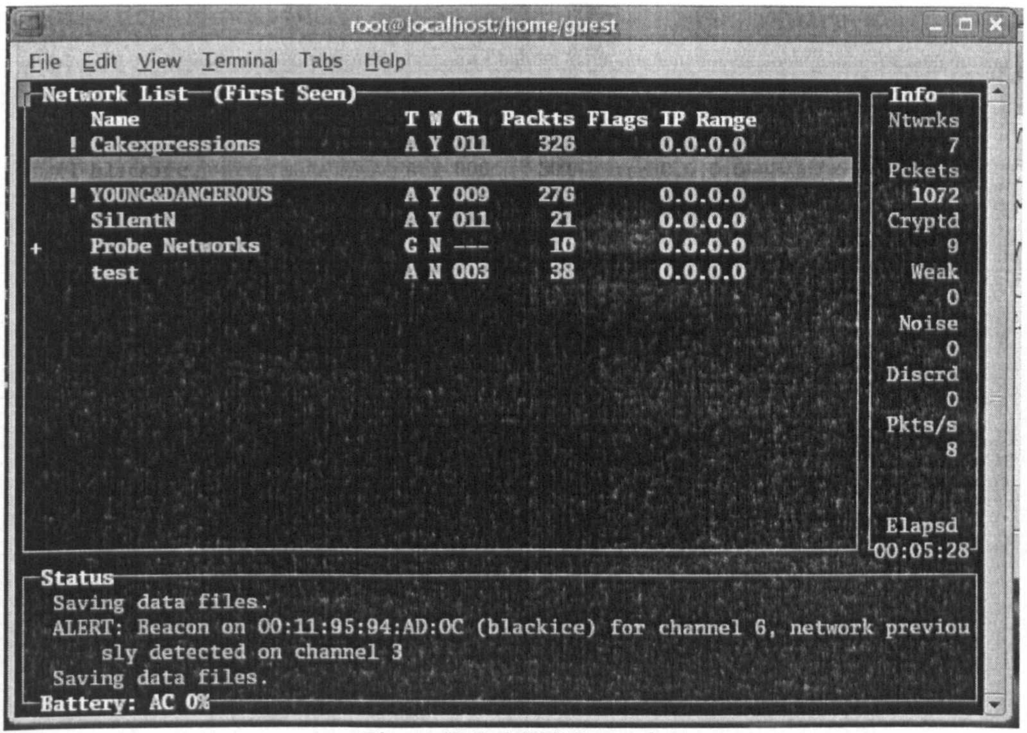
```
Network List—(First Seen)
     Name                       T W Ch  Packts Flags IP Range
   . WANADOO-5DAE               A O 001     27        0.0.0.0
   ! Cakexpressions             A Y 011     32        0.0.0.0

+    Probe Networks             G N ---      6        0.0.0.0
   ! YOUNG&DANGEROUS            A Y 009     37        0.0.0.0

Info
Ntwrks
     5
Pckets
   885
Cryptd
   358
  Weak
     0
 Noise
     0
Discrd
     0
Pkts/s
     2

Elapsd
00:00:45

Status
     probe response.
 Saving data files.
 ALERT: Suspicious traffic on 00:14:A4:59:5A:D7.  Data traffic within 10
     seconds of disassociate.
Battery: AC 0%
```

Figure B.2. Client de-authenticate attack

The next attack (Figure B.3) shows the MITM attack.

[root@localhost project] aireplay –0 0 –a 00:11:95:94:AD:0C –c

00:14:A4:59:5A:D7 rausb0



```
Network List—(First Seen)                                              Info
      Name                       T W Ch  Packts Flags IP Range         Ntwrks
    ! Cakexpressions             A Y 011    326         0.0.0.0             7
      blackice [illegible]                                             Pckets
    ! YOUNG&DANGEROUS            A Y 009    276         0.0.0.0          1072
      SilentN                    A Y 011     21         0.0.0.0        Cryptd
  +   Probe Networks             G N ---     10         0.0.0.0             9
      test                       A N 003     38         0.0.0.0          Weak
                                                                            0
                                                                        Noise
                                                                            0
                                                                       Discrd
                                                                            0
                                                                       Pkts/s
                                                                            8

                                                                       Elapsd
                                                                     00:05:28
Status
  Saving data files.
  ALERT: Beacon on 00:11:95:94:AD:0C (blackice) for channel 6, network previou
      sly detected on channel 3
  Saving data files.
Battery: AC 0%
```

Figure B.3. MITM attack

B.3. Prevention System Experiments for BRO IDS

To install Bro-IDS we typed the following commands.

[root@localhost project] tar –xzvf bro-1.X-current.tar.gz

[root@localhost project] cd bro-1.1

[root@localhost project] ./configure

[root@localhost project] make

[root@localhost project] make install-brolite

```
-----------------------------------------------------------------------------------------------
```

This installs BRO-IDS on Linux. The configurations provided by BRO should be adequate.

To test BRO we used the following commands.

```
-----------------------------------------------------------------------------------------------
[SYN flood]
[root@localhost project] hping 192.168.1.4 –S –p 21 -i u1 –w 30772
[Spoofed SYN flood]
[root@localhost project] hping 192.168.1.4 –S –p 23 –i u1 –w 30772 –a
192.168.1.1
[Weird activity]
[root@localhost project] hping 192.168.1.4 –SA –p 80 –i u1 –w 30772
-----------------------------------------------------------------------------------------------
```

BRO creates a report type log of all necessary details including attacks connections and so forth. This is an example of what it looks like.

```
Site Report for SECURE, from 2006/08/06 00:00:30 to 2006/08/07 00:00:30
generated on Mon Aug  7 16:25:24 2006

========================
Summary
========================
 Incident Count: 2
 Scanning Hosts
  Successful       1
  Unsuccessful      0
======================
Incident Details
======================
         # legend for connection type
#
         ------------------------------
         C Connection Status
           # number corresponds to alarm
triggered by the connection
           *       successful       connection,
otherwise unsuccessful.
         I Initiatator of Connection
           > connection  initiated  by  remote
host
           < connection  initiated  by  local
host
---------------------------------------------
-----
Incident     SECURE-000015
```

```
-----------------------------------------------
----
Alarm: SynFloodStart
    start    of    syn-flood    against
192.168.1.6; sampling packets now
    08/06                       15:28:40
192.168.1.5 ->
Alarm: SynFloodStatus
    syn-flood    against    192.168.1.6;
estimated 0 connections in last
     1.0 min 82.0 usecs
    08/06                       15:28:58
192.168.1.5 ->
Alarm: SynFloodStatus
    syn-flood    against    192.168.1.6;
estimated 20700 connections in
     last 1.0 min 195.0 usecs
    08/06                       15:29:58
192.168.1.5 ->
Alarm: SynFloodStatus
    syn-flood    against    192.168.1.6;
estimated 27800 connections in
     last 1.0 min 313.0 usecs
    08/06                       15:30:58
192.168.1.5 ->
Alarm: SynFloodEnd
    end of syn-flood against 192.168.1.6;
stopping sampling
    08/06                       15:30:58
192.168.1.5 ->
Alarm: SynFloodStart
    start    of    syn-flood    against
192.168.1.6; sampling packets now
    08/06                       16:01:50
192.168.1.5 ->
Alarm: SynFloodStatus
    syn-flood    against    192.168.1.6;
estimated 0 connections in last
     1.0 min 324.0 usecs
    08/06                       16:01:58
192.168.1.5 ->
Alarm: SynFloodStatus
    syn-flood    against    192.168.1.6;
estimated 17700 connections in
     last 1.0 min 837.0 usecs
    08/06                       16:02:58
192.168.1.5 ->
Alarm: SynFloodEnd
    end of syn-flood against 192.168.1.6;
stopping sampling
    08/06                       16:02:58
192.168.1.5 ->
Connections (only first 30 after alarm
are listed)
----------
            time    byte   remote    local
byte
 date  time  duration transfer port C  I
port transfer  protocol
----- -------- -------- -------- ------ ------ ----
-
  No connection data available
---------------------------------------------
Incident     SECURE-000016
---------------------------------------------
Alarm: SynFloodStart
    start    of    syn-flood    against
192.168.1.6; sampling packets now
    08/06                       15:37:30
192.168.1.2 ->
Alarm: SynFloodStatus
    syn-flood    against    192.168.1.6;
estimated 28000 connections in
     last 1.0 min 456.0 usecs
    08/06                       15:37:58
192.168.1.2 ->
Alarm: SynFloodStatus
    syn-flood    against    192.168.1.6;
estimated 27000 connections in
     last 1.0 min 144.0 usecs
    08/06                       15:38:58
192.168.1.2 ->
Alarm: SynFloodEnd
    end of syn-flood against 192.168.1.6;
stopping sampling
    08/06                       15:39:58
192.168.1.2 ->
Alarm: SynFloodStart
    start    of    syn-flood    against
192.168.1.6; sampling packets now
```

```
    08/06                    16:05:29
192.168.1.2 ->
Alarm: SynFloodStatus
    syn-flood    against    192.168.1.6;
estimated 0 connections in last
    1.0 min 910.0 usecs
    08/06                    16:05:58
192.168.1.2 ->
Alarm: SynFloodStatus
    syn-flood    against    192.168.1.6;
estimated 18000 connections in
    last 1.0 min 877.0 usecs
    08/06                    16:06:58
192.168.1.2 ->
Alarm: SynFloodEnd
    end of syn-flood against 192.168.1.6;
stopping sampling
    08/06                    16:06:58
192.168.1.2 ->
Connections (only first 30 after alarm
are listed)
----------
          time    byte   remote     local
byte
 date  time  duration transfer port C  I
port transfer  protocol
----- -------- -------- -------- ------ ------ ----
- --
  No connection data available
---------------------------
=====================
Signature Distributions
========================
 No data to report
=====================
Scans
========================
 Host:                    192.168.1.5
192.168.1.5
 Reason: 192.168.1.5 has scanned 50
ports of 192.168.1.6
=====================
Connection Log Summary
========================
Site-wide connection statistics
  Successful:  89
```

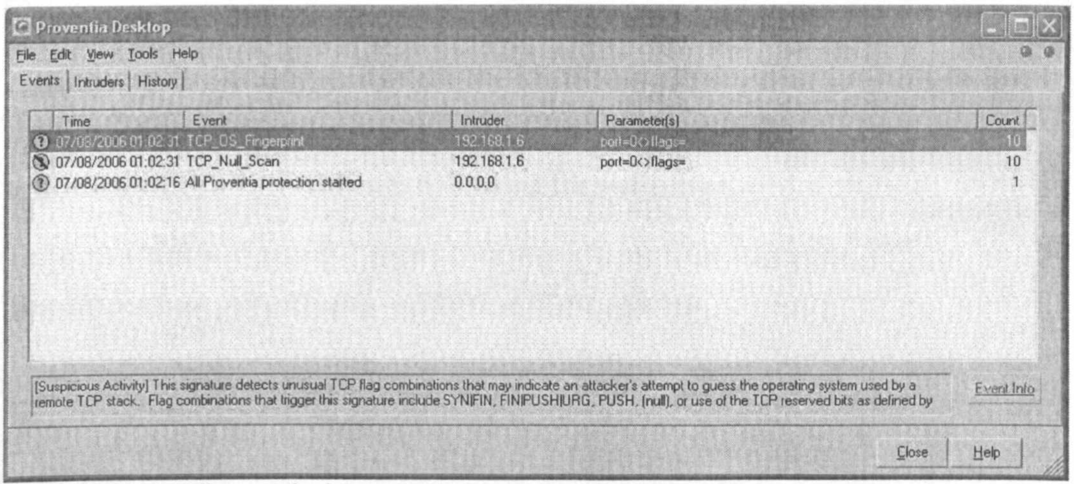
```
  Unsuccessful: 91669
  Ratio: 1:1029.
Top 20 Sources
          Host                    IP
Bytes  Conn. Count
------------------------------ --------------
            192.168.1.6    192.168.1.6
86179       83
            192.168.1.5    192.168.1.5
216         6
Top 20 Destinations
          Host                    IP
Bytes  Conn. Count
------------------------------ --------------
xmlrpc.rhn.redhat.com 209.132.177.100
194 K       78
            192.168.1.6    192.168.1.6
868         6
           194.72.108.2 194.72.108.2
23437       2
        web-cms-02.unl.ac.uk
163.167.82.61    7997        2
      ukadvip1.doubleclick.net
209.62.178.51    150         1
Top 20 Local Email Senders
 No data to report for this section
Top 20 Services
 Service        Conn. Count  % of Total
Bytes In  Bytes Out
----------- ----------- ---------- ---------
https              78     87.64     82986
194 K
 ftp-data           6      6.74       216
868
 http               5      5.62      3193
31584
=====================
Byte Transfer Pairs
========================
Hot Report - Top 20
                                  Local
Remote    Conn.
  Local Host              Remote Host
Bytes    Bytes    Count
-------------------- --------------------
```

```
xmlrpc.rhn.redhat.com        192.168.1.6        web-cms-02.unl.ac.uk        192.168.1.6
199091      82693  371                          7995        1130  4
194.72.108.2                 192.168.1.6        ...ip1.doubleclick.net      192.168.1.6
23435       1567  4                             149         491  2
```

B.4 Prevention System Experiments for Proventia

The first attack (Figure B.4) shows OS fingerprinting.

[root@localhost project] hping 192.168.1.4



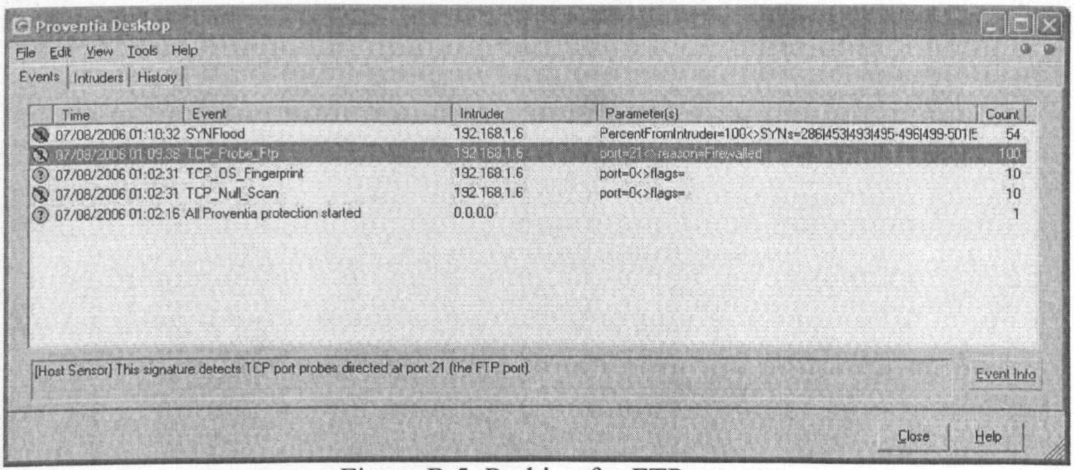
Time	Event	Intruder	Parameter(s)	Count
07/08/2006 01:02:31	TCP_OS_Fingerprint	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:31	TCP_Null_Scan	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:16	All Proventia protection started	0.0.0.0		1

[Suspicious Activity] This signature detects unusual TCP flag combinations that may indicate an attacker's attempt to guess the operating system used by a remote TCP stack. Flag combinations that trigger this signature include SYN|FIN, FIN|PUSH|URG, PUSH, (null), or use of the TCP reserved bits as defined by

Figure B.4. OS fingerprinting

The next attack (Figure B.5) shows Probing for FTP.

[root@localhost project] hping 192.168.1.4 –S –p 21



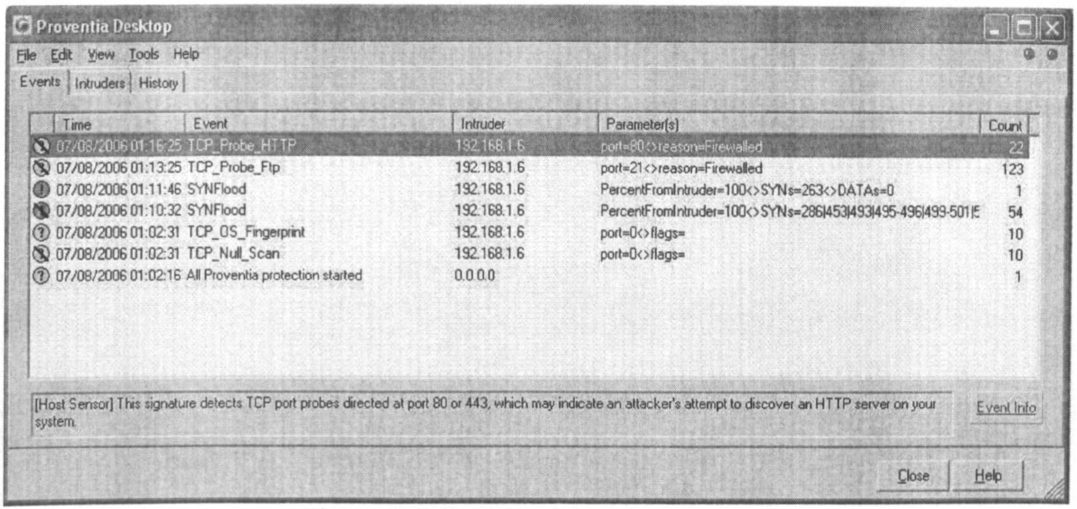
Time	Event	Intruder	Parameter(s)	Count
07/08/2006 01:10:32	SYNFlood	192.168.1.6	PercentFromIntruder=100<>SYNs=286|453|493|495-496|499-501|5	54
07/08/2006 01:09:38	TCP_Probe_Ftp	192.168.1.6	port=21<>reason=Firewalled	100
07/08/2006 01:02:31	TCP_OS_Fingerprint	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:31	TCP_Null_Scan	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:16	All Proventia protection started	0.0.0.0		1

[Host Sensor] This signature detects TCP port probes directed at port 21 (the FTP port).

Figure B.5. Probing for FTP

The next attack in Figure B.6 shows probing for web server

```
[root@localhost project] hping 192.168.1.4 –S –p 80
```



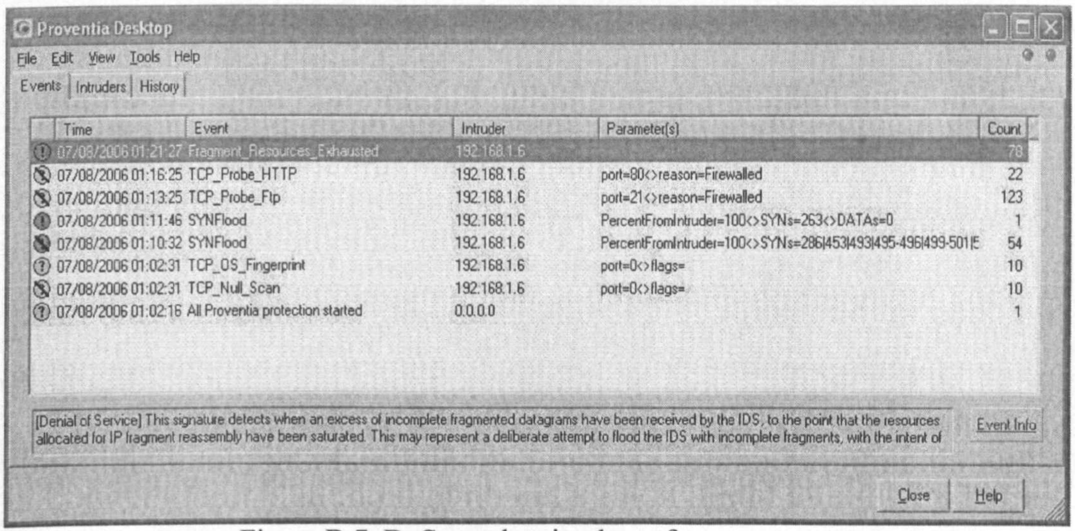
Time	Event	Intruder	Parameter(s)	Count
07/08/2006 01:16:25	TCP_Probe_HTTP	192.168.1.6	port=80<>reason=Firewalled	22
07/08/2006 01:13:25	TCP_Probe_Ftp	192.168.1.6	port=21<>reason=Firewalled	123
07/08/2006 01:11:46	SYNFlood	192.168.1.6	PercentFromIntruder=100<>SYNs=263<>DATAs=0	1
07/08/2006 01:10:32	SYNFlood	192.168.1.6	PercentFromIntruder=100<>SYNs=286|453|493|495-496|499-501|E	54
07/08/2006 01:02:31	TCP_OS_Fingerprint	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:31	TCP_Null_Scan	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:16	All Proventia protection started	0.0.0.0		1

[Host Sensor] This signature detects TCP port probes directed at port 80 or 443, which may indicate an attacker's attempt to discover an HTTP server on your system.

Figure B.6. Probing for web server

The next attack in Figure B.7 shows DoS attack using large fragments.

```
[root@localhost project] ping –s 65507 –I 0 192.168.1.4
```



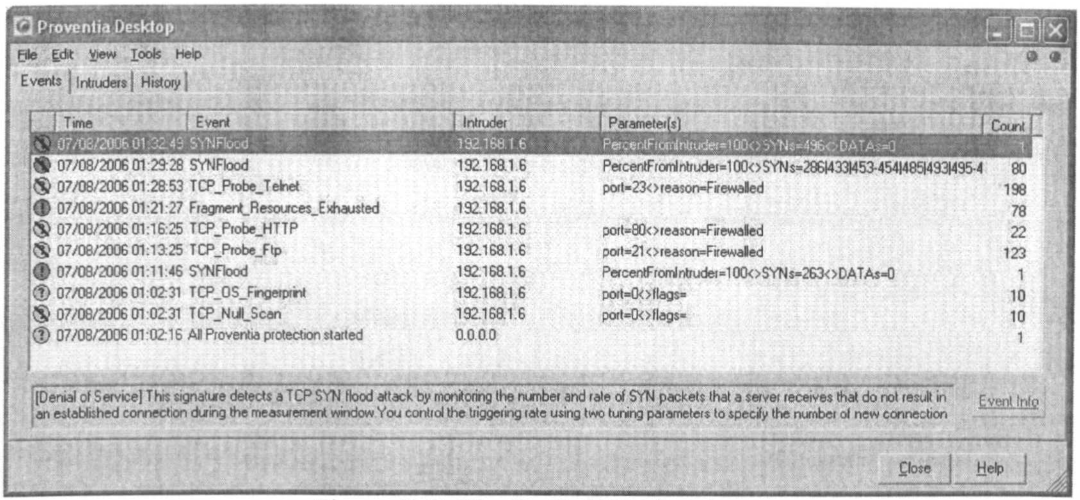
Time	Event	Intruder	Parameter(s)	Count
07/08/2006 01:21:27	Fragment_Resources_Exhausted	192.168.1.6		78
07/08/2006 01:16:25	TCP_Probe_HTTP	192.168.1.6	port=80<>reason=Firewalled	22
07/08/2006 01:13:25	TCP_Probe_Ftp	192.168.1.6	port=21<>reason=Firewalled	123
07/08/2006 01:11:46	SYNFlood	192.168.1.6	PercentFromIntruder=100<>SYNs=263<>DATAs=0	1
07/08/2006 01:10:32	SYNFlood	192.168.1.6	PercentFromIntruder=100<>SYNs=286|453|493|495-496|499-501|E	54
07/08/2006 01:02:31	TCP_OS_Fingerprint	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:31	TCP_Null_Scan	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:16	All Proventia protection started	0.0.0.0		1

[Denial of Service] This signature detects when an excess of incomplete fragmented datagrams have been received by the IDS, to the point that the resources allocated for IP fragment reassembly have been saturated. This may represent a deliberate attempt to flood the IDS with incomplete fragments, with the intent of

Figure B.7. DoS attack using large fragments

The next attack in Figure B.8 shows the DoS; SYN flood.

```
[root@localhost project] hping 192.168.1.4 –S –p +1 -i u1 –w 30772
```



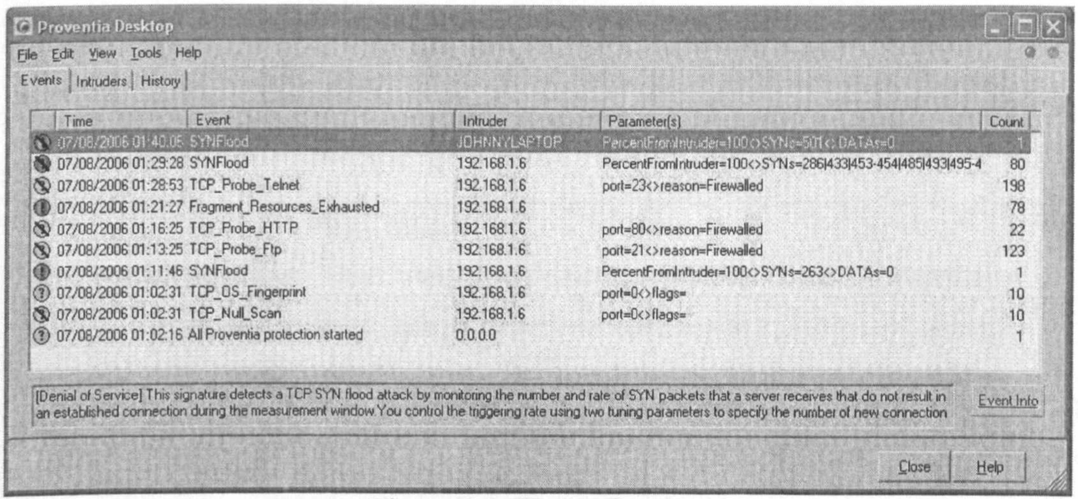
Time	Event	Intruder	Parameter(s)	Count
07/08/2006 01:32:49	SYNFlood	192.168.1.6	PercentFromIntruder=100<>SYNs=496<>DATAs=0	1
07/08/2006 01:29:28	SYNFlood	192.168.1.6	PercentFromIntruder=100<>SYNs=286|433|453-454|485|493|495-4	80
07/08/2006 01:28:53	TCP_Probe_Telnet	192.168.1.6	port=23<>reason=Firewalled	198
07/08/2006 01:21:27	Fragment_Resources_Exhausted	192.168.1.6		78
07/08/2006 01:16:25	TCP_Probe_HTTP	192.168.1.6	port=80<>reason=Firewalled	22
07/08/2006 01:13:25	TCP_Probe_Ftp	192.168.1.6	port=21<>reason=Firewalled	123
07/08/2006 01:11:46	SYNFlood	192.168.1.6	PercentFromIntruder=100<>SYNs=263<>DATAs=0	1
07/08/2006 01:02:31	TCP_OS_Fingerprint	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:31	TCP_Null_Scan	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:16	All Proventia protection started	0.0.0.0		1

[Denial of Service] This signature detects a TCP SYN flood attack by monitoring the number and rate of SYN packets that a server receives that do not result in an established connection during the measurement window.You control the triggering rate using two tuning parameters to specify the number of new connection

Figure B.8. The DoS; SYN flood

The next attack in Figure B.9 shows IP spoofing.

[root@localhost project] hping 192.168.1.4 –S –p 23 –i u1 –w 30772 –a 192.168.1.2



Time	Event	Intruder	Parameter(s)	Count
07/08/2006 01:40:05	SYNFlood	JOHNNYLAPTOP	PercentFromIntruder=100<>SYNs=501<>DATAs=0	1
07/08/2006 01:29:28	SYNFlood	192.168.1.6	PercentFromIntruder=100<>SYNs=286|433|453-454|485|493|495-4	80
07/08/2006 01:28:53	TCP_Probe_Telnet	192.168.1.6	port=23<>reason=Firewalled	198
07/08/2006 01:21:27	Fragment_Resources_Exhausted	192.168.1.6		78
07/08/2006 01:16:25	TCP_Probe_HTTP	192.168.1.6	port=80<>reason=Firewalled	22
07/08/2006 01:13:25	TCP_Probe_Ftp	192.168.1.6	port=21<>reason=Firewalled	123
07/08/2006 01:11:46	SYNFlood	192.168.1.6	PercentFromIntruder=100<>SYNs=263<>DATAs=0	1
07/08/2006 01:02:31	TCP_OS_Fingerprint	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:31	TCP_Null_Scan	192.168.1.6	port=0<>flags=	10
07/08/2006 01:02:16	All Proventia protection started	0.0.0.0		1

[Denial of Service] This signature detects a TCP SYN flood attack by monitoring the number and rate of SYN packets that a server receives that do not result in an established connection during the measurement window.You control the triggering rate using two tuning parameters to specify the number of new connection

Figure B.9. IP spoofing

Appendix C: Software Documentation and Components

C.1 Snort

The configuration of Snort for the experiment purposes required a good knowledge of its commands; therefore the following material has been gathered from Snort website below.

http://snort-wireless.org/docs/usersguide/

C.1.1 Snort Rules and Configuration Commands

MAC Addresses: Source and Destination MAC addresses may be specified in much the same way that IP addresses are specified in Snort rules. A single MAC address may be specified by a colon separated list of octets, or a list separated by commas and enclosed in braces. In addition the '!' character may prefix any MAC address to perform a logical NOT operation.

Example:

```
#                        Single                        MAC                        Address
00:DE:AD:BE:EF:00
#                         MAC                        Address                        List
[00:DE:AD:BE:EF:00, 00:DE:AD:C0:DE:00, ....]
```

The Direction Operator: Snort contains two operators for specifying the direction of the traffic that a rule applies to. The unidirectional operator -> translates to "source to destination." The bidirectional operator <> translates to "source to destination or destination to source."

Rule Options: Snort-Wireless has many 802.11 specific rule options available when creating rules that use the "wifi" protocol. Additionally many of Snort's standard rule options are available to use for "wifi" protocol rules.

C.1.2 Snort-Wireless Specific Keywords

Snort commands are shown as follow.

frame_control: tests the entire frame control field

type: tests the 802.11 frame's type

styp: tests the 802.11 frame's subtype

from_d: tests the from distribution system frame control flag

to_ds: tests the to distribution system frame control flag

more: frags :tests the more fragments frame control flag

retry: tests the retry frame control flag

pwr_mgmt: tests the power management frame control flag

more_data: tests the more data frame control flag

wep: tests the wep frame control flag

order: tests the order frame control flag

duration_id: tests the frame's duration/id field

bssid: tests the frame's BSSID

seqnum: tests the frame's sequence number

fragnum: tests the frame's fragment number

addr4: tests the frame's 4th address field

ssid: tests the frame's SSID

C.1.2.1 Frame control

The frame_control option allows checking whether or not the entire frame control field contains a specific value. Values may be specified in either hexadecimal or decimal notation. An 802.11 frame's control field is 16 bits wide, thus the minimum value allowed is 0 and the maximum is 65535. In addition, '!' may proceed the argument to specify a logical NOT operation on the comparison.

Format: frame_control:[!]<number>;

C.1.2.2 Type

The type option allows you to specify specific 802.11 frame types that a rule applies to. '!' may proceed the argument to specify a logical NOT operation on the comparison.

Format: **type:[!]<wifi type>;**

Valid Types:

STYPE_MANAGMENT
STYPE_CONTROL
STYPE_DATA

C.1.2.3 Stype

The stype option allows you to specify specific 802.11 frame subtypes that a given rule applies to. When using this rule option the frame's type is implicitly checked. In addition, '!' may proceed the argument to specify a logical NOT operation on the comparison.

Format: **stype:[!]<wifi subtype>;**

- **Valid Management Frame Subtypes**

  STYPE_ASSOCREQ

  STYPE_ASSOCRESP

  STYPE_REASSOC_REQ

  STYPE_REASSOC_RESP

  STYPE_PROBEREQ

  STYPE_PROBERESP

  STYPE_BEACON

  STYPE_ATIM

  STYPE_DISASSOC

STYPE_AUTH

STYPE_DEAUTH

- **Valid Control Frame Subtypes**

STYPE_PS

STYPE_RTS

STYPE_CTS

STYPE_ACK

STYPE_CFEND

STYPE_CFEND_CFACK

- **Valid Data Frame Subtypes**

STYPE_DATA

STYPE_CFACK

STYPE_CFPOLL

STYPE_CFACK_CFPOLL

STYPE_NULL

STYPE_CFACK_NULL

STYPE_CFPOLL_NULL

STYPE_CFACK_CFPOLL_NULL

C.1.2.4 From_ds

The from_ds option allows you to specify whether the rule applies to 802.11 frames leaving the distribution system (e.g., the frame is travelling from access point to stations). Valid arguments are ON, OFF, TRUE, or FALSE. For consistency and silliness, '!' may proceed the argument to perform a logical NOT operation on the comparison.

Format: **from_ds:[!] TRUE | FALSE | ON | OFF;**

C.1.2.5 To_ds

The to_ds option allows you to specify whether the rule applies to 802.11 frames entering the distribution system (e.g., the frame is travelling from a station to an access point). Valid arguments are ON, OFF, TRUE, or FALSE. For consistency and silliness, '!' may proceed the argument to perform a logical NOT operation on the comparison.

Format: **to_ds:[!] TRUE | FALSE | ON | OFF;**

C.1.2.6 More_frags

The more_frags option allows you to specify whether the rule applies 802.11 to frames that have more fragments following them. Valid arguments are ON, OFF, TRUE, or FALSE. For consistency and silliness, '!' may proceed the argument to perform a logical NOT operation on the comparison.

Format: **more_frags:[!] TRUE | FALSE | ON | OFF;**

C.1.2.7 Retry

The retry option allows you to specify whether the rule applies to 802.11 frames that have been retransmitted. Valid arguments are ON, OFF, TRUE, or FALSE. For consistency and silliness, '!' may proceed the argument to perform a logical NOT operation on the comparison.

Format: **retry:[!] TRUE | FALSE | ON | OFF;**

C.1.2.8 Pwr_mgmt

The pwr_mgmt option allows you to specify whether the rule applies to 802.11 frames that indicate whether the transmitting device is in power-save mode. Valid arguments are ON, OFF, TRUE, or FALSE. For consistency and silliness, '!' may proceed the argument to perform a logical NOT operation on the comparison.

Format: **pwr_mgmt:[!] TRUE | FALSE | ON | OFF;**

C.1.2.9 More_data

The more_data option allows you to specify whether the rule applies to 802.11 frames that have their more data control flag set. Valid arguments are ON, OFF, TRUE, or FALSE. For consistency and silliness, '!' may proceed the argument to perform a logical NOT operation on the comparison.

Format: **more_data:[!] TRUE | FALSE | ON | OFF;**

C.1.2.10 WEP

The wep option allows you to specify whether the rule applies to 802.11 frames that have been processed by the WEP algorithm. Valid arguments are ON, OFF, TRUE, or FALSE. For consistency and silliness, '!' may proceed the argument to perform a logical NOT operation on the comparison.

Format: **wep:[!] TRUE | FALSE | ON | OFF;**

C.1.2.11 Order

The order option allows you to specify whether the rule applies to 802.11 frames that are being transmitted using the strictly-ordered service class. Valid arguments are ON, OFF, TRUE, or FALSE. For consistency and silliness, '!' may proceed the argument to perform a logical NOT operation on the comparison.

Format: **order:[!] TRUE | FALSE | ON | OFF;**

C.1.2.12 Duration_id

The duration_id option allows you to specify whether the rule applies to 802.11 frames that either do or do not have a specific value for their Duration/ID field. Values may be specified in either hexadecimal or decimal notation. An 802.11 frame's Duration/ID field is 16 bits wide, thus the minimum value allowed is 0 and the maximum is 65535. In addition, '!' may proceed the argument to specify a logical NOT operation on the comparison.

Format: **duration_id:[!]<number>;**

C.1.2.13 BSSID

The bssid option allows you to specify whether the rule applies to 802.11 frames that either do or do not have a specific BSSID. BSSID's are specified as a 48-bit hexadecimal number (i.e., 0x000000000000 to 0xFFFFFFFFFFFF). In addition, '!' may proceed the argument to specify a logical NOT operation on the comparison.

Format: **bssid:[!]0x00DEADBEEF00;**

C.1.3 Summary of Snort Wireless features

Detection Features:

- Alerts based :
  - send alerts to syslog
  - write the alert in a simple format
  - send alerts to a UNIX socket that another program can listen on
  - Packets logged to a decoded ASCII format or a binary log file
  - send WinPopup alert messages to Windows machines
  - record log in a database

- record log in a cvs file

- Decoding based:
  - ARP decoding
  - Layer2 headers decoding
- Protocol based:
  - Protocols Snort analyses: tcp, udp, icmp, and ip.
  - test the IP header's TTL field value
  - test the IP header's TOS field value
  - test the IP header's fragment ID field for a specific value
  - watch the IP option fields for specific codes
  - test the fragmentation bits of the IP header
  - test the packet's payload size against a value
  - test the TCP flags for certain values
  - test the TCP sequence number field for a specific value
  - test the TCP acknowledgement field for a specific value
  - test the ICMP type field against a specific value
  - test the ICMP code field against a specific value
  - test the ICMP ECHO ID field against a specific value
  - test the ICMP ECHO sequence number against a specific value
- Frame structure based:
  - tests the entire frame control field
  - tests the 802.11 frame's type
  - tests the 802.11 frame's subtype
  - tests the from distribution system frame control flag
  - tests the to distribution system frame control flag
  - tests the more fragments frame control flag
  - tests the retry frame control flag
  - tests the power management frame control flag
  - tests the more data frame control flag
  - tests the wep frame control flag
  - tests the order frame control flag
  - tests the frame's duration/id field
  - tests the frame's BSSID
  - tests the frame's sequence number
  - tests the frame's fragment number
  - tests the frame's 4th address field
- Others:
  - Real-time
  - Promiscuous mode
  - search for a set of patterns in the packet's payload
  - dumps the application layer information for a given session
  - watch RPC services for specific application/procedure calls
  - search for a pattern in the URI portion of a packet
  - Detect portscan

Prevention Features:

- knock down connections
- block web sites

General Features:

- OS : *NIX
- License : Open Source
- Network based
- Passive monitoring of network traffic
- Signature based

C.2 Kismet

C.2.1 Summary of Kismet's features:

Detection Features:

- Real-time
- Networks with known WEP keys can be decrypted in real-time
- Runtime filtering of packets
- tracks probe requests and responses
- Fuzzy encryption detection
- BSSID (MAC address) of the network
- Display Last-advertised channel for network
- Display Number of clients (unique MACs) seen on network
- Display Network status flags
- Display Extra AP information included by some manufacturers
- Display Number of LLC packets
- Identifies Manufacturer and Model of AP and Client
- Display Maximum supported rate as advertised by AP
- Display Name of the network or group
- Display Last seen noise level
- Display Last seen signal level
- Display Network type (Probe, *ad hoc*, Infra, etc)
- Display WEP status (does network indicate it uses WEP)
- Display Last seen IP used by host
- Detects unusual probes
- Detects excessive disassociation
- Detects Netstumbler probe requests
- Detects Deauthenticate/Disassociate Flood
- Detects Lucent link test
- Detects SSID brute force attempt
- Detects AP changing to a new channel

- Detects Broadcast disconnect/deauthenticate
- Detects Invalid BSS timestamps
- Detects weak-iv packet

Prevention Features:

- None (it's a legacy IDS only not an IPS)

General Features:

- OS : *NIX
- License : Open Source
- Network Based
- Passive monitoring of network traffic
- Signature based
- Anomaly based

C.3 Bro-IDS:

The following is the documentation of Bro-IDS. It is required to study this before installing the Bro. The following is a technical review of Bro-IDS system.

C.3.1 Technical Details Overview

Architecturally, Bro is layered into three major components. It uses the now-standard *libpcap* packet-capture library developed by LBNL's Network Research Group to filter the packet stream in the kernel for high performance.

The filtered packets are then passed along to Bro's *event engine*, which reduces the filtered stream into a series of higher-level *events*. These events reflect network activity in *policy-neutral* terms. That is, what has happened but not why or whether it is significant.

Finally, Bro's *policy script interpreter* executes *event handlers* written in the custom *Bro* scripting language. These scripts express a site's security policy, i.e., what actions to take when the monitor detects different types of activity. The scripting language is very powerful, allowing Bro to track detailed information about the network's activity. Scripts can generate real-time *alerts* and can also execute arbitrary programs. This last means that Bro can respond to attacks. For example, at LBNL the operational Bro's configuration will sometimes *block* a remote Internet host from further access to the Laboratory's network, which can be viewed as a form of reactive firewall.

C.3.2. Detection of intrusions

Bro analyses network traffic against rules describing what sort of activity is deemed troublesome. These rules might describe restrictions on activity (e.g., only certain hosts can connect to certain services), policies regarding what activity is worth alerting (e.g., attempts to a given number of different hosts constitutes a "scan"), or signatures describing known attacks or access to known vulnerabilities.

C.3.3 Analysis of network traffic

First Bro filters the traffic, discarding elements of minimal important to its analysis. The remaining information is sent to its "event" engine, where Bro interprets the structure of the network packets and abstracts them into higher-level events describing the activity. Finally, Bro executes policy scripts against the stream of events, looking for activity that the rules indicate should generate alerts or actions, such as possible intrusions.

C.3.4 Bro's events

Events are actions that take place on the network. Examples of events might be a failed connection attempt, a connection established between two computers, a successful authentication and login, a Web browser requesting a URL, or the response sent back by the Web server.

C.3.5 Policy scripts

Policy scripts are programs written in the Bro language. They contain the rules that describe what sorts of activities are deemed troublesome. They analyse the network events and initiate actions based on the analysis.

C.3.6 Action taken by the scripts

Scripts generate a number of output files recording the activity seen on the network (including normal, non-attack activity). They also can generate *alerts* signifying that a problem has been seen. In addition, scripts can *execute programs*, which can terminate existing connections, block traffic from hostile hosts (by inserting blocks into a router access control list), send e-mail messages, or page the on-call staff.

C.3.7 Can Bro automatically block hostile connections?

Bro itself cannot block connections, but by using its ability to execute programs, Bro's detection capabilities can be integrated with network equipment that can create such blocks. For example, Bro can detect a *scan* and send the attacking IP address to an external program that can, in turn, insert an access control block into a router, thus stopping the attacking IP from further scanning.

C.3.8 Bro needs a special language

Bro's language is geared towards network analysis in general and security analysis in particular. It understands network notions such as "connections", "IP addresses", "ports", and various network protocols. Its operators and functions are designed to ease the task of expressing different forms of network analysis.

C.3.9 Summary of Bro IHS features

Identification Features:

- Alerts based :
  - send alerts to administrator:
    - email
    - pager
    - log alerts to event logs
    - log alerts to syslog
- Protocol based:
    - filter for SYN/FIN/RST TCP packets
    - process TCP fragments
    - FTP analysis
    - identify and log TFTP sessions
    - flag HTTP-based worm sources such as Code Red
    - ICMP analysis
    - detailed analysis of http requests
    - detailed analysis of http replies
    - DNS analysis
- Logging Based:
    - detects password scans
    - rlogin/telnet analyser
    - access and record connection events
- Others:
    - Real-time
    - detects vulnerability scans
    - detects port scans

    - record and analyse RPC portmapper requests
    - record and analyse email traffic
    - track software versions
    - looks for blaster worm
    - looks for synflood attacks
    - ssl analyser
    - Detects incoming and outgoing connections that are ssh
    - Looks for backdoors
    - Looks for clear text passwords

Prevention Features:

- terminate a connections
- block a malicious host
- edit router's ACL to block hostile hosts

General Features:

- OS : *NIX
- License : Open Source
- Network Based
- Passive monitoring of network traffic
- Signature based
- Anomaly based

C.4 Proventia:

The following material has been gathered from following website.

http://www.iss.net/products/Proventia_Network_Anomaly_Detection_System/product_main_page.html

C.4.1 Virtual Perimeters

Proventia Network ADS gives you the ability to construct virtual boundaries within your network, hardening and segmenting the network to prevent cross-infections while reducing downtime and cleanup costs.

C.4.2 Safe Vaccine

Proventia ADS allows network teams to perform detailed "what if?" analyses automatically. Pre-emptive filters on your current routing, switching or firewall infrastructure can be applied to stop infections before they can begin, without disrupting business.

C.4.3 Safe Quarantine

Proventia Network ADS provides instant notification of security threats. It can lock down an entire network to an automatic white list of good relationships instantly, dropping 99.99% of attack traffic without disrupting legitimate traffic.

C.4.4 Active Threat Feed

This feature automatically feeds Proventia Network ADS with the latest network and security behavioural intelligence. Once updated, Proventia Network ADS immediately begins analysing network-wide flow data for these new behaviours.

C.5 Airdefence

The following material has been gathered from following website.

http://www.airdefense.net/products/airdefense_ids.shtm

With its patent-pending Multi-Dimensional Detection Engine, AirDefense analyses threats to identify security breaches based on Signature analysis, Protocol analysis, Policy deviation and statistically anomalous behaviour.

AirDefense Guard responds to attacks with ActiveDefense technology, which interfaces with access points to disconnect the attacker's connection to the WLAN.

C.5.1 Not stop Audit & Policy Enforcement

Because new risks can arise with the easy deployment of unauthorised access points or careless break of security policy, wireless LANs should be continuously surveyed to track traffic patterns, ensure network fidelity and identify security vulnerabilities.

With 24x7 monitoring, AirDefense maintains a real-time audit of the network that identifies all WLAN hardware and activity in the area. AirDefense provides network administrators with easy-to-use tools to manage network policies, set alarms and receive detailed status reports. The policy manager is used to define, monitor and enforce business rules for wireless LANs such as:

- **Off-hours traffic**

  Notify security managers of suspicious late-night traffic.

- ***Ad hoc* networking**

  Prohibit the use of peer-to-peer *ad hoc* networks between wireless devices without an access point.

- **Channels**
  Limit access points to operate only on authorised channels.

- **Encryption & Authentication**
  Require all wireless LAN traffic to be encrypted and authenticated with WEP, 802.1x, WPA, LEAP or PEAP.

- **Encryption**
  Require all wireless LAN traffic to be encrypted with WEP or WPA.

With Active Defence Technology, AirDefense interfaces with access points to provide policy enforcement for security and service levels.

Appendix D. IIHSM

In IDMEF structure “analyser” and “manager” are respectively equivalent to “detection subsystem” and “management console” in IIHSM.

D.1 Internationalization and Localisation

IDMEF message formats shall support full internationalization and localisation.

D.1.1 Rationale

Since network security and IHS are areas that cross geographic, political, and cultural boundaries, the IDMEF messages MUST be formatted such that they can be presented to an operator in a local language and adhering to local presentation customs.

D.1.2 Scenario

An IDMEF specification might include numeric event identifiers. An IDMEF implementation might translate these numeric event identifiers into local language descriptions. In cases where the messages contain strings, the information might be represented using the ISO/IEC IS 10646-1 character set and encoded using the UTF-8 transformation format to facilitate internationalization [Wood2007].

D.2 Message Filtering and Aggregation

The format of IDMEF messages MUST support filtering and/or aggregation of data by the manager.

D.2.1 Rationale

Since it is anticipated that some managers might want to perform filtering and/or data aggregation functions on IDMEF messages, the IDMEF messages MUST be structured to facilitate these operations.

D.2.2 Scenario

An IDMEF specification proposal might recommend fixed-format messages with strong numerical semantics. This would lend itself to high- performance filtering and aggregation by the receiving station.

D.3 IDMEF Communication Protocol (IDP) Requirements

D.3.1 Reliable Message Transmission

The IDP must support reliable transmission of messages.

D.3.1.1 Rationale

IDS managers often rely on receipt of data from IDS analysers to do their jobs effectively. Since IDS managers will rely on IDMEF messages for this purpose, it is important that IDP deliver IDMEF messages reliably.

D.3.2 Interaction with Firewalls

The IDP must support transmission of messages between ID components across firewall boundaries without compromising security.

D.3.2.1 Rationale

Since it is expected that firewalls will often be deployed between DMEF capable analysers and their corresponding managers, the ability to relay messages via proxy or other suitable mechanism across firewalls is necessary. Setting up this communication MUST NOT require changes to the intervening firewall(s) that weaken the security of the protected network(s) nor SHOULD this be achieved by mixing IDMEF messages with other kinds of traffic (e.g., by overloading the HTTP POST method) since that would make it difficult for an organisation to apply separate policies to IDMEF traffic and other kinds of traffic.

D.3.2.2 Scenario

One possible design is the use of TCP to convey IDMEF messages. The general goal in this case is to avoid opening dangerous inbound "holes" in the firewall. When the manager is inside the firewall and the analysers are outside the firewall, this is often achieved by having the manager initiate an outbound connection to each analyser. However, it is also possible to place the manager outside the firewall and the analysers on the inside; this can occur when a third-party vendor (such as an ISP) is providing monitoring services to a user. In this case, the outbound connections would be initiated by each analyser to the manager. A mechanism that permits either the manager or the analyser to initiate connections would provide maximum flexibility in manager and analyser deployment.

D.3.3 Mutual Authentication

The IDP must support mutual authentication of the analyser and the manager to each other. Application-layer authentication is required irrespective of the underlying transport layer.

D.3.3.1 Rationale

Since the alert messages are used by a manager to direct responses or further investigation related to the security of an enterprise network, it is important that the receiver have confidence in the identity of the sender and that the sender have confidence in the identity of the receiver. This is peer-to-peer authentication of each party to the other. It MUST NOT be limited to authentication of the underlying communications mechanism, for example, because of the risk that this authentication process might be subverted or misconfigured.

D.3.4 Message Confidentiality

The IDP MUST support confidentiality of the message content during message exchange. The selected design MUST be capable of supporting a variety of encryption algorithms and MUST be adaptable to a wide variety of environments.

D.3.4.1 Rationale

IDMEF messages potentially contain extremely sensitive information (such as passwords) and would be of great interest to an intruder. Since it is likely some of these messages will be transmitted across uncontrolled network segments, it is important that the content be shielded. Furthermore, since the legal environment for encryption technologies is extremely varied and changes often, it is important that the design selected be capable of supporting a number of different encryption options and be adaptable by the user to a variety of environments.

D.3.5 Message Integrity

The IDP MUST ensure the integrity of the message content. The selected design MUST be capable of supporting a variety of integrity mechanisms and MUST be adaptable to a wide variety of environments.

D.3.5.1 Rationale

IDMEF messages are used by the manager to direct action related to the security of the protected enterprise network. It is vital for the manager to be certain that the content of the message has not been changed after transmission.

D.3.6 Per-source Authentication

The IDP MUST support separate authentication keys for each sender. If symmetric algorithms are used, these keys would need to be known to the manager it is communicating with.

D.3.6.1 Rationale

Given that sensitive security information is being exchanged via the IDMEF, it is important that the manager can authenticate each analyser sending alerts.

D.3.7 Denial of Service

The IDP SHOULD resist protocol denial-of-service attacks.

D.3.7.1 Rationale

A common way to defeat secure communications systems is through resource exhaustion. While this does not corrupt valid messages, it can prevent any communication at all. It is desirable that IDP resist such denial-of-service attacks.

D.3.7.2 Scenario

An attacker penetrates a network being defended by an IDS. Although the attacker is not certain that an IDS is present, he is certain that application-level encrypted traffic (i.e., IDMEF traffic) is being exchanged between components on the network being attacked. He decides to mask his presence and disrupt the encrypted communications by initiating one or more flood events. If the IDP can resist such an attack, the probability that the attacker will be stopped increases.

D.3.8 Message Duplication

The IDP SHOULD resist malicious duplication of messages.

D.3.8.1 Rationale

A common way to impair the performance of secure communications mechanisms is to duplicate the messages being sent, even though the attacker might not understand them, in an attempt to confuse the receiver. It is desirable that the IDP resist such message duplication.

D.3.8.2 Scenario

An attacker penetrates a network being defended by an IDS. The attacker suspects that an IDS is present and quickly identifies the encrypted traffic flowing between system components as being a possible threat. Even though she cannot read this traffic, she copies the messages and directs multiple copies at the receiver in an attempt to confuse it. If the IDP resists such message duplication, the probability that the attacker will be stopped increases.

D.3.9 Message Content Requirements

D.3.9.1 Detected Data

There are many different types of IDSs, such as those based on signatures, anomalies, correlation, network monitoring, host monitoring, or application monitoring. The IDMEF design MUST strive to accommodate these diverse approaches by concentrating on conveying *what* an IDS has detected, rather than *how* it detected it.

D.3.9.2 Rationale

There are many types of IDSs that analyse a variety of data sources. Some are profile based and operate on log files, attack signatures, etc. Others are anomaly based and define normal behaviour and detect deviations from the established baseline. Each of these IDSs reports different data that, in part, depends on their intrusion detection methodology. All MUST be supported by this standard.

D.3.10 Event Identity

The content of IDMEF messages MUST contain the identified name of the event (event identity) if it is known. This name MUST be drawn from a standardised list of events (if available) or will be an implementation-specific name if the event identity has not yet been standardised. It is not known how this standardised list will be defined or updated.

Requirements on the creation of this list are beyond our efforts. Other groups within the security arena are investigating the creation of such lists.

D.3.10.1 Rationale

Given that this document presents requirements on standardising ID message formats so that an ID manager is able to receive alerts from analysers from multiple implementations, it is important that the manager understand the semantics of the reported events. There is, therefore, a need to identify known events and store information concerning their methods and possible fixes to these events. Some events are well known and this recognition can help the operator.

D.3.10.2 Scenario

Intruder launches an attack that is detected by two different analysers from two distinct implementations. Both report the same event identity to the ID manager, even though the algorithms used to detect the attack by each analyser might have been different.

D.3.11 Event Background Information

The IDMEF message design MUST include information, which the sender should provide, that allows a receiver to locate background information on the kind of event that is being reported in the alert.

D.3.11.1 Rationale

This information is used by administrators to report and fix problems.

D.3.11.2 Scenario

Attacker performs a well-known attack. A reference to a URL to background information on the attack is included in the IDMEF message. The operator uses this information to initiate repairs on the vulnerable system.

D.3.12 Additional Data

The IDMEF message MUST be able to reference additional detailed data related to this specific underlying event. It is OPTIONAL for implementations to use this field. No requirements are placed on the format or content of this field. It is expected that this will be defined and described by the implementer.

D.3.12.1 Rationale

Operators might want more information on specifics of an event. This field, if filled in by the analyser, MAY point to additional or more detailed information about the event.

D.3.13 Event Source and Target Identity

The IDMEF message MUST contain the identity of the source of the event and target component identifier if it is known. In the case of a network-based event, this will be the source and destination IP address of the session used to launch the event. Note that the Identity of source and target will vary for other types of events, such as those launched/detected at the operating system or application level.

D.3.13.1 Rationale

This will allow the operator to identify the source and target of the event.

D.3.14 Device Address Types

The IDMEF message MUST support the representation of different types of device addresses.

D.3.14.1 Rationale

A device is a uniquely addressable element on the network (i.e., not limited to computers or networks or a specific level of the network protocol hierarchy). In addition, devices involved in an intrusion event might use addresses that are not IP-centric.

D.3.14.2 Scenario

The IDS recognises an intrusion on a particular device and includes both the IP address and the MAC address of the device in the IDMEF message. In another situation, the IDS recognises an intrusion on a device that has only a MAC address and includes only that address in the IDMEF message. Another situation involves analysers in an Asynchronous Transfer Mode (ATM) switch fabric that use E.164 address formats.

D.3.15 Event Impact

The IDMEF message MUST contain an indication of the possible impact of this event on the target. The IDMEF design document MUST define the scope of this value.

D.3.15.1 Rationale

Information concerning the possible impact of the event on the target system provides an indication of what the intruder is attempting to do and is critical data for the operator to perform damage assessment. Not all systems will be able to determine this, but it is important data to transmit for those systems that can. This requirement places no requirements on the list itself (e.g., properties of the list, maintenance, etc.); rather the requirement only specifies that the IDMEF must contain a field for specifying the impact and that the IDMEF must define the scope of such values.

D.3.16 Automatic Response

The IDMEF message MUST provide information about the automatic actions taken by the analyser in response to the event (if any).

D.3.16.1 Rationale

It is very important for the operator to know if there was an automated response and what that response was. This will help determine what further action to take, if any.

D.3.17 Analyser Location

The IDMEF message MUST include information that would make it possible to later identify and locate the individual analyser that reported the event.

D.3.17.1 Rationale

The identity of the detecting analyser often proves to be a valuable piece of data to have in determining how to respond to a particular event.

D.3.17.2 Scenario

One interesting scenario involves the progress of an intrusion event throughout a network. If the same event is detected and reported by multiple analysers, the identity of the analyser (in the case of a network-based analyser) might provide some indication of the network location of the target systems and might warrant a specific type of response. This might be implemented as an IP address.

D.3.18 Analyser Identity

The IDMEF message MUST be able to contain the identity of the implementer and the analyser that detected the event.

D.3.18.1 Rationale

Users might run multiple IDSs to protect their enterprise. This data will help the systems administrator determine which implementer and analyser detected the event.

D.3.18.2 Scenario

Analyser X from implementer Y detects a potential intrusion. A message is sent reporting that it found a potential break-in with X and Y specified. The operator is therefore able to include the known capabilities or weaknesses of analyser X in his decision regarding further action.

D.3.19 Degree of Confidence

The IDMEF message MUST be able to state the degree of confidence of the report. The completion of this field by an analyser is OPTIONAL, as this data might not be available at all analysers.

D.3.19.1 Rationale

Many IDSs contain thresholds to determine whether or not to generate an alert. This might influence the degree of confidence one has in the report or perhaps would indicate the likelihood of the report being a false alarm.

D.3.19.2 Scenario

The alarm threshold monitor is set at a low level to indicate that an organisation wants reports on any suspicious activity, regardless of the probability of a real attack. The degree-of-confidence measure is used to indicate whether this is a low-probability or high-probability event.

D.3.20 Alert Identification

The IDMEF message MUST be uniquely identifiable in that it can be distinguished from other IDMEF messages.

D.3.20.1 Rationale

An IDMEF message might be sent by multiple geographically-distributed analysers at different times. A unique identifier will allow an IDMEF message to be identified efficiently for data reduction and correlation purposes.

D.3.20.2 Scenario

The unique identifier might consist of a unique originator identifier (e.g., IPv4 or IPv6 address) concatenated with a unique sequence number generated by the originator. In a typical IDS deployment, a low-level event analyser will log the raw sensor information into, e.g., a database while analysing and reporting results to higher levels. In this case, the unique raw message identifier can be included in the result message as supporting evidence. Higher-level analysers can later use this identifier to retrieve the raw message from the database if necessary.

D.3.21 Alert Creation Date and Time

The IDMEF MUST support reporting alert creation date and time in each event, where the creation date and time refer to the date and time that the analyser decided to create an alert. The IDMEF MAY support additional dates and times, such as the date and time the event reference by the alert began.

D.3.21.1 Rationale

Time is important from both a reporting and correlation point of view. Event onset time might differ from the alert creation time because it might take some time for the sensor to accumulate information about a monitored activity before generating the event, and additional time for the analyser to receive the event and create an alert. The event onset time is therefore more representative of the actual time that the reported activity began than is the alert creation time.

D.3.21.2 Scenario

If an event is reported in the quiet hours of the night, the operator might assign a higher priority to it than she would to the same event reported in the busy hours of the day. Furthermore, an event (such as a lengthy port scan) may take place over a long period of time and it would be useful for the analyser to report the time of the alert as well as the time the event began.

D.3.22 Time Synchronization

Time SHALL be reported such that events from multiple analysers in different time zones can be received by the same manager and that the local time at the analyser can be inferred.

D.3.22.1 Rationale

For event correlation purposes, it is important that the manager be able to normalize the time information reported in the IDMEF alerts.

D.3.22.2 Scenario

A distributed ID system has analysers located in multiple time zones, all reporting to a single manager. An intrusion occurs that spans multiple time zones as well as multiple analysers. The central manager requires sufficient information to normalize these alerts and determine that all were reported near the same "time" and that they are part of the same attack.

D.3.23 Time Format

The format for reporting the date MUST be compliant with all current standards for Year 2000 rollover, and it MUST have sufficient capability to continue reporting date values past the year 2038.

D.3.23.1 Rationale

It is desirable that the IDMEF have a long lifetime and that implementations be suitable for use in a variety of environments. Therefore, characteristics that limit the lifespan of the IDMEF (such as 2038 date representation limitation) MUST be avoided.

D.3.24 Time Granularity and Accuracy

Time granularity and time accuracy in event messages SHALL NOT be specified by the IDMEF.

D.3.24.1 Rationale

The IDMEF cannot assume a certain clock granularity on sensing elements, and so cannot impose any requirements on the granularity of the event timestamps. Nor can the IDMEF assume that the clocks being used to timestamp the events have a specified accuracy.

D.3.25 Message Extensions

The IDMEF message MUST support an extension mechanism used by implementers to define implementation-specific data. The use of this mechanism by the implementer is OPTIONAL. This data contains implementation-specific information determined by each implementer. The implementer MUST indicate how to interpret these extensions, although there are no specific requirements placed on how implementers describe their implementation-specific extensions. The lack or presence of such message extensions for implementation specific data MUST NOT break interoperation.

D.3.25.1 Rationale

Implementers might wish to supply extra data such as the version number of their product or other data that they believe provides value added due to the specific nature of their product. Implementers may publish a document or web site describing their extensions; they might also use an in-band extension mechanism that is self-describing. Such extensions are not a license to break the interoperation of IDMEF messages.

D.3.26 Message Semantics

The semantics of the IDMEF message MUST be well defined.

D.3.26.1 Rationale

Good semantics are key to understanding what the message is trying to convey so there are no errors. Operators will decide what action to take based on these messages, so it is important that they can interpret them correctly.

D.3.26.2 Scenario

Without this requirement, the operator receives an IDMEF message and interprets it one way. The implementer who constructed the message intended it to have a different meaning from the operator's interpretation. The resulting corrective action is therefore incorrect.

D.3.27 Message Extensibility

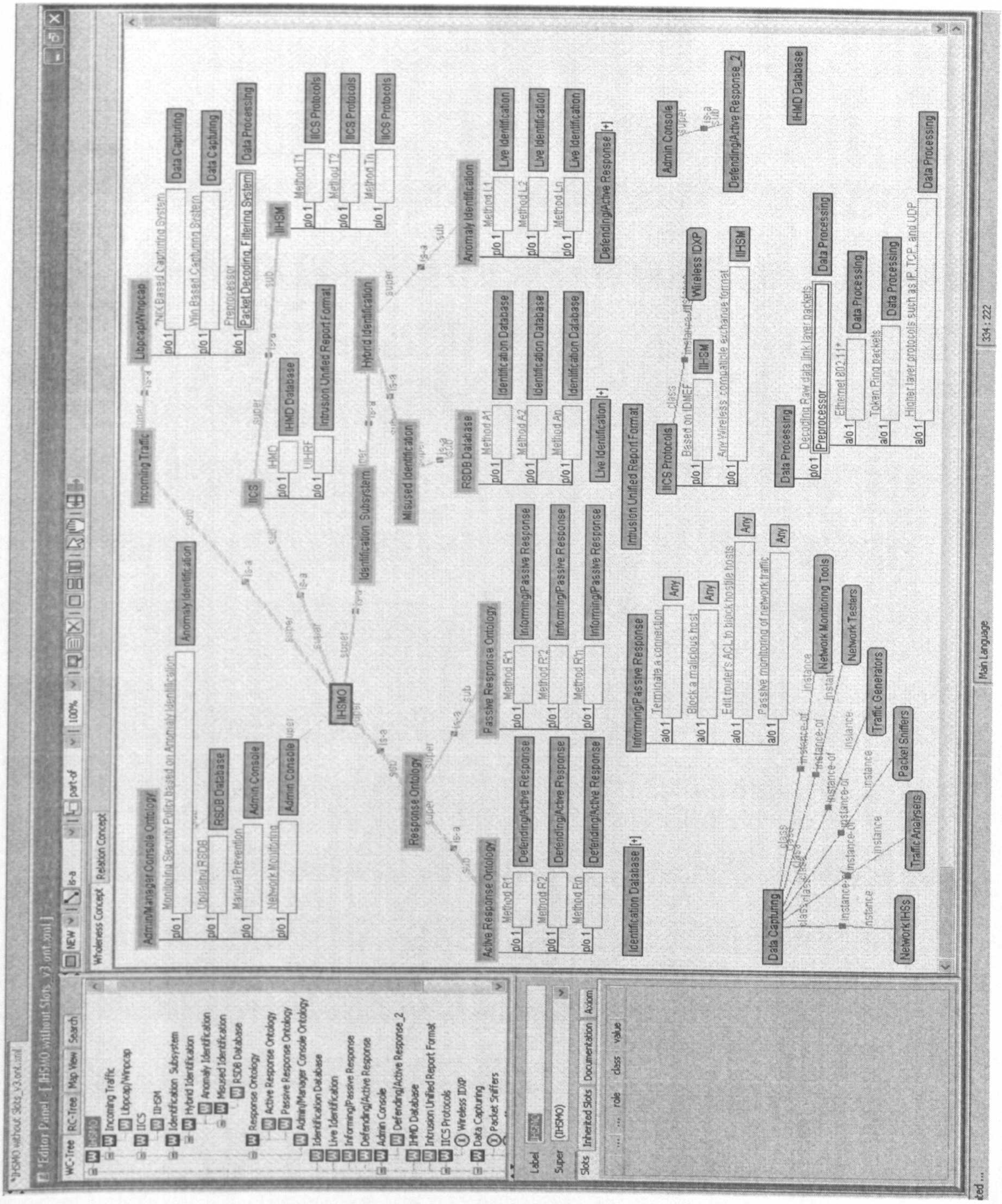
The IDMEF itself MUST be extensible. As new ID technologies emerge and as new information about events becomes available, the IDMEF message format MUST be able to include this new information. Such message extensibility must occur in such a manner that Interoperability is NOT impacted.

D.3.27.1 Rationale

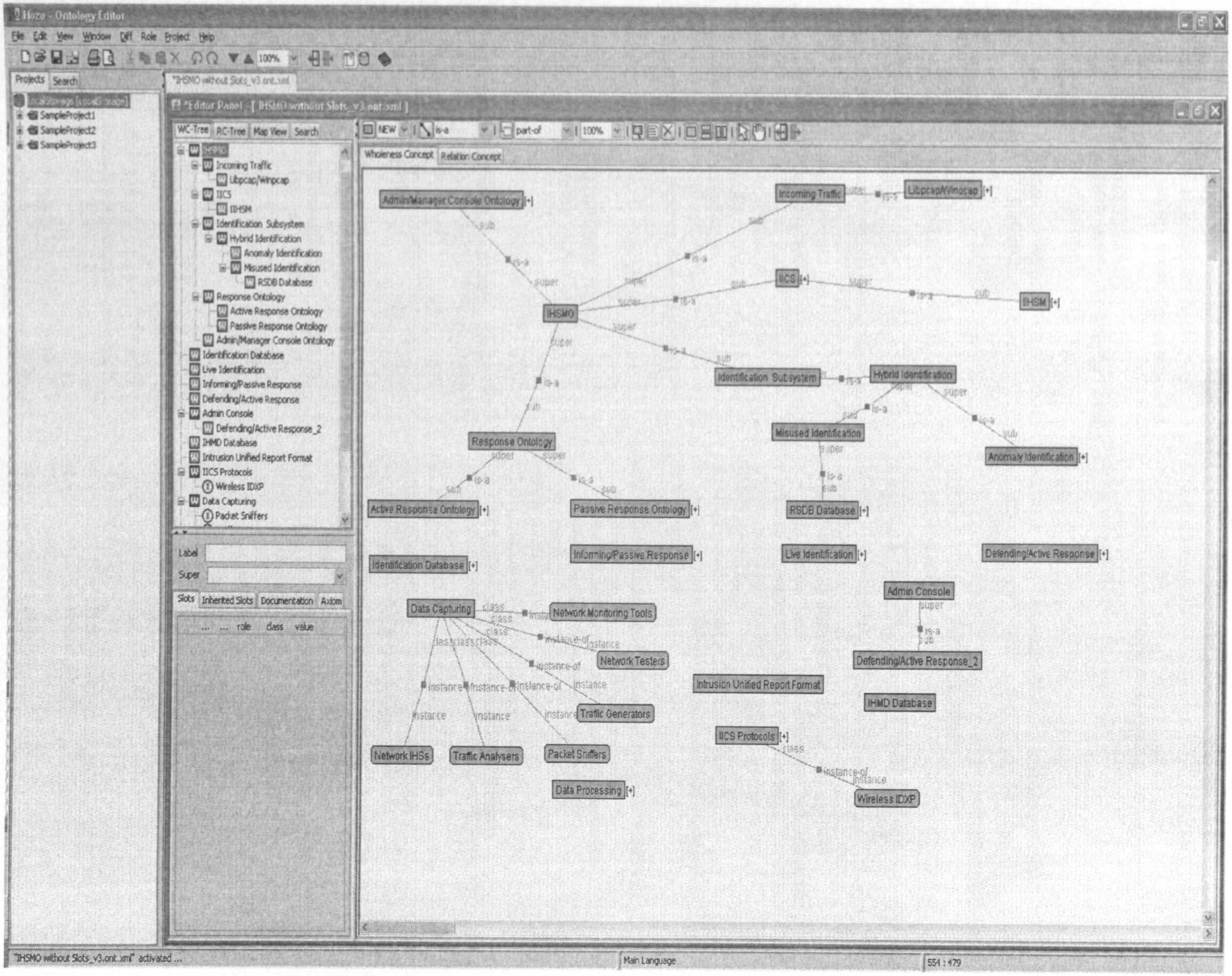
As intrusion detection technology continues to evolve, it is likely that additional information relating to detected events will become available. The IDMEF message format MUST be able to be extended by a specific implementation to encompass this new information. Such extensions are not a license to break the interoperation of IDMEF messages.

Appendix E. IHSMO using Hozo editor

E.1 IHSMO with slots



E.2 IHSMO without slots



"IHSMO without Slots_v3.ont.xml" activated ...

E.3 IHSMO source code with attributes/slots

```
    <?xml version="1.0" encoding="UTF-8" ?>
- <OE_FILE filename="IHSMO with Concepts and
    Slots_v2.ont.xml"
    ont_id="1218910021359ont">
  <FILENAME_ONT>IHSMO with Concepts and
    Slots_v2.ont.xml</FILENAME_ONT>
- <W_CONCEPTS>
  <CANVAS_SIZE w="4000" h="4000" />
- <CONCEPT id="1218901676562_n0">
  <LABEL>IHSMO</LABEL>
  <SUB_LABELS />
  <POS x="312" y="248" />
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
   <SUB_L label="IHSMO" def="" role_name=""
    class_const="" rh_name="" value="" />
   </CONCEPT>
- <CONCEPT id="1218901676562_n1">
  <LABEL>Incoming Traffic</LABEL>
  <SUB_LABELS />
  <POS x="572" y="4" />
  <PART_TREE>close</PART_TREE>
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
  <SUB_L label="Traffic Source Ontology" def=""
    role_name="" class_const="" rh_name=""
    value="" />
   </CONCEPT>
- <CONCEPT id="1218901676562_n2">
  <LABEL>Libpcap/Winpcap</LABEL>
  <SUB_LABELS />
  <POS x="748" y="0" />
- <SLOTS>
- <SLOT id="1218901676562_s3" type="NW"
    label="slot" kind="p/o" num="1" role="*NIX
    Based Capturing System"
    class_constraint="Data Capturing"
    rh_name="" value="">
  <SUB_LABELS />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="" def="" role_name=""
    class_const="Data Capturing" rh_name=""
    value="" />
   </SLOT>
- <SLOT id="1230956039562_s4" type="NW"
    label="slot3" kind="p/o" num="1" role="Win
    Based Capturing System"
    class_constraint="Data Capturing"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Data Capturing" rh_name=""
    value="" />
   </SLOT>
- <SLOT id="1218901676562_s4" type="NW"
    label="slot2" kind="p/o" num="1" role="Pre-
    processor" class_constraint="Data
    Processing" rh_name="Packet Decoding,
    Filtering System" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Data Processing" rh_name=""
    value="" />
   </SLOT>
   </SLOTS>
  <RELATIONS />
  <SUB_L label="Lipcap/Wincap Ontology" def=""
    role_name="" class_const="" rh_name=""
    value="" />
   </CONCEPT>
- <CONCEPT id="1218901676562_n5">
  <LABEL>IICS</LABEL>
  <SUB_LABELS />
  <POS x="572" y="108" />
  <SUB_TREE>open</SUB_TREE>
- <SLOTS>
- <SLOT id="1218901676562_s6" type="NW"
    label="slot" kind="p/o" num="1" role="IHMD"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
   </SLOT>
- <SLOT id="1218901676562_s7" type="NW"
    label="slot2" kind="p/o" num="1"
    role="UIHRF" class_constraint="Intrusion
    Unified Report Format" rh_name=""
    value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Intrusion Unified Report
    Format" rh_name="" value="" />
   </SLOT>
   </SLOTS>
  <RELATIONS />
  <SUB_L label="IICS Ontology" def=""
    role_name="" class_const="" rh_name=""
    value="" />
   </CONCEPT>
- <CONCEPT id="1218901676562_n12">
  <LABEL>Identification Subsystem</LABEL>
  <SUB_LABELS />
  <POS x="488" y="216" />
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
  <SUB_L label="Hybrid Subsystem Ontology"
    def="" role_name="" class_const=""
    rh_name="" value="" />
   </CONCEPT>
- <CONCEPT id="1218901676562_n13">
  <LABEL>Hybrid Identification</LABEL>
  <SUB_LABELS />
  <POS x="704" y="212" />
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
  <SUB_L label="Hybrid Identification Ontology"
     def="" role_name="" class_const=""
     rh_name="" value="" />
    </CONCEPT>
- <CONCEPT id="1218901676562_n14">
  <LABEL>Anomaly Identification</LABEL>
  <SUB_LABELS />
  <POS x="860" y="300" />
- <SLOTS>
- <SLOT id="1218904397000_s15" type="NW"
     label="slot" kind="p/o" num="1"
     role="Method L1" class_constraint="Live
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Live Identification" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1218904397000_s16" type="NW"
     label="slot2" kind="p/o" num="1"
     role="Method L2" class_constraint="Live
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218904397000_s17" type="NW"
     label="slot3" kind="p/o" num="1"
     role="Method Ln" class_constraint="Live
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="" def="" role_name=""
     class_const="Live Identification" rh_name=""
     value="" />
    </SLOT>
    </SLOTS>
  <RELATIONS />
  <SUB_L label="Anomaly Ontology" def=""
     role_name="" class_const="" rh_name=""
     value="" />
    </CONCEPT>
- <CONCEPT id="1218901676562_n15">
  <LABEL>Misuse Identification</LABEL>
  <SUB_LABELS />
  <POS x="568" y="276" />
  <PART_TREE>close</PART_TREE>
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
  <SUB_L label="Misuse Ontology" def=""
     role_name="" class_const="" rh_name=""
     value="" />
    </CONCEPT>
- <CONCEPT id="1218901676562_n16">
  <LABEL>Response Ontology</LABEL>
  <SUB_LABELS />
  <POS x="148" y="288" />
  <PART_TREE>close</PART_TREE>
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
  <SUB_L label="Response Ontology" def=""
     role_name="" class_const="" rh_name=""
     value="" />
    </CONCEPT>
- <CONCEPT id="1218901676562_n17">
  <LABEL>Active Response Ontology</LABEL>
  <SUB_LABELS />
  <POS x="0" y="364" />
- <SLOTS>
- <SLOT id="1218902034796_s18" type="NW"
     label="slot" kind="p/o" num="1"
     role="Method R1"
     class_constraint="Defending/Active
     Response" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Informing/Passive Response"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1218902034796_s19" type="NW"
     label="slot2" kind="p/o" num="1"
     role="Method R2"
     class_constraint="Defending/Active
     Response" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Defending/Active Response"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1218902034796_s20" type="NW"
     label="slot3" kind="p/o" num="1"
     role="Method Rn"
     class_constraint="Defending/Active
     Response" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Defending/Active Response"
     rh_name="" value="" />
    </SLOT>
    </SLOTS>
  <RELATIONS />
  <SUB_L label="Active Response Ontology"
     def="" role_name="" class_const=""
     rh_name="" value="" />
    </CONCEPT>
- <CONCEPT id="1218901676562_n18">
  <LABEL>Passive Response Ontology</LABEL>
  <SUB_LABELS />
  <POS x="288" y="360" />
- <SLOTS>
- <SLOT id="1218902034796_s22" type="NW"
     label="slot" kind="p/o" num="1"
     role="Method R'1"
     class_constraint="Informing/Passive
     Response" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Informing/Passive Response"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1218902034796_s23" type="NW"
     label="slot2" kind="p/o" num="1"
     role="Method R'2"
     class_constraint="Informing/Passive
     Response" rh_name="" value="">
  <SUB_LABELS />
```

```
    <SUB_L label="" def="" role_name=""
        class_const="Informing/Passive Response"
        rh_name="" value="" />
        </SLOT>
 - <SLOT id="1218902034796_s24" type="NW"
        label="slot3" kind="p/o" num="1"
        role="Method R'n"
        class_constraint="Informing/Passive
        Response" rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Informing/Passive Response"
        rh_name="" value="" />
        </SLOT>
        </SLOTS>
    <RELATIONS />
    <SUB_L label="Passive Response Ontology"
        def="" role_name="" class_const=""
        rh_name="" value="" />
        </CONCEPT>
 - <CONCEPT id="1218901676562_n19">
    <LABEL>Admin/Manager Console
        Ontology</LABEL>
    <SUB_LABELS />
    <POS x="0" y="8" />
 - <SLOTS>
 - <SLOT id="1218902034796_s26" type="NW"
        label="slot" kind="p/o" num="1"
        role="Monitoring Security Policy based on
        Anomaly Identification"
        class_constraint="Anomaly Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="slot" def="" role_name="Security
        Policy" class_const="Anomaly Ontology"
        rh_name="" value="" />
        </SLOT>
 - <SLOT id="1218902034796_s27" type="NW"
        label="slot2" kind="p/o" num="1"
        role="Updating RSDB"
        class_constraint="RSDB Database"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="RSDB Database" rh_name=""
        value="" />
        </SLOT>
 - <SLOT id="1218902034796_s28" type="NW"
        label="slot3" kind="p/o" num="1"
        role="Manual Prevention"
        class_constraint="Admin Console" rh_name=""
        value="">
    <SUB_LABELS />
    <PART_TREE>close</PART_TREE>
    <SUB_L label="" def="" role_name=""
        class_const="Admin Console" rh_name=""
        value="" />
        </SLOT>
 - <SLOT id="1218902034796_s29" type="NW"
        label="slot4" kind="p/o" num="1"
        role="Network Monitoring"
        class_constraint="Admin Console" rh_name=""
        value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Admin Console" rh_name=""
        value="" />
        </SLOT>
        </SLOTS>
    <RELATIONS />
    <SUB_L label="Admin/Manager Console
        Ontology" def="" role_name="" class_const=""
        rh_name="" value="" />
        </CONCEPT>
 - <CONCEPT id="1218904047109_n30">
    <LABEL>RSDB Database</LABEL>
    <SUB_LABELS />
    <POS x="588" y="360" />
 - <SLOTS>
 - <SLOT id="1218904397000_s34" type="NW"
        label="slot" kind="p/o" num="1"
        role="Method A1"
        class_constraint="Identification Database"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Identification Database"
        rh_name="" value="" />
        </SLOT>
 - <SLOT id="1218904397000_s35" type="NW"
        label="slot2" kind="p/o" num="1"
        role="Method A2"
        class_constraint="Identification Database"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
        </SLOT>
 - <SLOT id="1218904397000_s36" type="NW"
        label="slot3" kind="p/o" num="1"
        role="Method An"
        class_constraint="Identification Database"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
        </SLOT>
        </SLOTS>
    <RELATIONS />
        </CONCEPT>
 - <CONCEPT id="1218931988812_n37">
    <LABEL>Identification Database</LABEL>
    <SUB_LABELS />
    <POS x="44" y="624" />
 - <SLOTS>
 - <SLOT id="1218931988812_s38" type="NW"
        label="slot" kind="p/o" num="1"
        role="Decoding based (layer2
        Identifications)" class_constraint="Misuse
        Identification" rh_name="" value="">
    <SUB_LABELS />
    <POS x="92" y="520" />
 - <SLOTS>
 - <SLOT id="1218931988812_s39" type="R2-1"
        label="slot" kind="p/o" num="1" role="ARP
        decoding" class_constraint="Misuse
        Identification" rh_name="" value="">
```

```
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Rule Set Data Base" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1218931988812_s40" type="R2-1"
     label="slot2" kind="p/o" num="1"
     role="Layer 2 header decoding"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Misuse Ontology" rh_name=""
     value="" />
    </SLOT>
    </SLOTS>
  <SUB_L label="" def="" role_name=""
     class_const="Misuse Ontology" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1218931988812_s41" type="NW"
     label="slot2" kind="p/o" num="1"
     role="Protocol based (layer3
     Identifications)" class_constraint="Misuse
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <POS x="80" y="692" />
- <SLOTS>
- <SLOT id="1218931988812_s42" type="R2-1"
     label="slot" kind="a/o" num="1"
     role="Protocols Snort analyses: tcp, udp,
     icmp, and ip" class_constraint="Misuse
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s43" type="R2-1"
     label="slot2" kind="a/o" num="1" role="test
     the IP header's TTL field value"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s44" type="R2-1"
     label="slot3" kind="a/o" num="1" role="test
     the IP header's TOS field value"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Misuse Ontology" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1218931988812_s45" type="R2-1"
     label="slot4" kind="a/o" num="1" role="test
     the IP header's fragment ID field for a
     specific value" class_constraint="Misuse
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s46" type="R2-1"
     label="slot5" kind="a/o" num="1"
     role="watch the IP option fields for specific
     codes" class_constraint="Misuse
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s47" type="R2-1"
     label="slot6" kind="a/o" num="1" role="test
     the fragmentation bits of the IP header"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s48" type="R2-1"
     label="slot7" kind="a/o" num="1" role="test
     the packet's payload size against a value"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s49" type="R2-1"
     label="slot8" kind="a/o" num="1" role="test
     the TCP flags for certain values"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s50" type="R2-1"
     label="slot9" kind="a/o" num="1" role="test
     the TCP sequence number field for a
     specific value" class_constraint="Misuse
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s51" type="R2-1"
     label="slot10" kind="a/o" num="1" role="test
     the TCP acknowledgement field for a
     specific value" class_constraint="Misuse
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Misuse Ontology" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1218931988812_s52" type="R2-1"
     label="slot11" kind="a/o" num="1" role="test
     the ICMP type field against a specific value"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Misuse Ontology" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1218931988812_s53" type="R2-1"
     label="slot12" kind="a/o" num="1" role="test
     the ICMP code field against a specific
     value" class_constraint="Misuse
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s54" type="R2-1"
     label="slot13" kind="a/o" num="1" role="test
     the ICMP ECHO ID field against a specific
     value" class_constraint="Misuse
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s55" type="R2-1"
     label="slot14" kind="a/o" num="1" role="test
     the ICMP ECHO sequence number against a
     specific value" class_constraint="Misuse
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Misuse Ontology" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1218931988812_s56" type="R2-1"
     label="slot15" kind="a/o" num="1"
     role="filter for SYN/FIN/RST TCP packets"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s57" type="R2-1"
     label="slot16" kind="a/o" num="1"
     role="process TCP fragments"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s58" type="R2-1"
     label="slot17" kind="a/o" num="1" role="flag
     HTTP-based worm sources such as Code
     Red" class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Misuse Ontology" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1218931988812_s59" type="R2-1"
     label="slot18" kind="a/o" num="1"
     role="detailed analysis of http requests"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s60" type="R2-1"
     label="slot19" kind="a/o" num="1"
     role="detailed analysis of http replies"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
    </SLOTS>
  <SUB_L label="" def="" role_name=""
     class_const="Misuse Ontology" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230837580437_s62" type="NW"
     label="slot3" kind="p/o" num="1"
     role="Frame structure based (layer 2
     Identifications)" class_constraint="Misuse
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <POS x="640" y="664" />
- <SLOTS>
- <SLOT id="1230850840750_s64" type="R2-1"
     label="slot" kind="a/o" num="1" role="tests
     the entire frame control field"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230850840750_s65" type="R2-1"
     label="slot2" kind="a/o" num="1" role="tests
     the 802.11 frame's type"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230850840750_s66" type="R2-1"
     label="slot3" kind="a/o" num="1" role="tests
     the 802.11 frame's subtype"
     class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230850840750_s67" type="R2-1"
     label="slot4" kind="a/o" num="1" role="tests
     the from distribution system frame control
     flag" class_constraint="Misuse Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Misuse Ontology" rh_name=""
     value="" />
```

```
      </SLOT>
  - <SLOT id="1230850840750_s68" type="R2-1"
        label="slot5" kind="a/o" num="1" role="tests
        the to distribution system frame control
        flag" class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Misuse Ontology" rh_name=""
        value="" />
      </SLOT>
  - <SLOT id="1230850840750_s69" type="R2-1"
        label="slot6" kind="a/o" num="1" role="tests
        the more fragments frame control flag"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
      </SLOT>
  - <SLOT id="1230850840750_s70" type="R2-1"
        label="slot7" kind="a/o" num="1" role="tests
        the retry frame control flag"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
      </SLOT>
  - <SLOT id="1230850840750_s71" type="R2-1"
        label="slot8" kind="a/o" num="1" role="tests
        the power management frame control flag"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
      </SLOT>
  - <SLOT id="1230850840750_s72" type="R2-1"
        label="slot9" kind="a/o" num="1" role="tests
        the more data frame control flag"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
      </SLOT>
  - <SLOT id="1230850840750_s73" type="R2-1"
        label="slot10" kind="a/o" num="1"
        role="tests the wep frame control flag"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
      </SLOT>
  - <SLOT id="1230850840750_s74" type="R2-1"
        label="slot11" kind="a/o" num="1"
        role="tests the order frame control flag"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
      </SLOT>
  - <SLOT id="1230850840750_s75" type="R2-1"
        label="slot12" kind="a/o" num="1"
        role="tests the frame's duration/id field"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Misuse Ontology" rh_name=""
        value="" />
      </SLOT>
  - <SLOT id="1230850840750_s76" type="R2-1"
        label="slot13" kind="a/o" num="1"
        role="tests the frame's BSSID"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
      </SLOT>
  - <SLOT id="1230850840750_s77" type="R2-1"
        label="slot14" kind="a/o" num="1"
        role="tests the frame's sequence number"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
      </SLOT>
  - <SLOT id="1230850840750_s78" type="R2-1"
        label="slot15" kind="a/o" num="1"
        role="tests the frame's fragment number"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
      </SLOT>
  - <SLOT id="1230850840750_s79" type="R2-1"
        label="slot16" kind="a/o" num="1"
        role="tests the frame's 4th address field"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
        class_const="Any" rh_name="" value="" />
      </SLOT>
      </SLOTS>
    <SUB_L label="" def="" role_name=""
        class_const="Misuse Ontology" rh_name=""
        value="" />
      </SLOT>
  - <SLOT id="1230852626531_s78" type="NW"
        label="slot4" kind="p/o" num="1"
        role="Other unclassified Identifications"
        class_constraint="Misuse Identification"
        rh_name="" value="">
    <SUB_LABELS />
    <POS x="112" y="1404" />
  - <SLOTS>
  - <SLOT id="1230852626531_s82" type="R2-1"
        label="slot4" kind="a/o" num="1" role="role"
        class_constraint="Any" rh_name="" value="">
    <SUB_LABELS />
```

```
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
     </SLOTS>
   <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
     </SLOT>
     </SLOTS>
     </CONCEPT>
 - <CONCEPT id="1230852626531_n83">
   <LABEL>Live Identification</LABEL>
   <SUB_LABELS />
   <POS x="1144" y="608" />
 - <SLOTS>
 - <SLOT id="1230852626531_s94" type="NW"
      label="slot2" kind="p/o" num="1"
      role="Protocol based (layer3
      Identifications)" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
 - <SLOTS>
 - <SLOT id="1230852626531_s95" type="R2-1"
      label="slot" kind="a/o" num="1" role="FTP
      analysis" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1230852626531_s96" type="R2-1"
      label="slot2" kind="a/o" num="1"
      role="identify and log TFTP sessions"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
     </SLOT>
 - <SLOT id="1230852626531_s97" type="R2-1"
      label="slot3" kind="a/o" num="1" role="ICMP
      analysis" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1230852626531_s98" type="R2-1"
      label="slot4" kind="a/o" num="1" role="DNS
      analysis" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
     </SLOT>
 - <SLOT id="1230852626531_s99" type="R2-1"
      label="slot5" kind="a/o" num="1"
      role="search for a set of patterns in the
      packet's payload" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
     </SLOTS>
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
     </SLOT>
 - <SLOT id="1230855124031_s100" type="NW"
      label="slot3" kind="p/o" num="1"
      role="Logging based"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <POS x="684" y="512" />
 - <SLOTS>
 - <SLOT id="1230855124031_s101" type="R2-1"
      label="slot" kind="a/o" num="1"
      role="detects password scans"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1230855124031_s102" type="R2-1"
      label="slot2" kind="a/o" num="1"
      role="analysing rlogin/telnet"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1230855124031_s103" type="R2-1"
      label="slot3" kind="a/o" num="1"
      role="access and record connection events"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
     </SLOTS>
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
     </SLOT>
 - <SLOT id="1230855124031_s104" type="NW"
      label="slot4" kind="a/o" num="1"
      role="Frame structure based (layer2
      Identification)" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <POS x="1188" y="908" />
 - <SLOTS>
 - <SLOT id="1230855124031_s105" type="R2-1"
      label="slot" kind="a/o" num="1"
      role="Networks with known WEP keys can
      be decrypted in realtime"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
```

```
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s106" type="R2-1"
      label="slot2" kind="a/o" num="1"
      role="tracks probe requests and responses"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s107" type="R2-1"
      label="slot3" kind="p/o" num="1"
      role="Fuzzy encryption detection(UIHRF)"
      class_constraint="IHMD Database"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s108" type="R2-1"
      label="slot4" kind="p/o" num="1"
      role="BSSID (MAC address) of the
      network(UIHRF)" class_constraint="IHMD
      Database" rh_name="" value="">
  <SUB_LABELS />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="" def="" role_name=""
      class_const="IHMD Database" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s109" type="R2-1"
      label="slot5" kind="a/o" num="1"
      role="Display Last-advertised channel for
      network" class_constraint="Anomaly
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s110" type="R2-1"
      label="slot6" kind="a/o" num="1"
      role="Display Number of clients (unique
      MACs) seen on network"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s111" type="R2-1"
      label="slot7" kind="a/o" num="1"
      role="Display Network status flags"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s112" type="R2-1"
      label="slot8" kind="a/o" num="1"
      role="Display Extra AP information included
      by some manufacturers"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s113" type="R2-1"
      label="slot9" kind="a/o" num="1"
      role="Display Number of LLC packets"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s114" type="R2-1"
      label="slot10" kind="a/o" num="1"
      role="Identifies Manufacturer and Model pf
      AP and Client" class_constraint="Anomaly
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s115" type="R2-1"
      label="slot11" kind="a/o" num="1"
      role="Display Maximum supported rate as
      advertised by AP" class_constraint="Anomaly
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s116" type="R2-1"
      label="slot12" kind="a/o" num="1"
      role="Display Maximum supported rate as
      advertised by AP" class_constraint="Anomaly
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s117" type="R2-1"
      label="slot13" kind="a/o" num="1"
      role="Display Name of the network or
      group" class_constraint="Anomaly
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s118" type="R2-1"
      label="slot14" kind="a/o" num="1"
      role="Display Last seen noise level"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s119" type="R2-1"
      label="slot15" kind="a/o" num="1"
    role="Display Last seen signal level"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Anomaly Ontology" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s120" type="R2-1"
    label="slot16" kind="a/o" num="1"
    role="Display Network type (Probe, Adhoc,
    Infra, etc)" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Anomaly Ontology" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s121" type="R2-1"
    label="slot17" kind="a/o" num="1"
    role="Display WEP status"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Anomaly Ontology" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s122" type="R2-1"
    label="slot18" kind="p/o" num="1"
    role="Identifies unusual probes (UIHRF)"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s123" type="R2-1"
    label="slot19" kind="p/o" num="1"
    role="Identifies unusual probes (UIHRF)"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1612" />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s124" type="R2-1"
    label="slot20" kind="p/o" num="1"
    role="Identifies excessive
    disassociation(UIHRF)"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <POS x="1232" y="1640" />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s125" type="R2-1"
    label="slot21" kind="p/o" num="1"
    role="Identifies Netstumbler probe
    requests(UIHRF)" class_constraint="IHMD
    Database" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1232" y="1672" />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s126" type="R2-1"
    label="slot22" kind="p/o" num="1"
    role="Identifies
    Deauthenticate/Disassociate
    Flood(UIHRF)" class_constraint="IHMD
    Database" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1712" />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s127" type="R2-1"
    label="slot23" kind="p/o" num="1"
    role="Identifies Lucent link test(UIHRF)"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1744" />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s128" type="R2-1"
    label="slot24" kind="p/o" num="1"
    role="Identifies SSID brute force
    attempt(UIHRF)" class_constraint="IHMD
    Database" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1236" y="1816" />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s129" type="R2-1"
    label="slot25" kind="p/o" num="1"
    role="Identifies AP changing to a new
    channel(UIHRF)" class_constraint="IHMD
    Database" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1848" />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s130" type="R2-1"
    label="slot26" kind="p/o" num="1"
    role="Identifies Broadcast
    disconnect/deauthenticate(UIHRF)"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1884" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
   </SLOT>
- <SLOT id="1230855124031_s131" type="R2-1"
     label="slot27" kind="p/o" num="1"
     role="Identifies Invalid BSS
     timestamps(UIHRF)" class_constraint="IHMD
     Database" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1920" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
   </SLOT>
- <SLOT id="1230855124031_s132" type="R2-1"
     label="slot28" kind="p/o" num="1"
     role="Identifies weak-iv packet (UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="1248" y="1952" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
   </SLOT>
- <SLOT id="1230855124031_s133" type="R2-1"
     label="slot29" kind="p/o" num="1"
     role="Identifies MITM attacks(UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1780" />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
   </SLOT>
- <SLOT id="1230855124031_s134" type="R2-1"
     label="slot30" kind="p/o" num="1"
     role="Identifies MAC spoofing attacks
     (UIHRF)" class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1580" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
   </SLOT>
   </SLOTS>
  <SUB_L label="" def="" role_name=""
     class_const="Anomaly Ontology" rh_name=""
     value="" />
   </SLOT>
- <SLOT id="1230855124031_s135" type="NW"
     label="slot5" kind="p/o" num="1"
     role="Other unclassified Identifications"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="772" y="1704" />
- <SLOTS>
- <SLOT id="1230855124031_s136" type="R2-1"
     label="slot" kind="p/o" num="1"
     role="Identifies vulnerability scans(UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="816" y="1768" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
   </SLOT>
- <SLOT id="1230855124031_s138" type="R2-1"
     label="slot3" kind="p/o" num="1"
     role="Identifies port scans(UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="812" y="1804" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
   </SLOT>
- <SLOT id="1230855124031_s139" type="R2-1"
     label="slot4" kind="a/o" num="1"
     role="watch & record RPC services for
     specific application/procedure calls"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="248" y="1928" />
  <SUB_L label="" def="" role_name=""
     class_const="Anomaly Ontology" rh_name=""
     value="" />
   </SLOT>
- <SLOT id="1230855124031_s140" type="R2-1"
     label="slot5" kind="a/o" num="1"
     role="record and analyse email traffic"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="812" y="1860" />
  <SUB_L label="" def="" role_name=""
     class_const="Anomaly Ontology" rh_name=""
     value="" />
   </SLOT>
- <SLOT id="1230855124031_s141" type="R2-1"
     label="slot6" kind="a/o" num="1" role="track
     software versions"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="820" y="1900" />
  <SUB_L label="" def="" role_name=""
     class_const="Anomaly Ontology" rh_name=""
     value="" />
   </SLOT>
- <SLOT id="1230855124031_s142" type="R2-1"
     label="slot7" kind="p/o" num="1"
     role="Identifies for blaster worm(UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="812" y="1940" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
```

```
    </SLOT>
- <SLOT id="1230855124031_s143" type="R2-1"
      label="slot8" kind="p/o" num="1"
      role="Identifies synflood attacks(UIHRF)"
      class_constraint="IHMD Database"
      rh_name="" value="">
   <SUB_LABELS />
   <POS x="320" y="1964" />
   <SUB_L label="" def="" role_name=""
      class_const="IHMD Database" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s144" type="R2-1"
      label="slot9" kind="a/o" num="1" role="SSL
      analyser" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <POS x="324" y="1728" />
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s145" type="R2-1"
      label="slot10" kind="p/o" num="1"
      role="Identifies incoming and outgoing
      connections that are ssh(UIHRF)"
      class_constraint="IHMD Database"
      rh_name="" value="">
   <SUB_LABELS />
   <POS x="264" y="1760" />
   <SUB_L label="" def="" role_name=""
      class_const="IHMD Database" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s146" type="R2-1"
      label="slot11" kind="p/o" num="1"
      role="Identifies backdoors(UIHRF)"
      class_constraint="IHMD Database"
      rh_name="" value="">
   <SUB_LABELS />
   <POS x="324" y="1788" />
   <SUB_L label="" def="" role_name=""
      class_const="IHMD Database" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s147" type="R2-1"
      label="slot12" kind="a/o" num="1"
      role="Identifies clear text passwords"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <POS x="324" y="1820" />
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230858698968_s148" type="R2-1"
      label="slot13" kind="a/o" num="1"
      role="Promiscuous mode"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <POS x="320" y="1852" />
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230858698968_s149" type="R2-1"
      label="slot14" kind="a/o" num="1"
      role="search for a pattern in the URI
      portion of a packet"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <POS x="320" y="1888" />
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
    </SLOTS>
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230859515843_s137" type="NW"
      label="slot" kind="p/o" num="1" role="Alert
      based" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <POS x="600" y="1320" />
- <SLOTS>
- <SLOT id="1230859515843_s138" type="R2-1"
      label="slot" kind="a/o" num="1" role="send
      alerts to syslog" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="slot" def="" role_name="send
      alerts to syslog" class_const="Any"
      rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s139" type="R2-1"
      label="slot2" kind="p/o" num="1" role="write
      the alert in a UIHRF format"
      class_constraint="IHMD Database"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="slot2" def="" role_name="write
      the alert in a simple format"
      class_const="IHMD Database" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230859515843_s140" type="R2-1"
      label="slot3" kind="a/o" num="1" role="send
      alerts to a *NIX socket that another
      program can listen on"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="slot3" def="" role_name="send
      alerts to a *NIX socket that another
      program can listen on" class_const="Any"
      rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s141" type="R2-1"
      label="slot4" kind="a/o" num="1"
      role="Packets logged to a decoded ASCII
      format or a binary log file"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot4" def="" role_name="Packets
    logged to a decoded ASCII format or a
    binary log file" class_const="Any"
    rh_name="" value="" />
   </SLOT>
- <SLOT id="1230859515843_s142" type="R2-1"
    label="slot5" kind="a/o" num="1" role="send
    WinPopup alert messages to Windows
    machines" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot5" def="" role_name="send
    WinPopup alert messages to Windows
    machines" class_const="Any" rh_name=""
    value="" />
   </SLOT>
- <SLOT id="1230859515843_s143" type="R2-1"
    label="slot6" kind="a/o" num="1"
    role="record log in a database"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot6" def="" role_name="record
    log in a database" class_const="Any"
    rh_name="" value="" />
   </SLOT>
- <SLOT id="1230859515843_s144" type="R2-1"
    label="slot7" kind="a/o" num="1"
    role="record log in a cvs file"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot7" def="" role_name="record
    log in a cvs file" class_const="Any"
    rh_name="" value="" />
   </SLOT>
- <SLOT id="1230859515843_s145" type="R2-1"
    label="slot8" kind="a/o" num="1"
    role="sending alert to administrator's email
    or pager" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot8" def="" role_name="sending
    alert to administrator's email or pager"
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230859515843_s146" type="R2-1"
    label="slot9" kind="a/o" num="1" role="log
    alerts to event logs"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot9" def="" role_name="log
    alerts to event logs" class_const="Anomaly
    Ontology" rh_name="" value="" />
    </SLOT>
   </SLOTS>
  <SUB_L label="slot" def="" role_name="Alert
    based" class_const="Anomaly Ontology"
    rh_name="" value="" />
   </SLOT>
   </SLOTS>
   </CONCEPT>
- <CONCEPT id="1230859515843_n147">
  <LABEL>Informing/Passive Response</LABEL>
  <SUB_LABELS />
  <POS x="0" y="1520" />
- <SLOTS>
- <SLOT id="1230859515843_s148" type="NW"
    label="slot" kind="a/o" num="1"
    role="Terminate a connection"
    class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230859515843_s149" type="NW"
    label="slot2" kind="a/o" num="1"
    role="Block a malicious host"
    class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230859515843_s150" type="NW"
    label="slot3" kind="a/o" num="1" role="Edit
    router's ACL to block hostile hosts"
    class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230859515843_s151" type="NW"
    label="slot4" kind="a/o" num="1"
    role="Passive monitoring of network traffic"
    class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
   </SLOTS>
   </CONCEPT>
- <CONCEPT id="1230859515843_n152">
  <LABEL>Defending/Active Response</LABEL>
  <SUB_LABELS />
  <POS x="1172" y="0" />
- <SLOTS>
- <SLOT id="1230859515843_s153" type="NW"
    label="slot" kind="p/o" num="1" role="Alert
    based" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
- <SLOTS>
- <SLOT id="1230859515843_s154" type="R2-2"
    label="slot" kind="a/o" num="1" role="send
    alerts to syslog" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="slot" def="" role_name="send
    alerts to syslog"
    class_const="Informing/Passive Response"
    rh_name="" value="" />
   </SLOT>
```

```
- <SLOT id="1230859515843_s155" type="R2-2"
     label="slot2" kind="p/o" num="1" role="write
     the alert in a simple format(UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot2" def="" role_name="write
     the alert in a simple format"
     class_const="Anomaly Ontology" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230859515843_s156" type="R2-2"
     label="slot3" kind="a/o" num="1" role="send
     alerts to a *NIX socket that another
     program can listen on"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot3" def="" role_name="send
     alerts to a *NIX socket that another
     program can listen on"
     class_const="Informing/Passive Response"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s157" type="R2-2"
     label="slot4" kind="a/o" num="1"
     role="Packets logged to a decoded ASCII
     format or a binary log file"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot4" def="" role_name="Packets
     logged to a decoded ASCII format or a
     binary log file"
     class_const="Informing/Passive Response"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s158" type="R2-2"
     label="slot5" kind="a/o" num="1" role="send
     WinPopup alert messages to Windows
     machines" class_constraint="Anomaly
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot5" def="" role_name="send
     WinPopup alert messages to Windows
     machines" class_const="Informing/Passive
     Response" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s159" type="R2-1"
     label="slot6" kind="a/o" num="1"
     role="record log in a database"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot6" def="" role_name="record
     log in a database"
     class_const="Informing/Passive Response"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s160" type="R2-1"
     label="slot7" kind="a/o" num="1"
     role="record log in a cvs file"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot7" def="" role_name="record
     log in a cvs file"
     class_const="Informing/Passive Response"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s161" type="R2-1"
     label="slot8" kind="a/o" num="1"
     role="sending alert to administrator's email
     or pager" class_constraint="Anomaly
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot8" def="" role_name="sending
     alert to administrator's email or pager"
     class_const="Informing/Passive Response"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s162" type="R2-1"
     label="slot9" kind="a/o" num="1" role="log
     alerts to event logs"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot9" def="" role_name="log
     alerts to event logs"
     class_const="Informing/Passive Response"
     rh_name="" value="" />
    </SLOT>
    </SLOTS>
  <SUB_L label="slot" def="" role_name="Alert
     based" class_const="Anomaly Ontology"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s163" type="NW"
     label="slot2" kind="a/o" num="1"
     role="Terminate a connection"
     class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot" def=""
     role_name="Terminate a connection"
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s164" type="NW"
     label="slot22" kind="a/o" num="1"
     role="Block a malicious host"
     class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot2" def="" role_name="Block a
     malicious host" class_const="Any"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s165" type="NW"
     label="slot3" kind="a/o" num="1" role="Edit
     router's ACL to block hostile hosts"
     class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot3" def="" role_name="Edit
     router's ACL to block hostile hosts"
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s166" type="NW"
     label="slot4" kind="a/o" num="1"
     role="dumps the application layer
     information for a given session"
     class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot4" def="" role_name="dumps
     the application layer information for a
     given session" class_const="Any" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230859515843_s167" type="NW"
     label="slot5" kind="a/o" num="1" role="block
     web sites" class_constraint="Any" rh_name=""
     value="">
  <SUB_LABELS />
  <SUB_L label="slot5" def="" role_name="block
     web sites" class_const="Any" rh_name=""
     value="" />
    </SLOT>
    </SLOTS>
  <SUB_L label="Informing/Passive Response"
     def="" role_name="" class_const=""
     rh_name="" value="" />
    </CONCEPT>
- <CONCEPT id="1230938127140_n167">
  <LABEL>Admin Console</LABEL>
  <SUB_LABELS />
  <POS x="500" y="1380" />
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
    </CONCEPT>
- <CONCEPT id="1230938127140_n168">
  <LABEL>Defending/Active
     Response_2</LABEL>
  <SUB_LABELS />
  <POS x="376" y="1460" />
  <RELATIONS />
    </CONCEPT>
- <CONCEPT id="1230946789609_n165">
  <LABEL>IHMD Database</LABEL>
  <SUB_LABELS />
  <POS x="296" y="1516" />
    </CONCEPT>
- <CONCEPT id="1230946789609_n166">
  <LABEL>Intrusion Unified Report
     Format</LABEL>
  <SUB_LABELS />
  <POS x="300" y="1552" />
    </CONCEPT>
- <CONCEPT id="1230946789609_n167">
  <LABEL>IICS Protocols</LABEL>
  <SUB_LABELS />
  <POS x="308" y="1592" />
  <SUB_TREE>open</SUB_TREE>
- <SLOTS>
- <SLOT id="1230946789609_s168" type="NW"
     label="slot" kind="p/o" num="1" role="Based
     on IDMEF" class_constraint="IIHSM"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="IIHSM" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230946789609_s169" type="NW"
     label="slot2" kind="p/o" num="1" role="Any
     Wireless compatible exchange format"
     class_constraint="IIHSM" rh_name=""
     value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
    </SLOTS>
  <RELATIONS />
    </CONCEPT>
- <CONCEPT id="1230946789609_n170"
     instantiation="true">
  <LABEL>Wireless IDXP</LABEL>
  <SUB_LABELS />
  <POS x="504" y="1596" />
  <RELATIONS />
    </CONCEPT>
- <CONCEPT id="1230946789609_n171">
  <LABEL>IIHSM</LABEL>
  <SUB_LABELS />
  <POS x="908" y="132" />
- <SLOTS>
- <SLOT id="1230946789609_s172" type="NW"
     label="slot" kind="p/o" num="1"
     role="Method T1" class_constraint="IICS
     Protocols" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="IICS Protocol" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230946789609_s173" type="NW"
     label="slot2" kind="p/o" num="1"
     role="Method T2" class_constraint="IICS
     Protocols" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230946789609_s174" type="NW"
     label="slot3" kind="p/o" num="1"
     role="Method Tn" class_constraint="IICS
     Protocols" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
    </SLOTS>
  <RELATIONS />
    </CONCEPT>
- <CONCEPT id="1230990225468_n176">
  <LABEL>Data Capturing</LABEL>
  <SUB_LABELS />
  <POS x="12" y="1782" />
  <PART_TREE>close</PART_TREE>
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
  <SUB_L label="Data Capturing" def=""
     role_name="" class_const="" rh_name=""
     value="" />
    </CONCEPT>
- <CONCEPT id="1230990225468_n177"
     instantiation="true">
  <LABEL>Packet Sniffers</LABEL>
```

```
    <SUB_LABELS />
    <POS x="134" y="1881" />
    <RELATIONS />
      </CONCEPT>
- <CONCEPT id="1230990225468_n178"
      instantiation="true">
    <LABEL>Traffic Generators</LABEL>
    <SUB_LABELS />
    <POS x="169" y="1808" />
    <RELATIONS />
      </CONCEPT>
- <CONCEPT id="1230990225468_n179"
      instantiation="true">
    <LABEL>Network Testers</LABEL>
    <SUB_LABELS />
    <POS x="181" y="1719" />
    <RELATIONS />
      </CONCEPT>
- <CONCEPT id="1230990225468_n180"
      instantiation="true">
    <LABEL>Traffic Analysers</LABEL>
    <SUB_LABELS />
    <POS x="6" y="1884" />
    <RELATIONS />
      </CONCEPT>
- <CONCEPT id="1230990225468_n181"
      instantiation="true">
    <LABEL>Network Monitoring Tools</LABEL>
    <SUB_LABELS />
    <POS x="94" y="1684" />
    <RELATIONS />
      </CONCEPT>
- <CONCEPT id="1230990225468_n182"
      instantiation="true">
    <LABEL>Network IHSs</LABEL>
    <SUB_LABELS />
    <POS x="0" y="1704" />
    <RELATIONS />
      </CONCEPT>
- <CONCEPT id="1230990796046_n183">
    <LABEL>Data Processing</LABEL>
    <SUB_LABELS />
    <POS x="87" y="1968" />
- <SLOTS>
- <SLOT id="1230990796046_s184" type="NW"
      label="slot" kind="p/o" num="1"
      role="Decoding Raw data link layer packets"
      class_constraint="Data Processing"
      rh_name="Preprocessor" value="">
    <SUB_LABELS />
- <SLOTS>
- <SLOT id="1230990796046_s185" type="R2-1"
      label="slot" kind="a/o" num="1"
      role="Ethernet 802.11*"
      class_constraint="Data Processing"
      rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
      </SLOT>
- <SLOT id="1230990796046_s186" type="R2-1"
      label="slot2" kind="a/o" num="1"
      role="Token Ring packets"
      class_constraint="Data Processing"
      rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
      class_const="Data Processing" rh_name=""
      value="" />
      </SLOT>
- <SLOT id="1230990796046_s187" type="R2-1"
      label="slot3" kind="a/o" num="1"
      role="Higher layer protocols such as IP,
      TCP, and UDP" class_constraint="Data
      Processing" rh_name="" value="">
    <SUB_LABELS />
    <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
      </SLOT>
      </SLOTS>
    <SUB_L label="" def="" role_name=""
      class_const="Data Processing" rh_name=""
      value="" />
      </SLOT>
      </SLOTS>
      </CONCEPT>
    <ISA id="1218901676562_isa20"
      parent="IHSMO" child="Incoming Traffic" />
    <ISA id="1218901676562_isa21"
      parent="Incoming Traffic"
      child="Libpcap/Winpcap" />
    <ISA id="1218901676562_isa22"
      parent="IHSMO" child="IICS" />
    <ISA id="1218901676562_isa24"
      parent="IHSMO" child="Identification
      Subsystem" />
    <ISA id="1218901676562_isa25"
      parent="Identification Subsystem"
      child="Hybrid Identification" />
    <ISA id="1218901676562_isa26"
      parent="Hybrid Identification"
      child="Anomaly Identification" />
    <ISA id="1218901676562_isa27"
      parent="Hybrid Identification" child="Misuse
      Identification" />
    <ISA id="1218901676562_isa28"
      parent="IHSMO" child="Response Ontology"
      />
    <ISA id="1218901676562_isa29"
      parent="Response Ontology" child="Active
      Response Ontology" />
    <ISA id="1218901676562_isa30"
      parent="Response Ontology" child="Passive
      Response Ontology" />
    <ISA id="1218901676562_isa31"
      parent="IHSMO" child="Admin/Manager
      Console Ontology" />
    <ISA id="1218904047109_isa43"
      parent="Misuse Identification" child="RSDB
      Database" />
    <ISA id="1230938127140_isa185"
      parent="Admin Console"
      child="Defending/Active Response_2" />
    <ISA id="1230946789609_isa188" parent="IICS
      Protocols" child="Wireless IDXP" />
    <ISA id="1230946789609_isa189"
      parent="IICS" child="IIHSM" />
```

```
  <ISA id="1230990225468_isa198"
     parent="Data Capturing" child="Packet
     Sniffers" />
  <ISA id="1230990225468_isa199"
     parent="Data Capturing" child="Traffic
     Generators" />
  <ISA id="1230990225468_isa200"
     parent="Data Capturing" child="Network
     Testers" />
  <ISA id="1230990225468_isa201"
     parent="Data Capturing" child="Traffic
     Analysers" />
  <ISA id="1230990225468_isa202"
     parent="Data Capturing" child="Network
     Monitoring Tools" />
  <ISA id="1230990225468_isa203"
     parent="Data Capturing" child="Network
     IHSs" />
    </W_CONCEPTS>
- <R_CONCEPTS>
  <CANVAS_SIZE w="2000" h="2000" />


    </R_CONCEPTS>
    </OE_FILE>
```

E.4 IHSMO source codes without attributes/slots

```
    <?xml version="1.0" encoding="UTF-8" ?>
- <OE_FILE filename="IHSMO without
     Slots_v2.ont.xml"
     ont_id="1218910021359ont">
  <FILENAME_ONT>IHSMO without
     Slots_v2.ont.xml</FILENAME_ONT>
- <W_CONCEPTS>
  <CANVAS_SIZE w="4000" h="4000" />
- <CONCEPT id="1218901676562_n0">
  <LABEL>IHSMO</LABEL>
  <SUB_LABELS />
  <POS x="312" y="248" />
  <PART_TREE>close</PART_TREE>
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
  <SUB_L label="IHSMO" def="" role_name=""
     class_const="" rh_name="" value="" />
    </CONCEPT>
- <CONCEPT id="1218901676562_n1">
  <LABEL>Incoming Traffic</LABEL>
  <SUB_LABELS />
  <POS x="572" y="4" />
  <PART_TREE>close</PART_TREE>
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
  <SUB_L label="Traffic Source Ontology" def=""
     role_name="" class_const="" rh_name=""
     value="" />
    </CONCEPT>
- <CONCEPT id="1218901676562_n2">
  <LABEL>Libpcap/Winpcap</LABEL>
  <SUB_LABELS />
  <POS x="748" y="0" />
  <PART_TREE>close</PART_TREE>
- <SLOTS>
- <SLOT id="1218901676562_s3" type="NW"
     label="slot" kind="p/o" num="1" role="*NIX
     Based Capturing System"
     class_constraint="Data Capturing"
     rh_name="" value="">
  <SUB_LABELS />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="" def="" role_name=""
     class_const="Data Capturing" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230956039562_s4" type="NW"
     label="slot3" kind="p/o" num="1" role="Win
     Based Capturing System"
     class_constraint="Data Capturing"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Data Capturing" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1218901676562_s4" type="NW"
     label="slot2" kind="p/o" num="1"
     role="Preprocessor" class_constraint="Data
     Processing" rh_name="Packet Decoding,
     Filtering System" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Data Processing" rh_name=""
     value="" />
    </SLOT>
    </SLOTS>
  <RELATIONS />
  <SUB_L label="Lipcap/Wincap Ontology" def=""
     role_name="" class_const="" rh_name=""
     value="" />
    </CONCEPT>
- <CONCEPT id="1218901676562_n5">
  <LABEL>IICS</LABEL>
  <SUB_LABELS />
  <POS x="572" y="108" />
  <PART_TREE>close</PART_TREE>
  <SUB_TREE>open</SUB_TREE>
- <SLOTS>
- <SLOT id="1218901676562_s6" type="NW"
     label="slot" kind="p/o" num="1" role="IHMD"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1218901676562_s7" type="NW"
     label="slot2" kind="p/o" num="1"
     role="UIHRF" class_constraint="Intrusion
```

```
      Unified Report Format" rh_name=""
      value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Intrusion Unified Report
      Format" rh_name="" value="" />
     </SLOT>
     </SLOTS>
   <RELATIONS />
   <SUB_L label="IICS Ontology" def=""
      role_name="" class_const="" rh_name=""
      value="" />
     </CONCEPT>
 - <CONCEPT id="1218901676562_n12">
   <LABEL>Identification Subsystem</LABEL>
   <SUB_LABELS />
   <POS x="488" y="216" />
   <SUB_TREE>open</SUB_TREE>
   <RELATIONS />
   <SUB_L label="Hybrid Subsystem Ontology"
      def="" role_name="" class_const=""
      rh_name="" value="" />
     </CONCEPT>
 - <CONCEPT id="1218901676562_n13">
   <LABEL>Hybrid Identification</LABEL>
   <SUB_LABELS />
   <POS x="704" y="212" />
   <SUB_TREE>open</SUB_TREE>
   <RELATIONS />
   <SUB_L label="Hybrid Identification Ontology"
      def="" role_name="" class_const=""
      rh_name="" value="" />
     </CONCEPT>
 - <CONCEPT id="1218901676562_n14">
   <LABEL>Anomaly Identification</LABEL>
   <SUB_LABELS />
   <POS x="860" y="300" />
   <PART_TREE>close</PART_TREE>
 - <SLOTS>
 - <SLOT id="1218904397000_s15" type="NW"
      label="slot" kind="p/o" num="1"
      role="Method L1" class_constraint="Live
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Live Identification" rh_name=""
      value="" />
     </SLOT>
 - <SLOT id="1218904397000_s16" type="NW"
      label="slot2" kind="p/o" num="1"
      role="Method L2" class_constraint="Live
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1218904397000_s17" type="NW"
      label="slot3" kind="p/o" num="1"
      role="Method Ln" class_constraint="Live
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <PART_TREE>close</PART_TREE>
```

```
   <SUB_L label="" def="" role_name=""
      class_const="Live Identification" rh_name=""
      value="" />
     </SLOT>
     </SLOTS>
   <RELATIONS />
   <SUB_L label="Anomaly Ontology" def=""
      role_name="" class_const="" rh_name=""
      value="" />
     </CONCEPT>
 - <CONCEPT id="1218901676562_n15">
   <LABEL>Misuse Identification</LABEL>
   <SUB_LABELS />
   <POS x="568" y="276" />
   <PART_TREE>close</PART_TREE>
   <SUB_TREE>open</SUB_TREE>
   <RELATIONS />
   <SUB_L label="Misuse Ontology" def=""
      role_name="" class_const="" rh_name=""
      value="" />
     </CONCEPT>
 - <CONCEPT id="1218901676562_n16">
   <LABEL>Response Ontology</LABEL>
   <SUB_LABELS />
   <POS x="148" y="288" />
   <PART_TREE>close</PART_TREE>
   <SUB_TREE>open</SUB_TREE>
   <RELATIONS />
   <SUB_L label="Response Ontology" def=""
      role_name="" class_const="" rh_name=""
      value="" />
     </CONCEPT>
 - <CONCEPT id="1218901676562_n17">
   <LABEL>Active Response Ontology</LABEL>
   <SUB_LABELS />
   <POS x="0" y="364" />
   <PART_TREE>close</PART_TREE>
 - <SLOTS>
 - <SLOT id="1218902034796_s18" type="NW"
      label="slot" kind="p/o" num="1"
      role="Method R1"
      class_constraint="Defending/Active
      Response" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Informing/Passive Response"
      rh_name="" value="" />
     </SLOT>
 - <SLOT id="1218902034796_s19" type="NW"
      label="slot2" kind="p/o" num="1"
      role="Method R2"
      class_constraint="Defending/Active
      Response" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Defending/Active Response"
      rh_name="" value="" />
     </SLOT>
 - <SLOT id="1218902034796_s20" type="NW"
      label="slot3" kind="p/o" num="1"
      role="Method Rn"
      class_constraint="Defending/Active
      Response" rh_name="" value="">
```

```
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Defending/Active Response"
      rh_name="" value="" />
    </SLOT>
    </SLOTS>
  <RELATIONS />
  <SUB_L label="Active Response Ontology"
      def="" role_name="" class_const=""
      rh_name="" value="" />
    </CONCEPT>
- <CONCEPT id="1218901676562_n18">
  <LABEL>Passive Response Ontology</LABEL>
  <SUB_LABELS />
  <POS x="288" y="360" />
  <PART_TREE>close</PART_TREE>
- <SLOTS>
- <SLOT id="1218902034796_s22" type="NW"
      label="slot" kind="p/o" num="1"
      role="Method R'1"
      class_constraint="Informing/Passive
      Response" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Informing/Passive Response"
      rh_name="" value="" />
    </SLOT>
- <SLOT id="1218902034796_s23" type="NW"
      label="slot2" kind="p/o" num="1"
      role="Method R'2"
      class_constraint="Informing/Passive
      Response" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Informing/Passive Response"
      rh_name="" value="" />
    </SLOT>
- <SLOT id="1218902034796_s24" type="NW"
      label="slot3" kind="p/o" num="1"
      role="Method R'n"
      class_constraint="Informing/Passive
      Response" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Informing/Passive Response"
      rh_name="" value="" />
    </SLOT>
    </SLOTS>
  <RELATIONS />
  <SUB_L label="Passive Response Ontology"
      def="" role_name="" class_const=""
      rh_name="" value="" />
    </CONCEPT>
- <CONCEPT id="1218901676562_n19">
  <LABEL>Admin/Manager Console
      Ontology</LABEL>
  <SUB_LABELS />
  <POS x="0" y="8" />
  <PART_TREE>close</PART_TREE>
- <SLOTS>
- <SLOT id="1218902034796_s26" type="NW"
      label="slot" kind="p/o" num="1"
      role="Monitoring Security Policy based on
      Anomaly Identification"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot" def="" role_name="Security
      Policy" class_const="Anomaly Ontology"
      rh_name="" value="" />
    </SLOT>
- <SLOT id="1218902034796_s27" type="NW"
      label="slot2" kind="p/o" num="1"
      role="Updating RSDB"
      class_constraint="RSDB Database"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="RSDB Database" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1218902034796_s28" type="NW"
      label="slot3" kind="p/o" num="1"
      role="Manual Prevention"
      class_constraint="Admin Console" rh_name=""
      value="">
  <SUB_LABELS />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="" def="" role_name=""
      class_const="Admin Console" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1218902034796_s29" type="NW"
      label="slot4" kind="p/o" num="1"
      role="Network Monitoring"
      class_constraint="Admin Console" rh_name=""
      value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Admin Console" rh_name=""
      value="" />
    </SLOT>
    </SLOTS>
  <RELATIONS />
  <SUB_L label="Admin/Manager Console
      Ontology" def="" role_name="" class_const=""
      rh_name="" value="" />
    </CONCEPT>
- <CONCEPT id="1218904047109_n30">
  <LABEL>RSDB Database</LABEL>
  <SUB_LABELS />
  <POS x="588" y="360" />
  <PART_TREE>close</PART_TREE>
- <SLOTS>
- <SLOT id="1218904397000_s34" type="NW"
      label="slot" kind="p/o" num="1"
      role="Method A1"
      class_constraint="Identification Database"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Identification Database"
      rh_name="" value="" />
    </SLOT>
- <SLOT id="1218904397000_s35" type="NW"
      label="slot2" kind="p/o" num="1"
      role="Method A2"
      class_constraint="Identification Database"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1218904397000_s36" type="NW"
      label="slot3" kind="p/o" num="1"
      role="Method An"
      class_constraint="Identification Database"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
     </SLOTS>
   <RELATIONS />
     </CONCEPT>
 - <CONCEPT id="1218931988812_n37">
   <LABEL>Identification Database</LABEL>
   <SUB_LABELS />
   <POS x="44" y="624" />
   <PART_TREE>close</PART_TREE>
 - <SLOTS>
 - <SLOT id="1218931988812_s38" type="NW"
      label="slot" kind="p/o" num="1"
      role="Decoding based (layer2
      Identifications)" class_constraint="Misuse
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <POS x="92" y="520" />
 - <SLOTS>
 - <SLOT id="1218931988812_s39" type="R2-1"
      label="slot" kind="p/o" num="1" role="ARP
      decoding" class_constraint="Misuse
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Rule Set Data Base" rh_name=""
      value="" />
     </SLOT>
 - <SLOT id="1218931988812_s40" type="R2-1"
      label="slot2" kind="p/o" num="1"
      role="Layer 2 header decoding"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
     </SLOT>
     </SLOTS>
   <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
     </SLOT>
 - <SLOT id="1218931988812_s41" type="NW"
      label="slot2" kind="p/o" num="1"
      role="Protocol based (layer3
      Identifications)" class_constraint="Misuse
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <POS x="80" y="692" />
 - <SLOTS>
 - <SLOT id="1218931988812_s42" type="R2-1"
      label="slot" kind="a/o" num="1"
      role="Protocols Snort analyses: tcp, udp,
      icmp, and ip" class_constraint="Misuse
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1218931988812_s43" type="R2-1"
      label="slot2" kind="a/o" num="1" role="test
      the IP header's TTL field value"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1218931988812_s44" type="R2-1"
      label="slot3" kind="a/o" num="1" role="test
      the IP header's TOS field value"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
     </SLOT>
 - <SLOT id="1218931988812_s45" type="R2-1"
      label="slot4" kind="a/o" num="1" role="test
      the IP header's fragment ID field for a
      specific value" class_constraint="Misuse
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1218931988812_s46" type="R2-1"
      label="slot5" kind="a/o" num="1"
      role="watch the IP option fields for specific
      codes" class_constraint="Misuse
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1218931988812_s47" type="R2-1"
      label="slot6" kind="a/o" num="1" role="test
      the fragmentation bits of the IP header"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
 - <SLOT id="1218931988812_s48" type="R2-1"
      label="slot7" kind="a/o" num="1" role="test
      the packet's payload size against a value"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
```

```
- <SLOT id="1218931988812_s49" type="R2-1"
      label="slot8" kind="a/o" num="1" role="test
      the TCP flags for certain values"
      class_constraint="Misuse Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s50" type="R2-1"
      label="slot9" kind="a/o" num="1" role="test
      the TCP sequence number field for a
      specific value" class_constraint="Misuse
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s51" type="R2-1"
      label="slot10" kind="a/o" num="1" role="test
      the TCP acknowledgement field for a
      specific value" class_constraint="Misuse
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1218931988812_s52" type="R2-1"
      label="slot11" kind="a/o" num="1" role="test
      the ICMP type field against a specific value"
      class_constraint="Misuse Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1218931988812_s53" type="R2-1"
      label="slot12" kind="a/o" num="1" role="test
      the ICMP code field against a specific
      value" class_constraint="Misuse
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s54" type="R2-1"
      label="slot13" kind="a/o" num="1" role="test
      the ICMP ECHO ID field against a specific
      value" class_constraint="Misuse
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s55" type="R2-1"
      label="slot14" kind="a/o" num="1" role="test
      the ICMP ECHO sequence number against a
      specific value" class_constraint="Misuse
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1218931988812_s56" type="R2-1"
      label="slot15" kind="a/o" num="1"
      role="filter for SYN/FIN/RST TCP packets"
      class_constraint="Misuse Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s57" type="R2-1"
      label="slot16" kind="a/o" num="1"
      role="process TCP fragments"
      class_constraint="Misuse Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s58" type="R2-1"
      label="slot17" kind="a/o" num="1" role="flag
      HTTP-based worm sources such as Code
      Red" class_constraint="Misuse Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1218931988812_s59" type="R2-1"
      label="slot18" kind="a/o" num="1"
      role="detailed analysis of http requests"
      class_constraint="Misuse Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1218931988812_s60" type="R2-1"
      label="slot19" kind="a/o" num="1"
      role="detailed analysis of http replies"
      class_constraint="Misuse Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
    </SLOTS>
  <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230837580437_s62" type="NW"
      label="slot3" kind="p/o" num="1"
      role="Frame structure based (layer 2
      Identifications)" class_constraint="Misuse
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <POS x="640" y="664" />
- <SLOTS>
- <SLOT id="1230850840750_s64" type="R2-1"
      label="slot" kind="a/o" num="1" role="tests
      the entire frame control field"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230850840750_s65" type="R2-1"
      label="slot2" kind="a/o" num="1" role="tests
      the 802.11 frame's type"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230850840750_s66" type="R2-1"
      label="slot3" kind="a/o" num="1" role="tests
      the 802.11 frame's subtype"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230850840750_s67" type="R2-1"
      label="slot4" kind="a/o" num="1" role="tests
      the from distribution system frame control
      flag" class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
     </SLOT>
- <SLOT id="1230850840750_s68" type="R2-1"
      label="slot5" kind="a/o" num="1" role="tests
      the to distribution system frame control
      flag" class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
     </SLOT>
- <SLOT id="1230850840750_s69" type="R2-1"
      label="slot6" kind="a/o" num="1" role="tests
      the more fragments frame control flag"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230850840750_s70" type="R2-1"
      label="slot7" kind="a/o" num="1" role="tests
      the retry frame control flag"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230850840750_s71" type="R2-1"
      label="slot8" kind="a/o" num="1" role="tests
      the power management frame control flag"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230850840750_s72" type="R2-1"
      label="slot9" kind="a/o" num="1" role="tests
      the more data frame control flag"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230850840750_s73" type="R2-1"
      label="slot10" kind="a/o" num="1"
      role="tests the wep frame control flag"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230850840750_s74" type="R2-1"
      label="slot11" kind="a/o" num="1"
      role="tests the order frame control flag"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230850840750_s75" type="R2-1"
      label="slot12" kind="a/o" num="1"
      role="tests the frame's duration/id field"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
     </SLOT>
- <SLOT id="1230850840750_s76" type="R2-1"
      label="slot13" kind="a/o" num="1"
      role="tests the frame's BSSID"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230850840750_s77" type="R2-1"
      label="slot14" kind="a/o" num="1"
      role="tests the frame's sequence number"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
```

```
    </SLOT>
- <SLOT id="1230850840750_s78" type="R2-1"
      label="slot15" kind="a/o" num="1"
      role="tests the frame's fragment number"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230850840750_s79" type="R2-1"
      label="slot16" kind="a/o" num="1"
      role="tests the frame's 4th address field"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
    </SLOTS>
   <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230852626531_s78" type="NW"
      label="slot4" kind="p/o" num="1"
      role="Other unclassified Identifications"
      class_constraint="Misuse Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <POS x="112" y="1404" />
- <SLOTS>
- <SLOT id="1230852626531_s82" type="R2-1"
      label="slot4" kind="a/o" num="1" role="role"
      class_constraint="Any" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
    </SLOTS>
   <SUB_L label="" def="" role_name=""
      class_const="Misuse Ontology" rh_name=""
      value="" />
    </SLOT>
    </SLOTS>
    </CONCEPT>
- <CONCEPT id="1230852626531_n83">
   <LABEL>Live Identification</LABEL>
   <SUB_LABELS />
   <POS x="1144" y="608" />
   <PART_TREE>close</PART_TREE>
- <SLOTS>
- <SLOT id="1230852626531_s94" type="NW"
      label="slot2" kind="p/o" num="1"
      role="Protocol based (layer3
      Identifications)" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
- <SLOTS>
- <SLOT id="1230852626531_s95" type="R2-1"
      label="slot" kind="a/o" num="1" role="FTP
      analysis" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230852626531_s96" type="R2-1"
      label="slot2" kind="a/o" num="1"
      role="identify and log TFTP sessions"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230852626531_s97" type="R2-1"
      label="slot3" kind="a/o" num="1" role="ICMP
      analysis" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230852626531_s98" type="R2-1"
      label="slot4" kind="a/o" num="1" role="DNS
      analysis" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230852626531_s99" type="R2-1"
      label="slot5" kind="a/o" num="1"
      role="search for a set of patterns in the
      packet's payload" class_constraint="Anomaly
      Identification" rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
    </SLOTS>
   <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s100" type="NW"
      label="slot3" kind="p/o" num="1"
      role="Logging based"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <POS x="684" y="512" />
- <SLOTS>
- <SLOT id="1230855124031_s101" type="R2-1"
      label="slot" kind="a/o" num="1"
      role="detects password scans"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
   <SUB_LABELS />
   <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230855124031_s102" type="R2-1"
      label="slot2" kind="a/o" num="1"
    role="analysing rlogin/telnet"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230855124031_s103" type="R2-1"
    label="slot3" kind="a/o" num="1"
    role="access and record connection events"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
   </SLOTS>
  <SUB_L label="" def="" role_name=""
    class_const="Anomaly Ontology" rh_name=""
    value="" />
   </SLOT>
- <SLOT id="1230855124031_s104" type="NW"
    label="slot4" kind="a/o" num="1"
    role="Frame structure based (layer2
    Identification)" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1188" y="908" />
- <SLOTS>
- <SLOT id="1230855124031_s105" type="R2-1"
    label="slot" kind="a/o" num="1"
    role="Networks with known WEP keys can
    be decrypted in realtime"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230855124031_s106" type="R2-1"
    label="slot2" kind="a/o" num="1"
    role="tracks probe requests and responses"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230855124031_s107" type="R2-1"
    label="slot3" kind="p/o" num="1"
    role="Fuzzy encryption detection(UIHRF)"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Anomaly Ontology" rh_name=""
    value="" />
   </SLOT>
- <SLOT id="1230855124031_s108" type="R2-1"
    label="slot4" kind="p/o" num="1"
    role="BSSID (MAC address) of the
    network(UIHRF)" class_constraint="IHMD
    Database" rh_name="" value="">
  <SUB_LABELS />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
   </SLOT>
- <SLOT id="1230855124031_s109" type="R2-1"
    label="slot5" kind="a/o" num="1"
    role="Display Last-advertised channel for
    network" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Anomaly Ontology" rh_name=""
    value="" />
   </SLOT>
- <SLOT id="1230855124031_s110" type="R2-1"
    label="slot6" kind="a/o" num="1"
    role="Display Number of clients (unique
    MACs) seen on network"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230855124031_s111" type="R2-1"
    label="slot7" kind="a/o" num="1"
    role="Display Network status flags"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230855124031_s112" type="R2-1"
    label="slot8" kind="a/o" num="1"
    role="Display Extra AP information included
    by some manufacturers"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230855124031_s113" type="R2-1"
    label="slot9" kind="a/o" num="1"
    role="Display Number of LLC packets"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230855124031_s114" type="R2-1"
    label="slot10" kind="a/o" num="1"
    role="Identifies Manufacturer and Model pf
    AP and Client" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
   </SLOT>
- <SLOT id="1230855124031_s115" type="R2-1"
    label="slot11" kind="a/o" num="1"
    role="Display Maximum supported rate as
    advertised by AP" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
  </SLOT>
- <SLOT id="1230855124031_s116" type="R2-1"
    label="slot12" kind="a/o" num="1"
    role="Display Maximum supported rate as
    advertised by AP" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
  </SLOT>
- <SLOT id="1230855124031_s117" type="R2-1"
    label="slot13" kind="a/o" num="1"
    role="Display Name of the network or
    group" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
  </SLOT>
- <SLOT id="1230855124031_s118" type="R2-1"
    label="slot14" kind="a/o" num="1"
    role="Display Last seen noise level"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Any" rh_name="" value="" />
  </SLOT>
- <SLOT id="1230855124031_s119" type="R2-1"
    label="slot15" kind="a/o" num="1"
    role="Display Last seen signal level"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Anomaly Ontology" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s120" type="R2-1"
    label="slot16" kind="a/o" num="1"
    role="Display Network type (Probe, Adhoc,
    Infra, etc)" class_constraint="Anomaly
    Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Anomaly Ontology" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s121" type="R2-1"
    label="slot17" kind="a/o" num="1"
    role="Display WEP status"
    class_constraint="Anomaly Identification"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="Anomaly Ontology" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s122" type="R2-1"
    label="slot18" kind="p/o" num="1"
    role="Identifies unusual probes (UIHRF)"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s123" type="R2-1"
    label="slot19" kind="p/o" num="1"
    role="Identifies unusual probes (UIHRF)"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1612" />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s124" type="R2-1"
    label="slot20" kind="p/o" num="1"
    role="Identifies excessive
    disassociation(UIHRF)"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <POS x="1232" y="1640" />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s125" type="R2-1"
    label="slot21" kind="p/o" num="1"
    role="Identifies Netstumbler probe
    requests(UIHRF)" class_constraint="IHMD
    Database" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1232" y="1672" />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s126" type="R2-1"
    label="slot22" kind="p/o" num="1"
    role="Identifies
    Deauthenticate/Disassociate
    Flood(UIHRF)" class_constraint="IHMD
    Database" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1712" />
  <SUB_L label="" def="" role_name=""
    class_const="IHMD Database" rh_name=""
    value="" />
  </SLOT>
- <SLOT id="1230855124031_s127" type="R2-1"
    label="slot23" kind="p/o" num="1"
    role="Identifies Lucent link test(UIHRF)"
    class_constraint="IHMD Database"
    rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1744" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230855124031_s128" type="R2-1"
     label="slot24" kind="p/o" num="1"
     role="Identifies SSID brute force
     attempt(UIHRF)" class_constraint="IHMD
     Database" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1236" y="1816" />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230855124031_s129" type="R2-1"
     label="slot25" kind="p/o" num="1"
     role="Identifies AP changing to a new
     channel(UIHRF)" class_constraint="IHMD
     Database" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1848" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230855124031_s130" type="R2-1"
     label="slot26" kind="p/o" num="1"
     role="Identifies Broadcast
     disconnect/deauthenticate(UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1884" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230855124031_s131" type="R2-1"
     label="slot27" kind="p/o" num="1"
     role="Identifies Invalid BSS
     timestamps(UIHRF)" class_constraint="IHMD
     Database" rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1920" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230855124031_s132" type="R2-1"
     label="slot28" kind="p/o" num="1"
     role="Identifies weak-iv packet (UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="1248" y="1952" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230855124031_s133" type="R2-1"
     label="slot29" kind="p/o" num="1"
     role="Identifies MITM attacks(UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1780" />
  <PART_TREE>close</PART_TREE>
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230855124031_s134" type="R2-1"
     label="slot30" kind="p/o" num="1"
     role="Identifies MAC spoofing attacks
     (UIHRF)" class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="1240" y="1580" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
    </SLOTS>
  <SUB_L label="" def="" role_name=""
     class_const="Anomaly Ontology" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230855124031_s135" type="NW"
     label="slot5" kind="p/o" num="1"
     role="Other unclassified Identifications"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="772" y="1704" />
- <SLOTS>
- <SLOT id="1230855124031_s136" type="R2-1"
     label="slot" kind="p/o" num="1"
     role="Identifies vulnerability scans(UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="816" y="1768" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230855124031_s138" type="R2-1"
     label="slot3" kind="p/o" num="1"
     role="Identifies port scans(UIHRF)"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="812" y="1804" />
  <SUB_L label="" def="" role_name=""
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230855124031_s139" type="R2-1"
     label="slot4" kind="a/o" num="1"
     role="watch & record RPC services for
     specific application/procedure calls"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <POS x="248" y="1928" />
  <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s140" type="R2-1"
      label="slot5" kind="a/o" num="1"
      role="record and analyse email traffic"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <POS x="812" y="1860" />
  <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s141" type="R2-1"
      label="slot6" kind="a/o" num="1" role="track
      software versions"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <POS x="820" y="1900" />
  <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s142" type="R2-1"
      label="slot7" kind="p/o" num="1"
      role="Identifies for blaster worm(UIHRF)"
      class_constraint="IHMD Database"
      rh_name="" value="">
  <SUB_LABELS />
  <POS x="812" y="1940" />
  <SUB_L label="" def="" role_name=""
      class_const="IHMD Database" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s143" type="R2-1"
      label="slot8" kind="p/o" num="1"
      role="Identifies synflood attacks(UIHRF)"
      class_constraint="IHMD Database"
      rh_name="" value="">
  <SUB_LABELS />
  <POS x="320" y="1964" />
  <SUB_L label="" def="" role_name=""
      class_const="IHMD Database" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s144" type="R2-1"
      label="slot9" kind="a/o" num="1" role="SSL
      analyser" class_constraint="Anomaly
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <POS x="324" y="1728" />
  <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s145" type="R2-1"
      label="slot10" kind="p/o" num="1"
      role="Identifies incoming and outgoing
      connections that are ssh(UIHRF)"
      class_constraint="IHMD Database"
      rh_name="" value="">
  <SUB_LABELS />
  <POS x="264" y="1760" />
  <SUB_L label="" def="" role_name=""
      class_const="IHMD Database" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s146" type="R2-1"
      label="slot11" kind="p/o" num="1"
      role="Identifies backdoors(UIHRF)"
      class_constraint="IHMD Database"
      rh_name="" value="">
  <SUB_LABELS />
  <POS x="324" y="1788" />
  <SUB_L label="" def="" role_name=""
      class_const="IHMD Database" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230855124031_s147" type="R2-1"
      label="slot12" kind="a/o" num="1"
      role="Identifies clear text passwords"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <POS x="324" y="1820" />
  <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230858698968_s148" type="R2-1"
      label="slot13" kind="a/o" num="1"
      role="Promiscuous mode"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <POS x="320" y="1852" />
  <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230858698968_s149" type="R2-1"
      label="slot14" kind="a/o" num="1"
      role="search for a pattern in the URI
      portion of a packet"
      class_constraint="Anomaly Identification"
      rh_name="" value="">
  <SUB_LABELS />
  <POS x="320" y="1888" />
  <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
    </SLOTS>
  <SUB_L label="" def="" role_name=""
      class_const="Anomaly Ontology" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230859515843_s137" type="NW"
      label="slot" kind="p/o" num="1" role="Alert
      based" class_constraint="Anomaly
      Identification" rh_name="" value="">
  <SUB_LABELS />
  <POS x="600" y="1320" />
- <SLOTS>
- <SLOT id="1230859515843_s138" type="R2-1"
     label="slot" kind="a/o" num="1" role="send
     alerts to syslog" class_constraint="Anomaly
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot" def="" role_name="send
     alerts to syslog" class_const="Any"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s139" type="R2-1"
     label="slot2" kind="p/o" num="1" role="write
     the alert in a UIHRF format"
     class_constraint="IHMD Database"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot2" def="" role_name="write
     the alert in a simple format"
     class_const="IHMD Database" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230859515843_s140" type="R2-1"
     label="slot3" kind="a/o" num="1" role="send
     alerts to a *NIX socket that another
     program can listen on"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot3" def="" role_name="send
     alerts to a *NIX socket that another
     program can listen on" class_const="Any"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s141" type="R2-1"
     label="slot4" kind="a/o" num="1"
     role="Packets logged to a decoded ASCII
     format or a binary log file"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot4" def="" role_name="Packets
     logged to a decoded ASCII format or a
     binary log file" class_const="Any"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s142" type="R2-1"
     label="slot5" kind="a/o" num="1" role="send
     WinPopup alert messages to Windows
     machines" class_constraint="Anomaly
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot5" def="" role_name="send
     WinPopup alert messages to Windows
     machines" class_const="Any" rh_name=""
     value="" />
    </SLOT>
- <SLOT id="1230859515843_s143" type="R2-1"
     label="slot6" kind="a/o" num="1"
     role="record log in a database"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot6" def="" role_name="record
     log in a database" class_const="Any"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s144" type="R2-1"
     label="slot7" kind="a/o" num="1"
     role="record log in a cvs file"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot7" def="" role_name="record
     log in a cvs file" class_const="Any"
     rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s145" type="R2-1"
     label="slot8" kind="a/o" num="1"
     role="sending alert to administrator's email
     or pager" class_constraint="Anomaly
     Identification" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot8" def="" role_name="sending
     alert to administrator's email or pager"
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s146" type="R2-1"
     label="slot9" kind="a/o" num="1" role="log
     alerts to event logs"
     class_constraint="Anomaly Identification"
     rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot9" def="" role_name="log
     alerts to event logs" class_const="Anomaly
     Ontology" rh_name="" value="" />
    </SLOT>
    </SLOTS>
  <SUB_L label="slot" def="" role_name="Alert
     based" class_const="Anomaly Ontology"
     rh_name="" value="" />
    </SLOT>
    </SLOTS>
    </CONCEPT>
- <CONCEPT id="1230859515843_n147">
  <LABEL>Informing/Passive Response</LABEL>
  <SUB_LABELS />
  <POS x="0" y="1520" />
  <PART_TREE>close</PART_TREE>
- <SLOTS>
- <SLOT id="1230859515843_s148" type="NW"
     label="slot" kind="a/o" num="1"
     role="Terminate a connection"
     class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s149" type="NW"
     label="slot2" kind="a/o" num="1"
     role="Block a malicious host"
     class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
     class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s150" type="NW"
     label="slot3" kind="a/o" num="1" role="Edit
     router's ACL to block hostile hosts"
     class_constraint="Any" rh_name="" value="">
```

```
<SUB_LABELS />
<SUB_L label="" def="" role_name=""
  class_const="Any" rh_name="" value="" />
  </SLOT>
- <SLOT id="1230859515843_s151" type="NW"
  label="slot4" kind="a/o" num="1"
  role="Passive monitoring of network traffic"
  class_constraint="Any" rh_name="" value="">
<SUB_LABELS />
<SUB_L label="" def="" role_name=""
  class_const="Any" rh_name="" value="" />
  </SLOT>
  </SLOTS>
  </CONCEPT>
- <CONCEPT id="1230859515843_n152">
<LABEL>Defending/Active Response</LABEL>
<SUB_LABELS />
<POS x="1172" y="0" />
<PART_TREE>close</PART_TREE>
- <SLOTS>
- <SLOT id="1230859515843_s153" type="NW"
  label="slot" kind="p/o" num="1" role="Alert
  based" class_constraint="Anomaly
  Identification" rh_name="" value="">
<SUB_LABELS />
- <SLOTS>
- <SLOT id="1230859515843_s154" type="R2-2"
  label="slot" kind="a/o" num="1" role="send
  alerts to syslog" class_constraint="Anomaly
  Identification" rh_name="" value="">
<SUB_LABELS />
<PART_TREE>close</PART_TREE>
<SUB_L label="slot" def="" role_name="send
  alerts to syslog"
  class_const="Informing/Passive Response"
  rh_name="" value="" />
  </SLOT>
- <SLOT id="1230859515843_s155" type="R2-2"
  label="slot2" kind="p/o" num="1" role="write
  the alert in a simple format(UIHRF)"
  class_constraint="IHMD Database"
  rh_name="" value="">
<SUB_LABELS />
<SUB_L label="slot2" def="" role_name="write
  the alert in a simple format"
  class_const="Anomaly Ontology" rh_name=""
  value="" />
  </SLOT>
- <SLOT id="1230859515843_s156" type="R2-2"
  label="slot3" kind="a/o" num="1" role="send
  alerts to a *NIX socket that another
  program can listen on"
  class_constraint="Anomaly Identification"
  rh_name="" value="">
<SUB_LABELS />
<SUB_L label="slot3" def="" role_name="send
  alerts to a *NIX socket that another
  program can listen on"
  class_const="Informing/Passive Response"
  rh_name="" value="" />
  </SLOT>
- <SLOT id="1230859515843_s157" type="R2-2"
  label="slot4" kind="a/o" num="1"
  role="Packets logged to a decoded ASCII
  format or a binary log file"
  class_constraint="Anomaly Identification"
  rh_name="" value="">
<SUB_LABELS />
<SUB_L label="slot4" def="" role_name="Packets
  logged to a decoded ASCII format or a
  binary log file"
  class_const="Informing/Passive Response"
  rh_name="" value="" />
  </SLOT>
- <SLOT id="1230859515843_s158" type="R2-2"
  label="slot5" kind="a/o" num="1" role="send
  WinPopup alert messages to Windows
  machines" class_constraint="Anomaly
  Identification" rh_name="" value="">
<SUB_LABELS />
<SUB_L label="slot5" def="" role_name="send
  WinPopup alert messages to Windows
  machines" class_const="Informing/Passive
  Response" rh_name="" value="" />
  </SLOT>
- <SLOT id="1230859515843_s159" type="R2-1"
  label="slot6" kind="a/o" num="1"
  role="record log in a database"
  class_constraint="Anomaly Identification"
  rh_name="" value="">
<SUB_LABELS />
<SUB_L label="slot6" def="" role_name="record
  log in a database"
  class_const="Informing/Passive Response"
  rh_name="" value="" />
  </SLOT>
- <SLOT id="1230859515843_s160" type="R2-1"
  label="slot7" kind="a/o" num="1"
  role="record log in a cvs file"
  class_constraint="Anomaly Identification"
  rh_name="" value="">
<SUB_LABELS />
<SUB_L label="slot7" def="" role_name="record
  log in a cvs file"
  class_const="Informing/Passive Response"
  rh_name="" value="" />
  </SLOT>
- <SLOT id="1230859515843_s161" type="R2-1"
  label="slot8" kind="a/o" num="1"
  role="sending alert to administrator's email
  or pager" class_constraint="Anomaly
  Identification" rh_name="" value="">
<SUB_LABELS />
<SUB_L label="slot8" def="" role_name="sending
  alert to administrator's email or pager"
  class_const="Informing/Passive Response"
  rh_name="" value="" />
  </SLOT>
- <SLOT id="1230859515843_s162" type="R2-1"
  label="slot9" kind="a/o" num="1" role="log
  alerts to event logs"
  class_constraint="Anomaly Identification"
  rh_name="" value="">
<SUB_LABELS />
<SUB_L label="slot9" def="" role_name="log
  alerts to event logs"
  class_const="Informing/Passive Response"
  rh_name="" value="" />
```

```
    </SLOT>
    </SLOTS>
  <SUB_L label="slot" def="" role_name="Alert
      based" class_const="Anomaly Ontology"
      rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s163" type="NW"
      label="slot2" kind="a/o" num="1"
      role="Terminate a connection"
      class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot" def=""
      role_name="Terminate a connection"
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s164" type="NW"
      label="slot22" kind="a/o" num="1"
      role="Block a malicious host"
      class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot2" def="" role_name="Block a
      malicious host" class_const="Any"
      rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s165" type="NW"
      label="slot3" kind="a/o" num="1" role="Edit
      router's ACL to block hostile hosts"
      class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot3" def="" role_name="Edit
      router's ACL to block hostile hosts"
      class_const="Any" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230859515843_s166" type="NW"
      label="slot4" kind="a/o" num="1"
      role="dumps the application layer
      information for a given session"
      class_constraint="Any" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="slot4" def="" role_name="dumps
      the application layer information for a
      given session" class_const="Any" rh_name=""
      value="" />
    </SLOT>
- <SLOT id="1230859515843_s167" type="NW"
      label="slot5" kind="a/o" num="1" role="block
      web sites" class_constraint="Any" rh_name=""
      value="">
  <SUB_LABELS />
  <SUB_L label="slot5" def="" role_name="block
      web sites" class_const="Any" rh_name=""
      value="" />
    </SLOT>
    </SLOTS>
  <SUB_L label="Informing/Passive Response"
      def="" role_name="" class_const=""
      rh_name="" value="" />
    </CONCEPT>
- <CONCEPT id="1230938127140_n167">
  <LABEL>Admin Console</LABEL>
  <SUB_LABELS />
  <POS x="500" y="1380" />
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
    </CONCEPT>
- <CONCEPT id="1230938127140_n168">
  <LABEL>Defending/Active
      Response_2</LABEL>
  <SUB_LABELS />
  <POS x="376" y="1460" />
  <PART_TREE>close</PART_TREE>
  <RELATIONS />
    </CONCEPT>
- <CONCEPT id="1230946789609_n165">
  <LABEL>IHMD Database</LABEL>
  <SUB_LABELS />
  <POS x="296" y="1516" />
  <PART_TREE>close</PART_TREE>
    </CONCEPT>
- <CONCEPT id="1230946789609_n166">
  <LABEL>Intrusion Unified Report
      Format</LABEL>
  <SUB_LABELS />
  <POS x="300" y="1552" />
    </CONCEPT>
- <CONCEPT id="1230946789609_n167">
  <LABEL>IICS Protocols</LABEL>
  <SUB_LABELS />
  <POS x="308" y="1592" />
  <PART_TREE>close</PART_TREE>
  <SUB_TREE>open</SUB_TREE>
- <SLOTS>
- <SLOT id="1230946789609_s168" type="NW"
      label="slot" kind="p/o" num="1" role="Based
      on IDMEF" class_constraint="IIHSM"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="IIHSM" rh_name="" value="" />
    </SLOT>
- <SLOT id="1230946789609_s169" type="NW"
      label="slot2" kind="p/o" num="1" role="Any
      Wireless compatible exchange format"
      class_constraint="IIHSM" rh_name=""
      value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
    </SLOT>
    </SLOTS>
  <RELATIONS />
    </CONCEPT>
- <CONCEPT id="1230946789609_n170"
      instantiation="true">
  <LABEL>Wireless IDXP</LABEL>
  <SUB_LABELS />
  <POS x="504" y="1596" />
  <RELATIONS />
    </CONCEPT>
- <CONCEPT id="1230946789609_n171">
  <LABEL>IIHSM</LABEL>
  <SUB_LABELS />
  <POS x="908" y="132" />
  <PART_TREE>close</PART_TREE>
- <SLOTS>
- <SLOT id="1230946789609_s172" type="NW"
      label="slot" kind="p/o" num="1"
      role="Method T1" class_constraint="IICS
      Protocols" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="IICS Protocol" rh_name=""
      value="" />
     </SLOT>
- <SLOT id="1230946789609_s173" type="NW"
      label="slot2" kind="p/o" num="1"
      role="Method T2" class_constraint="IICS
      Protocols" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230946789609_s174" type="NW"
      label="slot3" kind="p/o" num="1"
      role="Method Tn" class_constraint="IICS
      Protocols" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
     </SLOTS>
  <RELATIONS />
     </CONCEPT>
- <CONCEPT id="1230990225468_n176">
  <LABEL>Data Capturing</LABEL>
  <SUB_LABELS />
  <POS x="11" y="1783" />
  <SUB_TREE>open</SUB_TREE>
  <RELATIONS />
  <SUB_L label="Data Capturing" def=""
      role_name="" class_const="" rh_name=""
      value="" />
     </CONCEPT>
- <CONCEPT id="1230990225468_n177"
      instantiation="true">
  <LABEL>Packet Sniffers</LABEL>
  <SUB_LABELS />
  <POS x="134" y="1881" />
  <RELATIONS />
     </CONCEPT>
- <CONCEPT id="1230990225468_n178"
      instantiation="true">
  <LABEL>Traffic Generators</LABEL>
  <SUB_LABELS />
  <POS x="169" y="1808" />
  <RELATIONS />
     </CONCEPT>
- <CONCEPT id="1230990225468_n179"
      instantiation="true">
  <LABEL>Network Testers</LABEL>
  <SUB_LABELS />
  <POS x="181" y="1719" />
  <RELATIONS />
     </CONCEPT>
- <CONCEPT id="1230990225468_n180"
      instantiation="true">
  <LABEL>Traffic Analysers</LABEL>
  <SUB_LABELS />
  <POS x="6" y="1884" />
  <RELATIONS />
     </CONCEPT>
- <CONCEPT id="1230990225468_n181"
      instantiation="true">
  <LABEL>Network Monitoring Tools</LABEL>
  <SUB_LABELS />
  <POS x="94" y="1684" />
  <RELATIONS />
     </CONCEPT>
- <CONCEPT id="1230990225468_n182"
      instantiation="true">
  <LABEL>Network IHSs</LABEL>
  <SUB_LABELS />
  <POS x="0" y="1704" />
  <RELATIONS />
     </CONCEPT>
- <CONCEPT id="1230990796046_n183">
  <LABEL>Data Processing</LABEL>
  <SUB_LABELS />
  <POS x="87" y="1968" />
  <PART_TREE>close</PART_TREE>
- <SLOTS>
- <SLOT id="1230990796046_s184" type="NW"
      label="slot" kind="p/o" num="1"
      role="Decoding Raw data link layer packets"
      class_constraint="Data Processing"
      rh_name="Preprocessor" value="">
  <SUB_LABELS />
- <SLOTS>
- <SLOT id="1230990796046_s185" type="R2-1"
      label="slot" kind="a/o" num="1"
      role="Ethernet 802.11*"
      class_constraint="Data Processing"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
- <SLOT id="1230990796046_s186" type="R2-1"
      label="slot2" kind="a/o" num="1"
      role="Token Ring packets"
      class_constraint="Data Processing"
      rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Data Processing" rh_name=""
      value="" />
     </SLOT>
- <SLOT id="1230990796046_s187" type="R2-1"
      label="slot3" kind="a/o" num="1"
      role="Higher layer protocols such as IP,
      TCP, and UDP" class_constraint="Data
      Processing" rh_name="" value="">
  <SUB_LABELS />
  <SUB_L label="" def="" role_name=""
      class_const="Any" rh_name="" value="" />
     </SLOT>
     </SLOTS>
  <SUB_L label="" def="" role_name=""
      class_const="Data Processing" rh_name=""
      value="" />
     </SLOT>
```

```
    </SLOTS>
    </CONCEPT>
  <ISA id="1218901676562_isa20"
     parent="IHSMO" child="Incoming Traffic" />
  <ISA id="1218901676562_isa21"
     parent="Incoming Traffic"
     child="Libpcap/Winpcap" />
  <ISA id="1218901676562_isa22"
     parent="IHSMO" child="IICS" />
  <ISA id="1218901676562_isa24"
     parent="IHSMO" child="Identification
     Subsystem" />
  <ISA id="1218901676562_isa25"
     parent="Identification Subsystem"
     child="Hybrid Identification" />
  <ISA id="1218901676562_isa26"
     parent="Hybrid Identification"
     child="Anomaly Identification" />
  <ISA id="1218901676562_isa27"
     parent="Hybrid Identification" child="Misuse
     Identification" />
  <ISA id="1218901676562_isa28"
     parent="IHSMO" child="Response Ontology"
     />
  <ISA id="1218901676562_isa29"
     parent="Response Ontology" child="Active
     Response Ontology" />
  <ISA id="1218901676562_isa30"
     parent="Response Ontology" child="Passive
     Response Ontology" />
  <ISA id="1218901676562_isa31"
     parent="IHSMO" child="Admin/Manager
     Console Ontology" />
  <ISA id="1218904047109_isa43"
     parent="Misuse Identification" child="RSDB
     Database" />
  <ISA id="1230938127140_isa185"
     parent="Admin Console"
     child="Defending/Active Response_2" />
  <ISA id="1230946789609_isa188" parent="IICS
     Protocols" child="Wireless IDXP" />
  <ISA id="1230946789609_isa189"
     parent="IICS" child="IIHSM" />
  <ISA id="1230990225468_isa198"
     parent="Data Capturing" child="Packet
     Sniffers" />
  <ISA id="1230990225468_isa199"
     parent="Data Capturing" child="Traffic
     Generators" />
  <ISA id="1230990225468_isa200"
     parent="Data Capturing" child="Network
     Testers" />
  <ISA id="1230990225468_isa201"
     parent="Data Capturing" child="Traffic
     Analysers" />
  <ISA id="1230990225468_isa202"
     parent="Data Capturing" child="Network
     Monitoring Tools" />
  <ISA id="1230990225468_isa203"
     parent="Data Capturing" child="Network
     IHSs" />
    </W_CONCEPTS>
- <R_CONCEPTS>
  <CANVAS_SIZE w="2000" h="2000" />
    </R_CONCEPTS>
    </OE_FILE>
```

E.5 Text output of Snort Wireless ontology based on IHSMO reference model

```
(Ontology Name)
Snort_Wireless  with Concepts and Slots_v1.ont.xml

(Wholeness Concept)
- IHSMO
  - Incoming Traffic
    - Libpcap/Winpcap
      p/o 1 *NIX Based Capturing System : Data Capturing
      p/o 1 Win Based Capturing System : Data Capturing
      p/o 1 Preprocessor [Packet Decoding, Filtering System] : Data
Processing
  - Identification Subsystem
    - NO Clear Hybrid Definition in Snort Wireless
      - Anomaly Identification
        p/o 1 Method L1 : Live Identification
        p/o 1 Method L2 : Live Identification
        p/o 1 Method Ln : Live Identification
      - Misuse Identification
        - RSDB Database
          p/o 1 Method A1 : Identification Database
          p/o 1 Method A2 : Identification Database
          p/o 1 Method An : Identification Database
  - Response Ontology
    - Active Response Ontology
      p/o 1 Method R1 : Defending/Active Response
      p/o 1 Method R2 : Defending/Active Response
      p/o 1 Method Rn : Defending/Active Response
    - Passive Response Ontology
      p/o 1 Method R'1 : Informing/Passive Response
      p/o 1 Method R'2 : Informing/Passive Response
      p/o 1 Method R'n : Informing/Passive Response
  - Admin/Manager Console Ontology
    p/o 1 Monitoring Security Policy based on Anomaly Identification :
Anomaly Identification
    p/o 1 Updating RSDB : RSDB Database
    p/o 1 Manual Prevention : Admin Console
    p/o 1 Network Monitoring : Admin Console
- Identification Database
  p/o 1 Decoding based (layer2 Identifications) : Misuse Identification
    p/o 1 ARP decoding : Misuse Identification
    p/o 1 Layer 2 header decoding : Misuse Identification
  p/o 1 Protocol based (layer3 Identifications) : Misuse Identification
    a/o 1 Protocols Snort analyses: tcp, udp, icmp, and ip : Misuse
Identification
    a/o 1 test the IP header's TTL field value : Misuse Identification
    a/o 1 test the IP header's TOS field value : Misuse Identification
    a/o 1 test the IP header's fragment ID field for a specific value :
Misuse Identification
    a/o 1 watch the IP option fields for specific codes : Misuse
Identification
    a/o 1 test the fragmentation bits of the IP header : Misuse
Identification
    a/o 1 test the packet's payload size against a value : Misuse
Identification
```

```
    a/o 1 test the TCP flags for certain values : Misuse Identification
    a/o 1 test the TCP sequence number field for a specific value : Misuse Identification
    a/o 1 test the TCP acknowledgement field for a specific value : Misuse Identification
    a/o 1 test the ICMP type field against a specific value : Misuse Identification
    a/o 1 test the ICMP code field against a specific value : Misuse Identification
    a/o 1 test the ICMP ECHO ID field against a specific value : Misuse Identification
    a/o 1 test the ICMP ECHO sequence number against a specific value : Misuse Identification
    a/o 1 filter for SYN/FIN/RST TCP packets : Misuse Identification
    a/o 1 process TCP fragments : Misuse Identification
    a/o 1 flag HTTP-based worm sources such as Code Red : Misuse Identification
    a/o 1 detailed analysis of http requests : Misuse Identification
    a/o 1 detailed analysis of http replies : Misuse Identification
  p/o 1 Frame structure based (layer 2 Identifications) : Misuse Identification
    a/o 1 tests the entire frame control field : Misuse Identification
    a/o 1 tests the 802.11 frame's type : Misuse Identification
    a/o 1 tests the 802.11 frame's subtype : Misuse Identification
    a/o 1 tests the from distribution system frame control flag : Misuse Identification
    a/o 1 tests the to distribution system frame control flag : Misuse Identification
    a/o 1 tests the more fragments frame control flag : Misuse Identification
    a/o 1 tests the retry frame control flag : Misuse Identification
    a/o 1 tests the power management frame control flag : Misuse Identification
    a/o 1 tests the more data frame control flag : Misuse Identification
    a/o 1 tests the wep frame control flag : Misuse Identification
    a/o 1 tests the order frame control flag : Misuse Identification
    a/o 1 tests the frame's duration/id field : Misuse Identification
    a/o 1 tests the frame's BSSID : Misuse Identification
    a/o 1 tests the frame's sequence number : Misuse Identification
    a/o 1 tests the frame's fragment number : Misuse Identification
    a/o 1 tests the frame's 4th address field : Misuse Identification
  p/o 1 Other unclassified Identifications : Misuse Identification
    a/o 1 role : Any
- Live Identification
  p/o 1 Protocol based (layer3 Identifications) : Anomaly Identification
    a/o 1 ICMP analysis: Anomaly Identification
    a/o 1 search for a set of patterns in the packet's payload : Anomaly Identification
  a/o 1 Frame structure based (layer2 Identification): Anomaly Identification
    a/o 1 Networks with known WEP keys can be decrypted in realtime: Anomaly Identification
    p/o 1 BSSID (MAC address) of the network(UIHRF) : Anomaly Identification
    a/o 1 Display Name of the network or group : Anomaly Identification
    a/o 1 Display Network type (Probe, Adhoc, Infra, etc): Anomaly Identification
  p/o 1 Other unclassified Identifications : Anomaly Identification
    a/o 1 watch & record RPC services for specific application/procedure calls: Anomaly Identification
    a/o 1 Promiscuous mode: Anomaly Identification
    a/o 1 search for a pattern in the URI portion of a packet: Anomaly Identification
  p/o 1 Alert based : Anomaly Identification
    a/o 1 send alerts to syslog: Anomaly Identification
    p/o 1 write the alert in a UIHRF format : IHMD Database
    a/o 1 send alerts to a *NIX socket that another program can listen on : Anomaly Identification
    a/o 1 Packets logged to a decoded ASCII format or a binary log file : Anomaly Identification
    a/o 1 send WinPopup alert messages to Windows machines : Anomaly Identification
    a/o 1 record log in a database: Anomaly Identification
    a/o 1 record log in a cvs file: Anomaly Identification
    a/o 1 log alerts to event logs: Anomaly Identification
- Informing/Passive Response
  a/o 1 Terminate a connection: Any
  a/o 1 Block a malicious host: Any
  a/o 1 Passive monitoring of network traffic: Any
- Defending/Active Response
  p/o 1 Alert based : Anomaly Identification
    a/o 1 send alerts to syslog: Anomaly Identification
    p/o 1 write the alert in a simple format
    a/o 1 send alerts to a *NIX socket that another program can listen on : Anomaly Identification
    a/o 1 Packets logged to a decoded ASCII format or a binary log file: Anomaly Identification
    a/o 1 send WinPopup alert messages to Windows machines: Anomaly Identification
    a/o 1 record log in a database: Anomaly Identification
    a/o 1 record log in a cvs file: Anomaly Identification
    a/o 1 Watch Remote Procedure Call (RPC) protocol services for specific application/procedure calls: Anomaly Identification
    a/o 1 Search for a pattern in the Uniform Resource Indicator (URI) portion of packet: Anomaly Identification
  a/o 1 Terminate a connection: Any
  a/o 1 Block a malicious host: Any
  a/o 1 Edit router's ACL to block hostile hosts: Any
  a/o 1 dumps the application layer information for a given session: Any
  a/o 1 block web sites: Any
- Admin Console
  - Defending/Active Response_2
- Data Capturing
  - Packet Sniffers
  - Traffic Generators
  - Network Testers
  - Traffic Analysers
  - Network Monitoring Tools
  - Network IHSs
- Data Processing
```

```
  p/o 1 Decoding Raw data link layer packets [Preprocessor] : Data
Processing
    a/o 1 Ethernet 802.11*: Data Processing
    a/o 1 Token Ring packets: Data Processing
    a/o 1 Higher layer protocols such as IP, TCP, and UDP: Data
Processing

(Relation Concept)
- equal
  p/i 1 value : number
  p/i 1 value : number
- sameAs
  p/i 1 class : Any
  p/i 1 class : Any
- larger-than
  p/i 1 larger : Any
  p/i 1 smaller : Any

-----------------------------------------------------------------------
```

E.6 Text output of IHSMO reference model and the missing parts of Snort wireless shown in highlight

The highlighted sections show the missing components of snort IHS system when evaluated using IHSMO reference model.

```
   (Ontology Name)
IHSMO with Concepts and Slots_v2.ont.xml

(Wholeness Concept)
- IHSMO
  - Incoming Traffic
    - Libpcap/Winpcap
      p/o 1 *NIX Based Capturing System : Data Capturing
      p/o 1 Win Based Capturing System : Data Capturing
      p/o 1 Preprocessor [Packet Decoding, Filtering System] : Data
Processing
  - IICS
    p/o 1 IHMD : IHMD Database
    p/o 1 UIHRF : Intrusion Unified Report Format
    - IIHSM
      p/o 1 Method T1 : IICS Protocols
      p/o 1 Method T2 : IICS Protocols
      p/o 1 Method Tn : IICS Protocols
  - Identification  Subsystem
    - Hybrid Identification
      - Anomaly Identification
        p/o 1 Method L1 : Live Identification
        p/o 1 Method L2 : Live Identification
        p/o 1 Method Ln : Live Identification
      - Misuse Identification
        - RSDB Database
          p/o 1 Method A1 : Identification Database
          p/o 1 Method A2 : Identification Database
          p/o 1 Method An : Identification Database
  - Response Ontology
    - Active Response Ontology
      p/o 1 Method R1 : Defending/Active Response
      p/o 1 Method R2 : Defending/Active Response
      p/o 1 Method Rn : Defending/Active Response
    - Passive Response Ontology
      p/o 1 Method R'1 : Informing/Passive Response
      p/o 1 Method R'2 : Informing/Passive Response
      p/o 1 Method R'n : Informing/Passive Response
    - Admin/Manager Console Ontology
    p/o 1 Monitoring Security Policy based on Anomaly Identification : Anomaly Identification
    p/o 1 Updating RSDB : RSDB Database
    p/o 1 Manual Prevention : Admin Console
    p/o 1 Network Monitoring : Admin Console
  - Identification Database
  p/o 1 Decoding based (layer2 Identifications) : Misuse Identification
    p/o 1 ARP decoding : Misuse Identification
    p/o 1 Layer 2 header decoding : Misuse Identification
  p/o 1 Protocol based (layer3 Identifications) : Misuse Identification
    a/o 1 Protocols being analysed: tcp, udp, icmp, and ip: Misuse Identification
    a/o 1 test the IP header's TTL field value: Misuse Identification
    a/o 1 test the IP header's TOS field value: Misuse Identification
    a/o 1 test the IP header's fragment ID field for a specific value: Misuse Identification
    a/o 1 watch the IP option fields for specific codes: Misuse Identification
    a/o 1 test the fragmentation bits of the IP header: Misuse Identification
    a/o 1 test the packet's payload size against a value: Misuse Identification
    a/o 1 test the TCP flags for certain values: Misuse Identification
    a/o 1 test the TCP sequence number field for a specific value: Misuse Identification
    a/o 1 test the TCP acknowledgement field for a specific value: Misuse Identification
    a/o 1 test the ICMP type field against a specific value: Misuse Identification
    a/o 1 test the ICMP code field against a specific value: Misuse Identification
    a/o 1 test the ICMP ECHO ID field against a specific value: Misuse Identification
    a/o 1 test the ICMP ECHO sequence number against a specific value: Misuse Identification
    a/o 1 filter for SYN/FIN/RST TCP packets: Misuse Identification
    a/o 1 process TCP fragments: Misuse Identification
    a/o 1 flag HTTP-based worm sources such as Code Red: Misuse Identification
    a/o 1 detailed analysis of http requests: Misuse Identification
    a/o 1 detailed analysis of http replies: Misuse Identification
  p/o 1 Frame structure based (layer 2 Identifications) : Misuse Identification
    a/o 1 tests the entire frame control field: Misuse Identification
    a/o 1 tests the 802.11 frame's type: Misuse Identification
    a/o 1 tests the 802.11 frame's subtype: Misuse Identification
    a/o 1 tests the from distribution system frame control flag: Misuse
Identification
    a/o 1 tests the to distribution system frame control flag: Misuse
Identification
    a/o 1 tests the more fragments frame control flag: Misuse
Identification
    a/o 1 tests the retry frame control flag: Misuse Identification
    a/o 1 tests the power management frame control flag: Misuse
Identification
    a/o 1 tests the more data frame control flag: Misuse Identification
    a/o 1 tests the wep frame control flag: Misuse Identification
    a/o 1 tests the order frame control flag: Misuse Identification
    a/o 1 tests the frame's duration/id field: Misuse Identification
    a/o 1 tests the frame's BSSID: Misuse Identification
    a/o 1 tests the frame's sequence number: Misuse Identification
    a/o 1 tests the frame's fragment number: Misuse Identification
    a/o 1 tests the frame's 4th address field: Misuse Identification
  p/o 1 Other unclassified Identifications : Misuse Identification
    a/o 1 role: Any
- Live Identification
  p/o 1 Protocol based (layer3 Identifications) : Anomaly
Identification
    a/o 1 FTP analysis: Anomaly Identification
    a/o 1 identify and log TFTP sessions : Anomaly Identification
    a/o 1 ICMP analysis: Anomaly Identification
    a/o 1 DNS analysis: Anomaly Identification
    a/o 1 search for a set of patterns in the packet's payload: Anomaly
Identification
  p/o 1 Logging based : Anomaly Identification
    a/o 1 detects password scans: Anomaly Identification
    a/o 1 analysing rlogin/telnet: Anomaly Identification
    a/o 1 access and record connection events : Anomaly Identification
  a/o 1 Frame structure based (layer2 Identification): Anomaly
Identification
    a/o 1 Networks with known WEP keys can be decrypted in realtime:
Anomaly Identification
    a/o 1 tracks probe requests and responses: Anomaly Identification
    p/o 1 Fuzzy encryption detection(UIHRF) : IHMD Database
    p/o 1 BSSID (MAC address) of the network(UIHRF) : IHMD Database
    a/o 1 Display Last-advertised channel for network: Anomaly
Identification
    a/o 1 Display Number of clients (unique MACs) seen on network:
Anomaly Identification
    a/o 1 Display Network status flags: Anomaly Identification
    a/o 1 Display Extra AP information included by some manufacturers:
Anomaly Identification
    a/o 1 Display Number of LLC packets: Anomaly Identification
    a/o 1 Identifies Manufacturer and Model pf AP and Client: Anomaly
Identification
    a/o 1 Display Maximum supported rate as advertised by AP: Anomaly
Identification
    a/o 1 Display Maximum supported rate as advertised by AP: Anomaly
Identification
    a/o 1 Display Name of the network or group : Anomaly Identification
    a/o 1 Display Last seen noise level: Anomaly Identification
    a/o 1 Display Last seen signal level: Anomaly Identification
    a/o 1 Display Network type (Probe, Adhoc, Infra, etc): Anomaly
Identification
    a/o 1 Display WEP status: Anomaly Identification
    p/o 1 Identifies unusual probes (UIHRF) : IHMD Database
    p/o 1 Identifies unusual probes (UIHRF) : IHMD Database
    p/o 1 Identifies excessive disassociation(UIHRF) : IHMD Database
    p/o 1 Identifies Netstumbler probe requests(UIHRF) : IHMD Database
    p/o 1 Identifies Deauthenticate/Disassociate Flood(UIHRF) : IHMD
Database
    p/o 1 Identifies Lucent link test(UIHRF) : IHMD Database
    p/o 1 Identifies SSID brute force attempt(UIHRF) : IHMD Database
    p/o 1 Identifies AP changing to a new channel(UIHRF) : IHMD
Database
    p/o 1 Identifies Broadcast disconnect/deauthenticate(UIHRF) : IHMD
Database
    p/o 1 Identifies Invalid BSS timestamps(UIHRF) : IHMD Database
    p/o 1 Identifies weak-iv packet (UIHRF) : IHMD Database
    p/o 1 Identifies  MITM attacks(UIHRF) : IHMD Database
    p/o 1 Identifies MAC spoofing attacks (UIHRF) : IHMD Database
  p/o 1 Other unclassified Identifications : Anomaly Identification
    p/o 1 Identifies vulnerability scans(UIHRF) : IHMD Database
    p/o 1 Identifies port scans(UIHRF) : IHMD Database
    a/o 1 watch & record RPC services for specific
application/procedure calls : Anomaly Identification
    a/o 1 record and analyse email traffic: Anomaly Identification
    a/o 1 track software versions: Anomaly Identification
    p/o 1 Identifies for blaster worm(UIHRF) : IHMD Database
    p/o 1 Identifies synflood attacks(UIHRF) : IHMD Database
    a/o 1 SSL analyser: Anomaly Identification
    p/o 1 Identifies incoming and outgoing connections that are
ssh(UIHRF) : IHMD Database
    p/o 1 Identifies backdoors(UIHRF) : IHMD Database
    a/o 1 Identifies clear text passwords : Anomaly Identification
    a/o 1 Promiscuous mode: Anomaly Identification
    a/o 1 search for a pattern in the URI portion of a packet: Anomaly
Identification
  p/o 1 Alert based : Anomaly Identification
    a/o 1 send alerts to syslog: Anomaly Identification
    p/o 1 write the alert in a UIHRF format : IHMD Database
    a/o 1 send alerts to a *NIX socket that another program can listen
on: Anomaly Identification
    a/o 1 Packets logged to a decoded ASCII format or a binary log
file: Anomaly Identification
    a/o 1 send WinPopup alert messages to Windows machines: Anomaly
Identification
    a/o 1 record log in a database: Anomaly Identification
    a/o 1 record log in a cvs file: Anomaly Identification
    a/o 1 sending alert to administrator's email or pager: Anomaly
Identification
    a/o 1 log alerts to event logs: Anomaly Identification
- Informing/Passive Response
  a/o 1 Terminate a connection: Any
  a/o 1 Block a malicious host : Any
  a/o 1 Edit router's ACL to block hostile hosts: Any
  a/o 1 Passive monitoring of network traffic: Any
- Defending/Active Response
  p/o 1 Alert based : Anomaly Identification
```

```
    a/o 1 send alerts to syslog: Anomaly Identification
    p/o 1 write the alert in a simple format(UIHRF) : IHMD Database
    a/o 1 send alerts to a *NIX socket that another program can listen
on: Anomaly Identification
    a/o 1 Packets logged to a decoded ASCII format or a binary log
file: Anomaly Identification
    a/o 1 send WinPopup alert messages to Windows machines: Anomaly
Identification
    a/o 1 record log in a database: Anomaly Identification
    a/o 1 record log in a cvs file: Anomaly Identification
    a/o 1 sending alert to administrator's email or pager : Anomaly
Identification
    a/o 1 log alerts to event logs : Anomaly Identification
    a/o 1 Watch Remote Procedure Call (RPC) protocol services for
specific application/procedure calls: Anomaly Identification
    a/o 1 Search for a pattern in the Uniform Resource Indicator (URI)
portion of packet: Anomaly Identification
  a/o 1 Terminate a connection: Any
  a/o 1 Block a malicious host: Any
  a/o 1 Edit router's ACL to block hostile hosts: Any
  a/o 1 dumps the application layer information for a given session :
Any
  a/o 1 block web sites: Any
- Admin Console
  - Defending/Active Response_2
- IHMD Database
- Intrusion Unified Report Format
- IICS Protocols
  p/o 1 Based on IDMEF : IIHSM
  p/o 1 Any Wireless  compatible exchange format : IIHSM
  - Wireless IDXP
- Data Capturing
  - Packet Sniffers
  - Traffic Generators
  - Network Testers
  - Traffic Analysers
  - Network Monitoring Tools
  - Network IHSs
- Data Processing
  p/o 1 Decoding Raw data link layer packets [Preprocessor] : Data
Processing
    a/o 1 Ethernet 802.11*: Data Processing
    a/o 1 Token Ring packets: Data Processing
    a/o 1 Higher layer protocols such as IP, TCP, and UDP: Data
Processing

(Relation Concept)
- equal
  p/i 1 value : number
  p/i 1 value : number
- sameAs
  p/i 1 class : Any
  p/i 1 class : Any
- larger-than
  p/i 1 larger : Any
  p/i 1 smaller : Any
```

E.7 Text output of Bro IHS ontology based on IHSMO reference model

```
(Ontology Name)
Bro_IHS with Concepts and Slots_v1.ont.xml

(Wholeness Concept)
- IHSMO
  - Incoming Traffic
    - Libpcap/Winpcap
      p/o 1 *NIX Based Capturing System : Data Capturing
      p/o 1 Kernel [Filtering System] : Data Processing
  - Identification  Subsystem
    - No Hybrid Identification in Bro IHS System
      - Anomaly Identification
        p/o 1 Method L1 : Live Identification
        p/o 1 Method L2 : Live Identification
        p/o 1 Method Ln : Live Identification
      - Misuse Identification
        - RSDB Database
          p/o 1 Method A1 : Identification Database
          p/o 1 Method A2 : Identification Database
          p/o 1 Method An : Identification Database
  - Response Ontology
    - Active Response Ontology
      p/o 1 Method R1 : Defending/Active Response
      p/o 1 Method R2 : Defending/Active Response
      p/o 1 Method Rn : Defending/Active Response
    - Passive Response Ontology
      p/o 1 Method R'1 : Informing/Passive Response
      p/o 1 Method R'2 : Informing/Passive Response
      p/o 1 Method R'n : Informing/Passive Response
  - Admin/Manager Console Ontology
    p/o 1 Monitoring Security Policy based on Anomaly Identification :
Anomaly Identification
    p/o 1 Updating RSDB : RSDB Database
    p/o 1 Manual Prevention : Admin Console
    p/o 1 Network Monitoring : Admin Console
- Identification Database
  p/o 1 Protocol based (layer3 Identifications) : Misuse Identification
    a/o 1 filter for SYN/FIN/RST TCP packets : Misuse Identification
    a/o 1 process TCP fragments : Misuse Identification
    a/o 1 flag HTTP-based worm sources such as Code Red : Misuse
Identification
    a/o 1 detailed analysis of http requests : Misuse Identification
    a/o 1 detailed analysis of http replies : Misuse Identification
- Live Identification
  p/o 1 Protocol based (layer3 Identifications) : Anomaly
Identification
    a/o 1 FTP analysis : Anomaly Identification
    a/o 1 identify and log TFTP sessions : Anomaly Identification
    a/o 1 ICMP analysis : Anomaly Identification
    a/o 1 DNS analysis : Anomaly Identification
  p/o 1 Logging based : Anomaly Identification
```

```
    a/o 1 detects password scans : Anomaly Identification
    a/o 1 analysing rlogin/telnet : Anomaly Identification
    a/o 1 access and record connection events : Anomaly Identification
  p/o 1 Other unclassified Identifications : Anomaly Identification
    p/o 1 Identifies vulnerability scans : Anomaly Identification
    p/o 1 Identifies port scans : Anomaly Identification
    a/o 1 watch & record Remote Procedure Call (RPC) protocol services
for specific application/procedure calls : Anomaly Identification
    a/o 1 record and analyse email traffic : Anomaly Identification
    a/o 1 track software versions : Anomaly Identification
    p/o 1 Identifies for blaster worm : Anomaly Identification
    a/o 1 SSL analyser : Anomaly Identification
    p/o 1 Identifies incoming and outgoing connections that are ssh :
Anomaly Identification
    p/o 1 Identifies backdoors : Anomaly Identification
    a/o 1 Identifies clear text passwords : Anomaly Identification
  p/o 1 Alert based : Anomaly Identification
    a/o 1 send alerts to syslog : Anomaly Identification
    a/o 1 record log in a database : Anomaly Identification
    a/o 1 sending alert to administrator's email or pager : Anomaly
Identification
    a/o 1 log alerts to event logs : Anomaly Identification
- Informing/Passive Response
  a/o 1 Terminate a connection : Any
  a/o 1 Block a malicious host : Any
  a/o 1 Edit router's ACL to block hostile hosts : Any
  a/o 1 Passive monitoring of network traffic : Any
- Defending/Active Response
  p/o 1 Alert based : Anomaly Identification
    a/o 1 send alerts to syslog : Anomaly Identification
    a/o 1 record log in a database : Anomaly Identification
    a/o 1 sending alert to administrator's email or pager : Anomaly
Identification
    a/o 1 log alerts to event logs : Anomaly Identification
  a/o 1 Terminate a connection : Any
  a/o 1 Block a malicious host : Any
  a/o 1 Edit router's ACL to block hostile hosts : Any
- Admin Console
  - Defending/Active Response_2
- Data Capturing
  - Packet Sniffers
  - Traffic Generators
  - Network Testers
  - Traffic Analysers
  - Network Monitoring Tools
  - Network IHSs

(Relation Concept)
- equal
  p/i 1 value : number
  p/i 1 value : number
- sameAs
  p/i 1 class : Any
  p/i 1 class : Any
- larger-than
  p/i 1 larger : Any
  p/i 1 smaller : Any
```

E.8 Text output of IHSMO reference model and the missing parts of Bro IHS System shown in highlight

The highlighted sections show the missing components of Bro IHS system when evaluated using IHSMO reference model.

```
(Ontology Name)
IHSMO with Concepts and Slots_v2.ont.xml

(Wholeness Concept)
- IHSMO
  - Incoming Traffic
    - Libpcap/Winpcap
      p/o 1 *NIX Based Capturing System : Data Capturing
      p/o 1 Win Based Capturing System : Data Capturing
      p/o  1  Preprocessor  [Packet  Decoding,  Filtering  System]  :  Data
Processing
  - IICS
    p/o 1 IHMD : IHMD Database
    p/o 1 UIHRF : Intrusion Unified Report Format
    - IIHSM
      p/o 1 Method T1 : IICS Protocols
      p/o 1 Method T2 : IICS Protocols
      p/o 1 Method Tn : IICS Protocols
  - Identification  Subsystem
    - Hybrid Identification
      - Anomaly Identification
        p/o 1 Method L1 : Live Identification
        p/o 1 Method L2 : Live Identification
        p/o 1 Method Ln : Live Identification
      - Misuse Identification
        - RSDB Database
          p/o 1 Method A1 : Identification Database
          p/o 1 Method A2 : Identification Database
          p/o 1 Method An : Identification Database
  - Response Ontology
    - Active Response Ontology
      p/o 1 Method R1 : Defending/Active Response
      p/o 1 Method R2 : Defending/Active Response
      p/o 1 Method Rn : Defending/Active Response
    - Passive Response Ontology
      p/o 1 Method R'1 : Informing/Passive Response
      p/o 1 Method R'2 : Informing/Passive Response
      p/o 1 Method R'n : Informing/Passive Response
  - Admin/Manager Console Ontology
    p/o 1 Monitoring Security Policy based on Anomaly Identification :
Anomaly Identification
    p/o 1 Updating RSDB : RSDB Database
    p/o 1 Manual Prevention : Admin Console
    p/o 1 Network Monitoring : Admin Console
- Identification Database
```

```
  p/o 1 Decoding based (layer2 Identifications) : Misuse Identification
    p/o 1 ARP decoding : Misuse Identification
    p/o 1 Layer 2 header decoding : Misuse Identification
  p/o 1 Protocol based (layer3 Identifications) : Misuse Identification
    a/o 1 Protocols being analysed: tcp, udp, icmp, and ip : Misuse
Identification
    a/o 1 test the IP header's TTL field value : Misuse Identification
    a/o 1 test the IP header's TOS field value : Misuse Identification
    a/o 1 test the IP header's fragment ID field for a specific value :
Misuse Identification
    a/o 1 watch the IP option fields for specific codes : Misuse
Identification
    a/o 1 test the fragmentation bits of the IP header : Misuse
Identification
    a/o 1 test the packet's payload size against a value : Misuse
Identification
    a/o 1 test the TCP flags for certain values : Misuse Identification
    a/o 1 test the TCP sequence number field for a specific value :
Misuse Identification
    a/o 1 test the TCP acknowledgement field for a specific value :
Misuse Identification
    a/o 1 test the ICMP type field against a specific value : Misuse
Identification
    a/o 1 test the ICMP code field against a specific value : Misuse
Identification
    a/o 1 test the ICMP ECHO ID field against a specific value : Misuse
Identification
    a/o 1 test the ICMP ECHO sequence number against a specific value :
Misuse Identification
    a/o 1 filter for SYN/FIN/RST TCP packets : Misuse Identification
    a/o 1 process TCP fragments : Misuse Identification
    a/o 1 flag HTTP-based worm sources such as Code Red : Misuse
Identification
    a/o 1 detailed analysis of http requests : Misuse Identification
    a/o 1 detailed analysis of http replies : Misuse Identification
  p/o 1 Frame structure based (layer 2 Identifications) : Misuse
Identification
    a/o 1 tests the entire frame control field : Misuse Identification
    a/o 1 tests the 802.11 frame's type : Misuse Identification
    a/o 1 tests the 802.11 frame's subtype : Misuse Identification
    a/o 1 tests the from distribution system frame control flag :
Misuse Identification
    a/o 1 tests the to distribution system frame control flag : Misuse
Identification
    a/o 1 tests the more fragments frame control flag : Misuse
Identification
    a/o 1 tests the retry frame control flag : Misuse Identification
    a/o 1 tests the power management frame control flag : Misuse
Identification
    a/o 1 tests the more data frame control flag : Misuse
Identification
    a/o 1 tests the wep frame control flag : Misuse Identification
    a/o 1 tests the order frame control flag : Misuse Identification
    a/o 1 tests the frame's duration/id field : Misuse Identification
    a/o 1 tests the frame's BSSID : Misuse Identification
    a/o 1 tests the frame's sequence number : Misuse Identification
    a/o 1 tests the frame's fragment number : Misuse Identification
    a/o 1 tests the frame's 4th address field : Misuse Identification
  p/o 1 Other unclassified Identifications : Misuse Identification
    a/o 1 role : Any
- Live Identification
  p/o 1 Protocol based (layer3 Identifications) : Anomaly
Identification
    a/o 1 FTP analysis : Anomaly Identification
    a/o 1 identify and log TFTP sessions : Anomaly Identification
    a/o 1 ICMP analysis : Anomaly Identification
    a/o 1 DNS analysis : Anomaly Identification
    a/o 1 search for a set of patterns in the packet's payload :
Anomaly Identification
  p/o 1 Logging based : Anomaly Identification
    a/o 1 detects password scans : Anomaly Identification
    a/o 1 analysing rlogin/telnet : Anomaly Identification
    a/o 1 access and record connection events : Anomaly Identification
  a/o 1 Frame structure based (layer2 Identification) : Anomaly
Identification
    a/o 1 Networks with known WEP keys can be decrypted in realtime :
Anomaly Identification
    a/o 1 tracks probe requests and responses : Anomaly Identification
    p/o 1 Fuzzy encryption detection(UIHRF) : IHMD Database
    p/o 1 BSSID (MAC address) of the network(UIHRF) : IHMD Database
    a/o 1 Display Last-advertised channel for network : Anomaly
Identification
    a/o 1 Display Number of clients (unique MACs) seen on network :
Anomaly Identification
    a/o 1 Display Network status flags : Anomaly Identification
    a/o 1 Display Extra AP information included by some manufacturers :
Anomaly Identification
    a/o 1 Display Number of LLC packets : Anomaly Identification
    a/o 1 Identifies Manufacturer and Model pf AP and Client : Anomaly
Identification
    a/o 1 Display Maximum supported rate as advertised by AP : Anomaly
Identification
    a/o 1 Display Maximum supported rate as advertised by AP : Anomaly
Identification
    a/o 1 Display Name of the network or group : Anomaly Identification
    a/o 1 Display Last seen noise level : Anomaly Identification
    a/o 1 Display Last seen signal level : Anomaly Identification
    a/o 1 Display Network type (Probe, Adhoc, Infra, etc) : Anomaly
Identification
    a/o 1 Display WEP status  : Anomaly Identification
    p/o 1 Identifies unusual probes (UIHRF) : IHMD Database
    p/o 1 Identifies unusual probes (UIHRF) : IHMD Database
    p/o 1 Identifies excessive disassociation(UIHRF) : IHMD Database
    p/o 1 Identifies Netstumbler probe requests(UIHRF) : IHMD Database
    p/o 1 Identifies Deauthenticate/Disassociate Flood(UIHRF) : IHMD
Database
    p/o 1 Identifies Lucent link test(UIHRF) : IHMD Database
    p/o 1 Identifies SSID brute force attempt(UIHRF) : IHMD Database
    p/o 1 Identifies AP changing to a new channel(UIHRF) : IHMD
Database
    p/o 1 Identifies Broadcast disconnect/deauthenticate(UIHRF) : IHMD
Database
    p/o 1 Identifies Invalid BSS timestamps(UIHRF) : IHMD Database
    p/o 1 Identifies weak-iv packet (UIHRF) : IHMD Database
    p/o 1 Identifies  MITM attacks(UIHRF) : IHMD Database
    p/o 1 Identifies MAC spoofing attacks (UIHRF) : IHMD Database
  p/o 1 Other unclassified Identifications : Anomaly Identification
    p/o 1 Identifies vulnerability scans(UIHRF) : IHMD Database
    p/o 1 Identifies port scans(UIHRF) : IHMD Database
    a/o 1 watch & record RPC services for specific application/procedure calls : Anomaly Identification
    a/o 1 record and analyse email traffic : Anomaly Identification
    a/o 1 track software versions : Anomaly Identification
    p/o 1 Identifies for blaster worm(UIHRF) : IHMD Database
    p/o 1 Identifies synflood attacks(UIHRF) : IHMD Database
    a/o 1 SSL analyser : Anomaly Identification
    p/o 1 Identifies incoming and outgoing connections that are ssh(UIHRF) : IHMD Database
    p/o 1 Identifies backdoors(UIHRF) : IHMD Database
    a/o 1 Identifies clear text passwords : Anomaly Identification
    a/o 1 Promiscuous mode : Anomaly Identification
    a/o 1 search for a pattern in the URI portion of a packet : Anomaly Identification
  p/o 1 Alert based : Anomaly Identification
    a/o 1 send alerts to syslog : Anomaly Identification
    p/o 1 write the alert in a UIHRF format : IHMD Database
    a/o 1 send alerts to a *NIX socket that another program can listen on : Anomaly Identification
    a/o 1 Packets logged to a decoded ASCII format or a binary log file : Anomaly Identification
    a/o 1 send WinPopup alert messages to Windows machines : Anomaly Identification
    a/o 1 record log in a database : Anomaly Identification
    a/o 1 record log in a cvs file : Anomaly Identification
    a/o 1 sending alert to administrator's email or pager : Anomaly Identification
    a/o 1 log alerts to event logs : Anomaly Identification
- Informing/Passive Response
  a/o 1 Terminate a connection : Any
  a/o 1 Block a malicious host : Any
  a/o 1 Edit router's ACL to block hostile hosts : Any
  a/o 1 Passive monitoring of network traffic : Any
- Defending/Active Response
  p/o 1 Alert based : Anomaly Identification
    a/o 1 send alerts to syslog : Anomaly Identification
    p/o 1 write the alert in a simple format(UIHRF) : IHMD Database
    a/o 1 send alerts to a *NIX socket that another program can listen on : Anomaly Identification
    a/o 1 Packets logged to a decoded ASCII format or a binary log file : Anomaly Identification
    a/o 1 send WinPopup alert messages to Windows machines : Anomaly Identification
    a/o 1 record log in a database : Anomaly Identification
    a/o 1 record log in a cvs file : Anomaly Identification
    a/o 1 sending alert to administrator's email or pager : Anomaly Identification
    a/o 1 log alerts to event logs : Anomaly Identification
    a/o 1 Watch Remote Procedure Call (RPC) protocol services for specific application/procedure calls : Anomaly Identification
    a/o 1 Search for a pattern in the Uniform Resource Indicator (URI) portion of packet : Anomaly Identification
  a/o 1 Terminate a connection : Any
  a/o 1 Block a malicious host : Any
  a/o 1 Edit router's ACL to block hostile hosts : Any
  a/o 1 dumps the application layer information for a given session :
Any
  a/o 1 block web sites : Any
- Admin Console
  - Defending/Active Response_2
- IHMD Database
- Intrusion Unified Report Format
- IICS Protocols
  p/o 1 Based on IDMEF : IIHSM
  p/o 1 Any Wireless  compatible exchange format : IIHSM
  - Wireless IDXP
- Data Capturing
  - Packet Sniffers
  - Traffic Generators
  - Network Testers
  - Traffic Analysers
  - Network Monitoring Tools
  - Network IHSs
- Data Processing
  p/o 1 Decoding Raw data link layer packets [Preprocessor] : Data
Processing
    a/o 1 Ethernet 802.11* : Data Processing
    a/o 1 Token Ring packets : Data Processing
    a/o 1 Higher layer protocols such as IP, TCP, and UDP : Data
Processing

(Relation Concept)
- equal
  p/i 1 value : number
  p/i 1 value : number
- sameAs
  p/i 1 class : Any
  p/i 1 class : Any
- larger-than
  p/i 1 larger : Any
  p/i 1 smaller : Any
```

E.9 Text output of Kismet IHS ontology based on IHSMO reference model

```
(Ontology Name)
Kismet IHS with Concepts and Slots_v1.ont.xml

(Wholeness Concept)
- IHSMO
  - Incoming Traffic
    - Libpcap/Winpcap
      p/o 1 *NIX Based Capturing System : Data Capturing
      p/o 1 Filtering System [ Filtering System] : Data Processing
  - Identification  Subsystem
    - No Hybrid Identification
      - Misuse Identification
        - RSDB Database
```

```
     p/o 1 Method A1 : Identification Database
     p/o 1 Method A2 : Identification Database
     p/o 1 Method An : Identification Database
 - Admin/Manager Console Ontology
  p/o 1 Updating RSDB : RSDB Database
  p/o 1 Manual Prevention : Admin Console
  p/o 1 Network Monitoring : Admin Console
- Identification Database
 p/o 1 Decoding based (layer2 Identifications) : Misuse Identification
  p/o 1 ARP decoding : Misuse Identification
  p/o 1 Layer 2 header decoding : Misuse Identification
 p/o 1 Frame structure based (layer 2 Identifications) : Misuse
Identification
  a/o 1 tests the entire frame control field : Misuse Identification
  a/o 1 tests the 802.11 frame's type : Misuse Identification
  a/o 1 tests the 802.11 frame's subtype : Misuse Identification
  a/o 1 tests the from distribution system frame control flag :
Misuse Identification
  a/o 1 tests the to distribution system frame control flag : Misuse
Identification
  a/o 1 tests the wep frame control flag : Misuse Identification
  a/o 1 tests the frame's duration/id field : Misuse Identification
  a/o 1 tests the frame's BSSID : Misuse Identification
  a/o 1 tests the frame's sequence number : Misuse Identification
  a/o 1 tests the frame's fragment number : Misuse Identification
- Data Capturing
 - Packet Sniffers
 - Traffic Generators
 - Network Testers
 - Traffic Analysers
 - Network Monitoring Tools
 - Network IHSs

(Relation Concept)
- equal
 p/i 1 value : number
 p/i 1 value : number
- sameAs
 p/i 1 class : Any
 p/i 1 class : Any
- larger-than
 p/i 1 larger : Any
 p/i 1 smaller : Any
```

E.10 Text output of IHSMO reference model and the missing parts of Kismet IHS System shown in highlight

The highlighted sections show the missing components of Kismet IHS system when evaluated using IHSMO reference model.

```
(Ontology Name)
IHSMO with Concepts and Slots_v2.ont.xml

(Wholeness Concept)
- IHSMO
  - Incoming Traffic
    - Libpcap/Winpcap
      p/o 1 *NIX Based Capturing System : Data Capturing
      p/o 1 Win Based Capturing System : Data Capturing
      p/o 1 Preprocessor [Packet Decoding, Filtering System] : Data
Processing
  - IICS
    p/o 1 IHMD : IHMD Database
    p/o 1 UIHRF : Intrusion Unified Report Format
    - IIHSM
      p/o 1 Method T1 : IICS Protocols
      p/o 1 Method T2 : IICS Protocols
      p/o 1 Method Tn : IICS Protocols
  - Identification  Subsystem
    - Hybrid Identification
      - Anomaly Identification
        p/o 1 Method L1 : Live Identification
        p/o 1 Method L2 : Live Identification
        p/o 1 Method Ln : Live Identification
      - Misuse Identification
        - RSDB Database
          p/o 1 Method A1 : Identification Database
          p/o 1 Method A2 : Identification Database
          p/o 1 Method An : Identification Database
  - Response Ontology
    - Active Response Ontology
      p/o 1 Method R1 : Defending/Active Response
      p/o 1 Method R2 : Defending/Active Response
      p/o 1 Method Rn : Defending/Active Response
    - Passive Response Ontology
      p/o 1 Method R'1 : Informing/Passive Response
      p/o 1 Method R'2 : Informing/Passive Response
      p/o 1 Method R'n : Informing/Passive Response
  - Admin/Manager Console Ontology
    p/o 1 Monitoring Security Policy based on Anomaly Identification :
Anomaly Identification
    p/o 1 Updating RSDB : RSDB Database
    p/o 1 Manual Prevention : Admin Console
    p/o 1 Network Monitoring : Admin Console
- Identification Database
  p/o 1 Decoding based (layer2 Identifications) : Misuse Identification
    p/o 1 ARP decoding : Misuse Identification
    p/o 1 Layer 2 header decoding : Misuse Identification
  p/o 1 Protocol based (layer3 Identifications) : Misuse Identification
    a/o 1 Protocols being analysed: tcp, udp, icmp, and ip : Misuse
Identification
    a/o 1 test the IP header's TTL field value : Misuse Identification
    a/o 1 test the IP header's TOS field value : Misuse Identification
    a/o 1 test the IP header's fragment ID field for a specific value :
Misuse Identification
    a/o 1 watch the IP option fields for specific codes : Misuse
Identification
    a/o 1 test the fragmentation bits of the IP header : Misuse
Identification
    a/o 1 test the packet's payload size against a value : Misuse
Identification
    a/o 1 test the TCP flags for certain values : Misuse Identification
    a/o 1 test the TCP sequence number field for a specific value :
Misuse Identification
    a/o 1 test the TCP acknowledgement field for a specific value :
Misuse Identification
    a/o 1 test the ICMP type field against a specific value : Misuse
Identification
    a/o 1 test the ICMP code field against a specific value : Misuse
Identification
    a/o 1 test the ICMP ECHO ID field against a specific value : Misuse
Identification
    a/o 1 test the ICMP ECHO sequence number against a specific value :
Misuse Identification
    a/o 1 filter for SYN/FIN/RST TCP packets : Misuse Identification
    a/o 1 process TCP fragments : Misuse Identification
    a/o 1 flag HTTP-based worm sources such as Code Red : Misuse
Identification
    a/o 1 detailed analysis of http requests : Misuse Identification
    a/o 1 detailed analysis of http replies : Misuse Identification
  p/o 1 Frame structure based (layer 2 Identifications) : Misuse
Identification
    a/o 1 tests the entire frame control field : Misuse Identification
    a/o 1 tests the 802.11 frame's type : Misuse Identification
    a/o 1 tests the 802.11 frame's subtype : Misuse Identification
    a/o 1 tests the from distribution system frame control flag :
Misuse Identification
    a/o 1 tests the to distribution system frame control flag : Misuse
Identification
    a/o 1 tests the more fragments frame control flag : Misuse
Identification
    a/o 1 tests the retry frame control flag : Misuse Identification
    a/o 1 tests the power management frame control flag : Misuse
Identification
    a/o 1 tests the more data frame control flag : Misuse
Identification
    a/o 1 tests the wep frame control flag : Misuse Identification
    a/o 1 tests the order frame control flag : Misuse Identification
    a/o 1 tests the frame's duration/id field : Misuse Identification
    a/o 1 tests the frame's BSSID : Misuse Identification
    a/o 1 tests the frame's sequence number : Misuse Identification
    a/o 1 tests the frame's fragment number : Misuse Identification
    a/o 1 tests the frame's 4th address field : Misuse Identification
  p/o 1 Other unclassified Identifications : Misuse Identification
```

```
    a/o 1 role : Any
- Live Identification
  p/o 1 Protocol based (layer3 Identifications) : Anomaly
Identification
    a/o 1 FTP analysis : Anomaly Identification
    a/o 1 identify and log TFTP sessions : Anomaly Identification
    a/o 1 ICMP analysis : Anomaly Identification
    a/o 1 DNS analysis : Anomaly Identification
    a/o 1 search for a set of patterns in the packet's payload :
Anomaly Identification
  p/o 1 Logging based : Anomaly Identification
    a/o 1 detects password scans : Anomaly Identification
    a/o 1 analysing rlogin/telnet : Anomaly Identification
    a/o 1 access and record connection events : Anomaly Identification
  a/o 1 Frame structure based (layer2 Identification) : Anomaly
Identification
    a/o 1 Networks with known WEP keys can be decrypted in realtime :
Anomaly Identification
    a/o 1 tracks probe requests and responses : Anomaly Identification
    p/o 1 Fuzzy encryption detection(UIHRF) : IHMD Database
    p/o 1 BSSID (MAC address) of the network(UIHRF) : IHMD Database
    a/o 1 Display Last-advertised channel for network : Anomaly
Identification
    a/o 1 Display Number of clients (unique MACs) seen on network :
Anomaly Identification
    a/o 1 Display Network status flags : Anomaly Identification
    a/o 1 Display Extra AP information included by some manufacturers :
Anomaly Identification
    a/o 1 Display Number of LLC packets : Anomaly Identification
    a/o 1 Identifies Manufacturer and Model pf AP and Client : Anomaly
Identification
    a/o 1 Display Maximum supported rate as advertised by AP : Anomaly
Identification
    a/o 1 Display Maximum supported rate as advertised by AP : Anomaly
Identification
    a/o 1 Display Name of the network or group : Anomaly Identification
    a/o 1 Display Last seen noise level : Anomaly Identification
    a/o 1 Display Last seen signal level : Anomaly Identification
    a/o 1 Display Network type (Probe, Adhoc, Infra, etc) : Anomaly
Identification
    a/o 1 Display WEP status  : Anomaly Identification
    p/o 1 Identifies unusual probes (UIHRF) : IHMD Database
    p/o 1 Identifies unusual probes (UIHRF) : IHMD Database
    p/o 1 Identifies excessive disassociation(UIHRF) : IHMD Database
    p/o 1 Identifies Netstumbler probe requests(UIHRF) : IHMD Database
    p/o 1 Identifies Deauthenticate/Disassociate Flood(UIHRF) : IHMD
Database
    p/o 1 Identifies Lucent link test(UIHRF) : IHMD Database
    p/o 1 Identifies SSID brute force attempt(UIHRF) : IHMD Database
    p/o 1 Identifies AP changing to a new channel(UIHRF) : IHMD
Database
    p/o 1 Identifies Broadcast disconnect/deauthenticate(UIHRF) : IHMD
Database
    p/o 1 Identifies Invalid BSS timestamps(UIHRF) : IHMD Database
    p/o 1 Identifies weak-iv packet (UIHRF) : IHMD Database
    p/o 1 Identifies  MITM attacks(UIHRF) : IHMD Database
    p/o 1 Identifies MAC spoofing attacks (UIHRF) : IHMD Database
  p/o 1 Other unclassified Identifications : Anomaly Identification
    p/o 1 Identifies vulnerability scans(UIHRF) : IHMD Database
    p/o 1 Identifies port scans(UIHRF) : IHMD Database
    a/o 1 watch & record RPC services for specific
application/procedure calls : Anomaly Identification
    a/o 1 record and analyse email traffic : Anomaly Identification
    a/o 1 track software versions : Anomaly Identification
    p/o 1 Identifies for blaster worm(UIHRF) : IHMD Database
    p/o 1 Identifies synflood attacks(UIHRF) : IHMD Database
    a/o 1 SSL analyser : Anomaly Identification
    p/o 1 Identifies incoming and outgoing connections that are
ssh(UIHRF) : IHMD Database
    p/o 1 Identifies backdoors(UIHRF) : IHMD Database
    a/o 1 Identifies clear text passwords : Anomaly Identification
    a/o 1 Promiscuous mode : Anomaly Identification
    a/o 1 search for a pattern in the URI portion of a packet : Anomaly
Identification
  p/o 1 Alert based : Anomaly Identification
    a/o 1 send alerts to syslog : Anomaly Identification
    p/o 1 write the alert in a UIHRF format : IHMD Database
    a/o 1 send alerts to a *NIX socket that another program can listen
on : Anomaly Identification
    a/o 1 Packets logged to a decoded ASCII format or a binary log file
: Anomaly Identification
    a/o 1 send WinPopup alert messages to Windows machines : Anomaly
Identification
    a/o 1 record log in a database : Anomaly Identification
    a/o 1 record log in a cvs file : Anomaly Identification
    a/o 1 sending alert to administrator's email or pager : Anomaly
Identification
    a/o 1 log alerts to event logs : Anomaly Identification
- Informing/Passive Response
  a/o 1 Terminate a connection : Any
  a/o 1 Block a malicious host : Any
  a/o 1 Edit router's ACL to block hostile hosts : Any
  a/o 1 Passive monitoring of network traffic : Any
- Defending/Active Response
  p/o 1 Alert based : Anomaly Identification
    a/o 1 send alerts to syslog : Anomaly Identification
    p/o 1 write the alert in a simple format(UIHRF) : IHMD Database
    a/o 1 send alerts to a *NIX socket that another program can listen
on : Anomaly Identification
    a/o 1 Packets logged to a decoded ASCII format or a binary log file
: Anomaly Identification
    a/o 1 send WinPopup alert messages to Windows machines : Anomaly
Identification
    a/o 1 record log in a database : Anomaly Identification
    a/o 1 record log in a cvs file : Anomaly Identification
    a/o 1 sending alert to administrator's email or pager : Anomaly
Identification
    a/o 1 log alerts to event logs : Anomaly Identification
    a/o 1 Watch Remote Procedure Call (RPC) protocol services for
specific application/procedure calls : Anomaly Identification
    a/o 1 Search for a pattern in the Uniform Resource Indicator (URI)
portion of packet : Anomaly Identification
  a/o 1 Terminate a connection : Any
  a/o 1 Block a malicious host : Any
  a/o 1 Edit router's ACL to block hostile hosts : Any
  a/o 1 dumps the application layer information for a given session :
Any
  a/o 1 block web sites : Any
- Admin Console
  - Defending/Active Response_2
- IHMD Database
- Intrusion Unified Report Format
- IICS Protocols
  p/o 1 Based on IDMEF : IIHSM
  p/o 1 Any Wireless  compatible exchange format : IIHSM
  - Wireless IDXP
- Data Capturing
  - Packet Sniffers
  - Traffic Generators
  - Network Testers
  - Traffic Analysers
  - Network Monitoring Tools
  - Network IHSs
- Data Processing
  p/o 1 Decoding Raw data link layer packets [Preprocessor] : Data
Processing
    a/o 1 Ethernet 802.11* : Data Processing
    a/o 1 Token Ring packets : Data Processing
    a/o 1 Higher layer protocols such as IP, TCP, and UDP : Data
Processing

(Relation Concept)
- equal
  p/i 1 value : number
  p/i 1 value : number
- sameAs
  p/i 1 class : Any
  p/i 1 class : Any
- larger-than
  p/i 1 larger : Any
  p/i 1 smaller : Any
```

Appendix F. Abstracts of the Published Papers

An Ontology Approach to Construction of Identification Subsystem for Intrusion Handling Systems in Wireless LANs**

S. Salekzamankhani, A. Pakštas, B.Virdee
Faculty of Computing
London Metropolitan University
Holloway Road, London, N7 8DB, UK

Abstract- **Intrusions to WLANs is a fact of everyday life and handling them is becoming more and more challenging. Currently there is no standardised reference model which can be used to compare and evaluate existing or design future Intrusion Handling Systems for WLANs. Hence this paper first reviews structure and components of Intrusion Handling System's reference model , then it describes and discusses the construction of Identification Subsystem Modelling Ontology (ISMO) of this model. The proposed ontology is based on the concepts of various ontology modelling and simulation tools. Careful attention is given to support two important functions: manage the dependencies between ontologies and at the same time to keep and restore their consistencies if they change in order to accommodate new information, or to adjust the representation of the domain as the world changes.**

**** Short listed for best paper awards in World Congress on Engineering (WCE), The 2010 International Conference of Wireless Networks (ICWN'10), Imperial College, 30 June - 2 July 2010,London, UK, ISBN: 978-988-17012-9-9.***

Ontology Approach to Construction of Response and Management Console Subsystems for Intrusion Handling Systems in Wireless LANs*

S. Salekzamankhani, A. Pakštas, B.Virdee
Faculty of Computing
London Metropolitan University
Holloway Road, London, N7 8DB, UK

Abstract– **Intrusive attacks to WLANs is pervasive nowadays and combating them is becoming more and more demanding. Currently there is no standardised reference model which can be used to compare and evaluate existing or design future Intrusion Handling Systems for WLANs. Hence, this paper describes and discusses the construction of Response Subsystem Modelling Ontology and Management Console Ontology for Intrusion Handling System reference model. The proposed ontology is based on the concepts of various ontology modelling and simulation tools. This gives careful attention to support two important functions, that is to manage the dependencies between ontologies and at the same time to keep and restore their consistencies if they alter in order to accommodate new information, or to adjust the representation of the domain as the world changes.**

**** Best Paper Award of the 2010 International Conference of Wireless Networks (ICWN'10) part of World Congress on Engineering (WCE), Imperial College, 30 June - 2 July 2010, London, UK, ISBN: 978-988-17012-9-9. (Ref:http://www.iaeng.org/WCE2010/congress_awards.html)***

Why we need a Reference Model for Intrusion Handling Systems for Wireless LANs?

S. Salekzamankhani, A. Pakštas
Department of Computing, Communications Technology and Mathematics
London Metropolitan University
Holloway Road, London, N7 8DB, UK

Abstract-**WLANs are different from the traditional wired LANs in terms of their exposure to potential threats, vulnerability and security techniques. Intrusions to WLANs are facts of the everyday life and handling them becomes more and more difficult. Currently there is no standardised reference model which may help to design, compare and evaluate the existing or future Intrusion Handling Systems (IHSs) for WLANs. Analysis of the commercial IHSs shows that they all are built as a proprietary systems which are neither taking into consideration existence of other IHSs nor they are trying to find the ways to establish inter-IHS collaboration which may help to achieve better security for the end-users. Hence this paper starts by an analysis of the features of existing IHSs for WLANs, follows with proposing the new ways to enhance their design, and then is discussing development of such a novel reference model for a scalable distributed IHS architecture.**

***Telecommunications and Computer Networks, SoftCOM2007, 15th International Conference,* 27-29 Sept, 2007 pp.1– 6. ISBN: 978-953-6114-95-5.**

Fighting Intrusions in Wireless LANs: A Need for the Reference Model

A. Pakštas, S. Salekzamankhani, B.Virdee
Department of Computing, Communications Technology and Mathematics
London Metropolitan University
166-220 Holloway Road, London, N7 8DB, UK
{a.pakstas, s.salek, b.virdee}@londonmet.ac.uk

Abstract-**A traditional way to improve security of wired computer network can be achieved by applying access control policies to the front door of network. However these policies may be not sufficient and effective any more when applied to the Wireless Local Area Networks (WLANs). WLANs are different from the traditional wired LANs in terms of their exposure to potential threats, vulnerability and security techniques. Intrusions to WLANs are facts of the everyday life and handling them becomes more and more difficult. Currently there is no standardised reference model which may help to design, compare and evaluate the existing or future Intrusion Handling Systems (IHSs) for WLANs. Analysis of the commercial IHSs shows that they all are built as a proprietary systems which are neither taking into consideration existence of other IHSs nor they are trying to find the ways to establish inter-IHS collaboration which may help to achieve better security for the end-users. Hence this paper starts by an analysis of the features of existing IHSs for WLANs, follows with proposing the new ways to enhance their design, and then is discussing development of such a novel reference model for a scalable distributed IHS architecture.**

***Proc. 2nd IEEE and IFIP International Conference in Central Asia on the Next Generation of Mobile, Wireless and Optical Communications Networks, (ICI 2006),* Tashkent, Uzbekistan , Sep. 19, 2006, ISBN: 1-4244-0543-2**

Towards Development of a Reference Model for Intrusion Detection Systems for Wireless LANs

S. Salekzamankhani, A. Pakštas, B.Virdee
Department of Computing, Communications Technology and Mathematics
London Metropolitan University
Holloway Road, London, N7 8DB, UK

Abstract-**Wireless Local Area Networks (WLANs) are different from the traditional wired LANs in terms of their potential threats, vulnerability and security techniques. The traditional mechanisms of protecting networks with firewalls and encryption software as it is done in the wired networks are no longer sufficient and effective in WLANs. The paper is focusing on the analysis of the existing Intrusion Detection Systems (IDSs) for WLANs and proposing ways to enhance their design. Currently there is no standardised reference model to help design, compare and evaluate the existing or future IDSs for WLANs. Hence this paper focuses on development of such a novel reference model.**

IEEE Globecom 2005, Workshop on Adaptive Wireless Networks, AWIN **Nov. 30, 2005, San Louis, USA.**

The end.