Ontology Approach to Construction of Response and Management Console Subsystems for Intrusion Handling Systems in Wireless LANs

S. Salekzamankhani, A. Pakštas and B.Virdee

***Abstract*– Intrusive attacks to WLANs is pervasive nowadays and combating them is becoming more and more demanding. Currently there is no standardized 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.**

***Index Terms*- Intrusion Detection/Prevention Systems, Ontology, Reference Model, Wireless Networks.**

I. Introduction

The rapid deployment of IEEE 802.11 also known as Wireless Local Area Networks (WLANs) can be credited to their obvious benefits over the traditional wired LANs by eliminating the need for cables. Hence, WLANs are different from wired LANs in terms of their exposure to potential threats, vulnerability and security techniques. Therefore there is an urgent need for an effective Intrusion Handling System (IHSs[1]) to decrease the severity of such threats.

Analysis of the commercial IHSs shows that they are all built as a proprietary systems which do not take into consideration the existence of other IHSs nor do they try to find ways to establish inter-IHS collaboration that may enable the realization of better security for the end-users [1,2,3].Fig. 1 shows the proposed IHS reference model in [1,2,3] which comprises of following components: identification subsystem, response subsystem, inter-IHS communication subsystem, management console, source data, inter-IHS communication systems, etc. The structure of response subsystem and management console is the focus of this paper. An ontology approach will be taken to describe its building block and its structure. This includes the construction of Response Subsystem Modelling Ontology (RSMO) and Management Console Ontology (MCO) for IHS reference model.

In recent years the development of ontologies (explicit formal specification of the terms in the domain and relations among them [4]) has been moving from the realm of the Artificial-Intelligence laboratories to the desktops of domain experts. Many disciplines now develop standardised ontologies which can be used by domain experts to share and annotate information [5].

The main reasons of ontology development [4] include:

- 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.

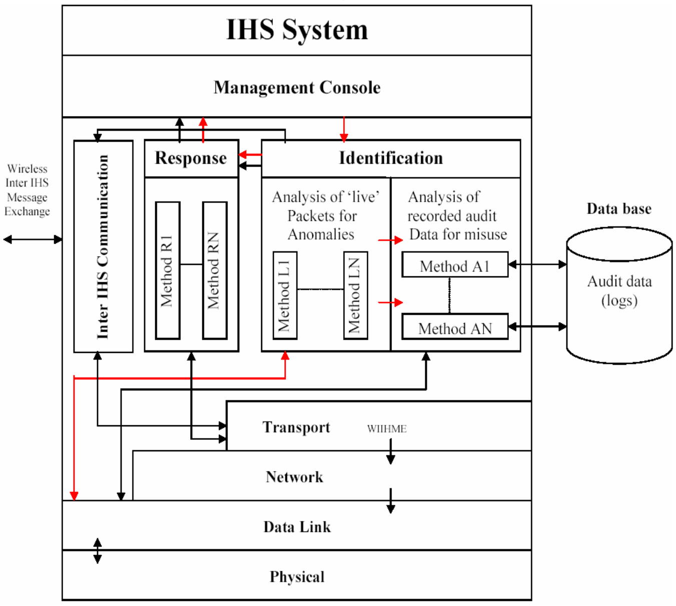

FIG. 1. ARCHITECTURE OF IHS REFERENCE MODEL

Manuscript received February 17, 2010.

S. Salekzamankhani is with Faculty of Computing, London Metropolitan University, London, N7 8DB, UK (Phone: 020 7133 2207, email: s.salek@londonmet.ac.uk).

A. Pakštas is with Faculty of Computing, London Metropolitan University, London, N7 8DB, UK (email: a.pakstas@londonmet.ac.uk)

B. Virdee is also part of Faculty of Computing, London Metropolitan University, London, N7 8DB, UK (email: b.virdee@londonmet.ac.uk)

[1] In this paper, the general term "Intrusion Handling System" (IHS) is used as described in [1,2] with the exception of Intrusion Detection/Protection Systems.

Sharing common understanding of the structure of information among people or software agents is one of the more common goals in developing ontologies [4,6].

There are some related works [11, 12] on legacy IDS systems; however the new IHS reference model is different to them. This paper is focusing on a reference model and not a system itself. Hence the new IHS reference model can be used to evaluate the legacy IDS systems.

II. Ontology Terminology and Formal Definitions

A. Basic Terminology

Classes represent *concepts* in the domain and not the words that denote these concepts. It should be remembered that an 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 [5].

Slots are the properties of each concept describing various features and attributes of the concepts. It is also called a *role* or *property* [5].

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*. The name of a class may change if a different terminology is chosen, but the term itself represents the objective reality in the world [5].

A *Taxonomy* is a hierarchy of concepts which defines relationship between concepts with the help of links such as an “is-a” or “part-of” link [7].

A *vocabulary* is a set of words where each word indicates some concept. Vocabulary is language dependent [7].

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 as follow:

1) To represent the meaning of concepts rigorously.

2) 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 [8].

Finally a formal ontology is axiomatic description of an ontology. It can answer questions about the capability of ontology. An 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. We do not have to use logic to describe it. Formally an ontology consists of terms, their definitions and axiom relating to them; terms are typically organized in a taxonomy [8].

B. Symbols

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 are used for the definitions of the ontology construction. As shown below the concept/class is represented by a rectangle and the slot/attribute is shown by ellipse/oval. It can be noticed that the links between concepts, slots are represented by two different arrows indicating the *part-of* or *is-a* relationship. The first arrow shows the *part-of* relation between concept to concept and second arrow shows the *part-of* relation for concept to slot respectively.

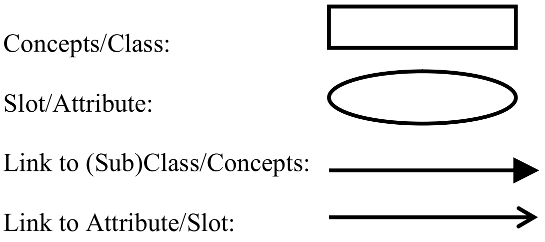

Table I shows symbols which are used to represent axioms.

C. Axioms

Some axioms which will be used are borrowed from [7,8,9]. 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:

$$(\forall x)\neg partOf(x,x) \qquad \text{(A1)}$$

$$(\forall x,y)partOf(y,x)\rightarrow\neg \mathrm{partOf}(x,y) \qquad \text{(A2)}$$

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 a component of the third part.

TABLE I
Symbols Used to Represent the Axioms

Symbol	**Meaning**
∀	For all
∃	There exist
¬	Not
∧	And
∨	Or
→	Implication
↔	Equivalent
⊆	Belong to
∪	Union

$$(\forall x,y,z)\ partOf(\mathrm{z,y}) \wedge partOf(\mathrm{y,x}) \rightarrow partOf(z,x) \qquad \text{(A3)}$$

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.

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

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.
  $$(\forall x)\mathrm{primitive}(x) \rightarrow (\neg\exists y) partOf(y,x) \qquad \text{(A5)}$$
  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.
  $$(\forall x)\mathrm{composite}(x) \rightarrow \neg\mathrm{primitive}(x) \qquad \text{(A6)}$$

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

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

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.

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

That is, *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 of *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:

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

III. Design of Response Subsystem Modelling 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 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 RSMO ontology identify separately according to their domain or conceptual level before they are all compiled to realize a single unified RSMO ontology.

The Intrusion Response Subsystem (IRS) is capable of stopping attacks against a given network and provides the following real-time defense mechanism:

- Prevention: it permanently stops detected misuse attacks from executing. In the case of anomaly based attacks, it temporarily stops the detected attack from executing either automatically or via true management console. In scenarios, anomaly and misuse, the IHS would send logs to a management console through the response subsystem.
- Reaction in anomaly based attacks: it immunizes the system from future attacks from the same malicious source by storing the rule-set created by administrator so the next time the same attack will be considered 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 action, the IHS will establish a manual action method. This is referred to as informing or passive response.

B. Concepts and Axioms of RSMO

The attributes/slots, concepts/class, possible constraints and values related to the response ontology are shown in Table II:

A value-type facet shown below describes what types of value can fill in the slot or concept. The most common value type is 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 II is to represent the composite types and other facets, also used in [5].

TABLE II
THE CONCEPTS FOR RESPONSE ONTOLOGY AND THEIR FACETS

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

TABLE III
AXIOMS FOR RESPONSE CONCEPT

ID	Axioms	ID	Axioms
1	partOf(Active,Response)	2	partOf(Passive, Response)
3	Composite (Active)	4	Composite (Passive)
5	(∀*x*, ∃*y*) Response (*x*) ∧ Active (*y*) ∧ partOf (Active, Response) ∨ True		
6	(∀*x*, ∃*y*) Response (*x*) ∧ Passive (*y*) ∧ partOf (Passive, Response) ∨ True		

Similar approach to represent axioms in Table II have also been used in [7,8,10].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 ⊆ E therefore: E=F∪G

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

The active concept can be expressed by the attributes/slots Method R1, MethodR2 and Method Rn. The different methods will be described later in this section. These attributes belong to the class active response as shown in Fig. 2.

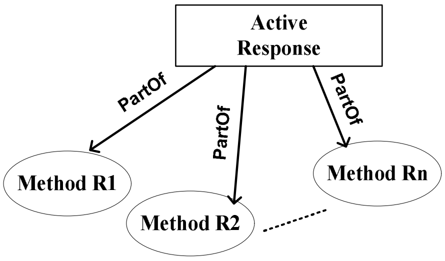

FIG. 2. ACTIVE RESPONSE CONCEPT

TABLE IV
AXIOMS FOR ACTIVE CONCEPT

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

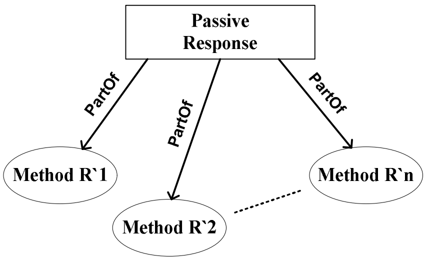

FIG. 3. PASSIVE RESPONSE CONCEPT

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

f1: Method R1, f2: Method R2,……, fn: Method Rn

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

f1, f2,….fn ⊆ F hence F = f1∪f2…. ∪fn

The axioms of active concepts are shown in Table IV.

In general, possible methods in order to prevent intruder are summarized below:

- 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, MethodR'2 and Method R'n. The different methods will be describes later in this section. These attributes belong to the class passive response as shown in Fig. 3.

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

g1: Method R'1, g2: Method R'2,……, gn: Method R'n

It can be noted that these slots are subsets of concept G (Passive response), therefore:

g1, g2,….gn ⊆ G hence G = g1∪g2…. ∪gn

The axioms of passive concepts are shown in Table V.

In general, the following are the reaction of passive response methods (i.e. manual action method) in order to prevent intruder:

TABLE V
AXIOMS FOR PASSIVE RESPONSE CONCEPTS

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)

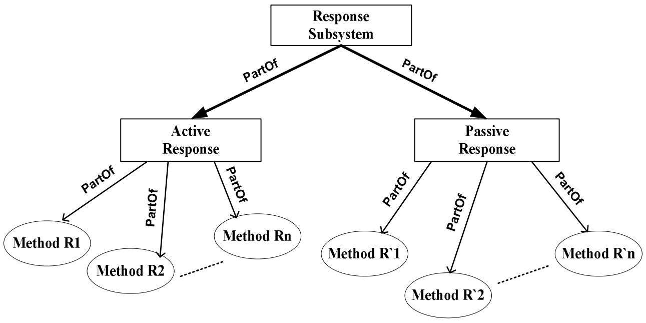

FIG. 4. FINAL VIEW OF RESPONSE ONTOLOGY

- 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 is composed of the Active and Passive response ontologies and can be formally described as

E=F∪G

Fig. 4 shows a final view of Response ontology.

IV. MANAGEMENT CONSOLE ONTOLOGY

An administrator is defined as a person who setups the network and is responsible for management, security and maintenance. An administrator is also responsible for managing operators. In fact an administrator has full control over manager/management console. Therefore for the sake of ontology definition it has been assumed that the administrator and manager have the same responsibilities.

The attributes/slots, concept/class, possible constraints and values related to the Management Console Ontology (MCO) are shown in Table VI.

For simplicity the concepts and slots of MCO are denoted as follows:

TABLE VI
THE SLOTS FOR ONTOLOGY OF MANAGEMENT ENTITY AND FACETS FOR THESE SLOTS

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

TABLE VII
AXIOMS FOR MANAGEMENT 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		

- H: Administrator/Manager Console;
- h1: Security Policy Database;
- h2: Updating Rule Set Database;
- h3: Manual Prevention;
- h4: Network Monitoring.

It can be noted that these slots are all subsets of H which are formally described as:

$h1, h2, h3, h4 \subseteq H$ hence: $H = h1 \cup h2 \cup h3 \cup h4$

The axioms of MCO are shown in Table VII and its final view in Fig. 5.

V. CONCLUSIONS

A unique IHS reference model structure was described in this paper which employs an ontology approach to define response and management console ontology modelling.

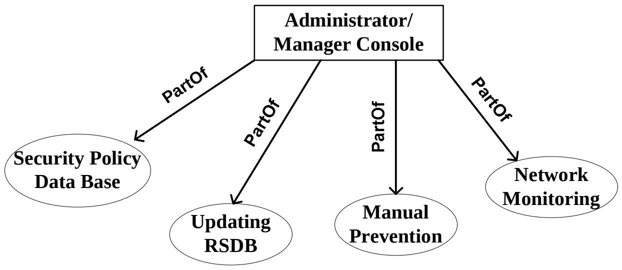

FIG. 5. FINAL VIEW OF MANAGEMENT CONSOLE ONTOLOGY

A novel response management console subsystems have been designed and proposed for IHSs in WLANs. Use of such reference model should allow the characterization of different IHSs in a standardised and efficient format.

The future paper will focus on the ontology approach to define the Inter IHS communications subsystem and finally demonstrate the final version of Ontology based IHS reference model which has been evaluated using existing WLANs IHS systems to prove its efficiency and accuracy in order to compare and evaluate the existing or future IHSs for WLANs.

REFERENCES

[1] S. Salekzamankhani, A. Pakštas, "Why we need a reference model for intrusion handling systems for Wireless LANs?" Telecommunications and Computer Networks, 2007. SoftCOM2007. 15th International Conference on Volume, Issue, 27-29 Sept. 2007 Page(s):1 – 6.

[2] 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

[3] 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,

[4] T.R. Gruber, "A translation approach to portable ontology specifications", Proc. of JKAW, 1992. pp 89-108. Available: http://portal.acm.org/citation.cfm?id=173747

[5] 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. Available:http://protege.stanford.edu/publications/ontology_development/ontology101.pdf.

[6] M.A. Musen, "Dimensions of knowledge sharing and reuse", Computers and Biomedical Research 25, 1992, pp 435-467. Available: http://portal.acm.org/citation.cfm?id=176404.

[7] M.A Rahman, A. Pakštas, 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.

[8] 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.

[9] J. Lin, M.S. Fox, T. Bilgic, "A requirement ontology for engineering design", Concurrent Engineering: Research and Applications, Vol. 4, No4, Sept1996, pp279-291. Available:http://cer.sagepub.com/cgi/content/abstract/4/3/279.

[10] 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.

[11] S.Hung, D.Liu, "A user-oriented ontology-based approach for network intrusion detection", ACM Portal, Volume 30, Issue 1-2, Pages 78-88. 2008. Available: http://portal.acm.org/citation.cfm?id=1298807.

[12] J. Undercoffer, A. Joshi, and J.Pinkston, "Modeling Computer Attacks, An Ontology for Intrusion Detection" Proc. The Sixth International Symposium on Recent Advances in Intrusion Detection, Springer, LNCS-2516, pp.113–135,Sept2003.Available: http://ebiquity.umbc.edu/paper/html/id/64/Modeling-Computer-Attacks-An-Ontology-for-Intrusion-Detection