On Web, Semantics, and Data Mining: Intrusion Detection as a Case Study

Anupam Joshi and Jeffrey Undercoffer
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
{joshi, junder2}@cs.umbc.edu

Abstract

We examine the intersection of data mining and semantic web in this paper. We briefly identify some points where they can impact one another, and then develop a specific example of intrusion detection, an application of distributed data mining. We have produced an ontology specifying a model of computer attacks. Our model is based upon an analysis of over 4,000 classes of computer attacks and their corresponding attack strategies using data derived from CERT/CC advisories and NIST's ICAT meta-base. We present our attack model first as a taxonomy and convert it to a target-centric ontology that will be refined and expanded over time. We state the benefits of forgoing dependence upon taxonomies for the classification of computer attacks and intrusions, in favor of ontologies. We illustrate the benefits of utilizing an ontology by comparing a use case scenario of our ontology and the IETF's Intrusion Detection Exchange Message Format Data Model.

1 Introduction

The field of data mining has been the object of much attention over the last decade. In its broadest sense, it has been defined as the extraction of "implicit, non trivial knowledge" from data [19]. However, despite this definition, much of the data mining research has been more narrowly focussed on mathematical or statistical methods, and has built on prior work in pattern recognition. Clustering, Association (Correlation) Rules and Sequence analysis methods all typically operate by looking for "patterns" at the syntactic level, as it were.

Separate from the data mining work is a large body of work in logic, that in some sense attacks a similar problem when it "infers" new knowledge from given facts. The advent of the semantic web provides a new opportunity for exploring the possible interrelations between these hithertofore disparate fields.

One can argue that this interaction is enabled by the semantic web for two reasons. For one, mining the web has been one of the major applications of data mining. This has taken several forms, from mining of web log data, to mining of link structures to actual text mining on the content of the pages. The addition of semantic tags to the web content clealy provides an opportunity to revisit some of the techniques developed for web mining. Secondly, and perhaps more generally, the web is also today used as a platform for distributed data access. One can therefore also examine how (distributed) data mining in general is impacted when the data sources have semantically rich markup.

In this paper, we will talk about some possible application scenarios where data mining can interact with the semantic web and make use of semantics associated with the data. We will then delve in some detail about an intrusion detection application which combines semantics with data mining.

2 Semantic Web meets Data Mining

Perhaps the most obvious instance of interaction between data mining and semantic web is web mining. Let us consider the case of web log mining first. There has been a significant body of work in web log mining (e.g. [2, 4, 18]), including our own prior work [10, 15]. Most of it relies on finding co-occurance on web pages in access logs made by the same user or same user groups. In other words, given a user the attempt is to identify which set of pages she visits. A critical problem here is identifying similar pages. Much of this research has either ignored this problem, or used purely syntactic means to infer semantic similarity (such as using the document tree heirarchy). Clearly, semanticlly rich metadata can help here. Of course one could always do text analysis of the pages to indentify similarity, but that is very computationally intensive. Even text mining tasks could be guided by the semantic metadata.

As another instance, consider association rule type techniques. In most recent work in this space, the emphasis has been on ”pruning” itemsets that will prove uninteresting. Since the techniques constrain themselves at the syntactic level, the pruning is essentially based on support judgements. While this is certainly a valid approach, it constrains the notion of uninteresting. Sometimes, associations can be uninteresting because of the semantics – in other words a rule with adequate support and confidence can still not be useful to the end-user. Semantically described domain models can thus help in the pruning process.

Finally consider distributed data mining. Today, much of the work in distrtibuted data mining assumes that the data sources are predefined, and the key task is to be able to mine across the sources without moving the data to any centralized site. There are also efforts to make this process privacy preserving. However, in many critical applications, the sources to be mined will need to be dynamically selected based on the patters that are discovered. We bring out this particular application in the next section.

3 Ontologies and Distributed Data Mining for Intrusion Detection

Intrusion detection is a very important component of modern day security systems. In standalone host based IDSs, data gathered at the machine (network logs, audit logs, kernel parameters etc.) are ”mined” to find patterns that would indicate an intrusion (buffer overflow, rootkits etc.). In the past few years, there has been the recognition that intrusion detection is often a distributed task. In other words, only parts of the evidence of an attack can be found in any particular host or router. Often, when one host suspects an intrusion, it needs to cooperate with others to verify and confirm that an attack actually ocurred. The mining is thus distributed. However, the sources where the data has to be mined is known only when an attack is suspected. The places where data will be mined will be different when there is a DDoS attack compared to when IP spoofing is suspected. We show how developing a domain model in a semantic web language can help guide the distributed data mining task.

Based upon empirical evidence we have produced a model of computer attacks categorized by: the system component targeted, the means and consequence of attack, and the location of the attacker. Our model is represented as a *target-centric* ontology, where the structural properties of the classification scheme is in terms of features that are observable and measurable by the target of the attack or some software system acting on the target's behalf. In turn, this ontology will be used to facilitate the reasoning process of detecting and mitigating computer intrusions.

Traditionally, the characterization and classification of computer attacks and other intrusive behaviors have been limited to simple taxonomies. Taxonomies, however, lack the necessary and essential constructs needed by an intrusion detection system (IDS) to reason over an instance representative of the domain of a computer attack. Unlike taxonomies, ontologies provide powerful constructs that include machine interpretable definitions of the concepts within a domain and the relations between them. Ontologies provide software systems with the ability to share a common understanding of the information at issue in turn enabling the software system with a greater ability to reason over and analyze this information.

As detailed by Allen, et. al [1], and McHugh [14], the taxonomic characterization of intrusive behavior has typically been from the attacker's point of view, each suggesting that alternative taxonomies need to be developed. Allen et. al state that intrusion detection is an immature discipline and has yet to establish a commonly accepted framework. McHugh suggests classifying attacks according to protocol layer or, as an alternative, whether or not a completed protocol handshake is required. Likewise, Guha [8] suggests an analysis of each layer of the TCP/IP protocol stack to serve as the foundation for an attack taxonomy.

The Intrusion Detection Working Group of Internet Engineering Task Force (IETF) has proposed the Intrusion Detection Message Exchange Requirements [25] which, in addition to defining the requirements for the Intrusion Detection Message Exchange Format, also specifies the architecture of an intrusion detection system (IDS). The Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML)Document Type Definition [5] (IDMEF) is a profound effort to establish an industry wide data model which defines computer intrusions. IDMEF has its shortcomings, however. Specifically, it uses XML which is limited to a syntactic representation of the data model¿ This limitation requires each IDS to interpret and implement the data model programatically. Moreover, XML does not support the notion of inheritance, which means that the data model will not benefit from substitutability – a property allowing a value of a subtype to be used in place of a supertype without prior knowledge of the subtype.

As an alternative to IDMEF, we propose a data model represented by an ontology representation language such as the Resource Description Framework Schema (RDFS) [20]. We illustrate the benefits of using ontologies for IDS's by presenting an example of our ontology being utilized by IDSs supported by *SHOMAR* [22], a framework for distributed intrusion detection services. SHOMAR is an optimization of [21] and [11] architectures that provide secure service discovery and access in heterogeneous network and computing environments.

Generally, IDS's are either adjacent to or co-located with the target of an attack. It is imperative, therefor, that any classification scheme used to represent an attack be *target-centric*, where each taxonomic character is comprised of properties and features that are observable by the target of the attack. Consequently, our taxonomy, and subsequently our ontology, defines properties and attributes In terms of characteristics that are observable and measurable by the target of an attack.

As a basis for establishing our *a posteriori* target-centric attack ontology we evaluated and analyzed over 4,000 computer vulnerabilities and their corresponding attack strategies.

4 Target-Centric Taxonomy

In gathering data for our study, we relied upon the *CERT/CC Advisories* maintained by the “Computer Emergency Response Team/Coordination Center” of Carnegie Mellon University’s Software Engineering Institute and the *“Internet Catalog of Assailable Technologies”* (ICAT) maintained by the National Institute of Standards. Both provide a listing of known computer vulnerabilities and exploits. CERT obtains its data from computer incident reports made by the public at large. CERT, after a forensic examination of the reported incident, and providing the incident has wide spread impact, posts an advisory. ICAT is a compilation of vulnerabilities derived from multiple sources, including but not limited to: CERT, Internet Security Systems (ISS), Bugtraq, Microsoft and Security Focus.

Currently, the ICAT meta-base contains 4,160 entries and is classified according to severity, loss type, vulnerability type, exposed system component, etc. The ICAT classification scheme is not mutually exclusive. Therefore, for our study, we only considered 4,048 entries from the ICAT data set. Furthermore, we reclassified many of the ICAT entries to ensure that each subcategory was mutually exclusive and non-ambiguous. For example, ICAT lists the exposed component of the *Land* attack as both the network protocol stack and the operating system as well as stating that multiple vulnerabilities are responsible for enabling the *Land* attack: “Input Validation Error”, “Buffer Overflow”, “Boundary Overflow” and an “Exceptional Condition Handling Error”. CERT, however, states that *Land* is an attack comprised of a SYN packet in which the source address and port are the same as the destination address and port, resulting in an input validation error.

CERT has issued 286 advisories since its inception in 1985, and we have included all of these in our study. We compared the statistics derived exclusively from the CERT advisories with those derived from ICAT (which includes CERT) for continuity between the two data sets.

The purpose of our analysis is to identify the means of attack that are most frequently employed (i.e. as manifested at and experienced by the target), the most likely consequence of an attack (i.e. as experienced by the target), the component of the target that is most often targeted by an attack and the most common location from whence the attack originated.

Because an IDS has no knowledge of the attacker’s motivation or the tools employed to conduct the attack we believe that to be successful, the IDS needs to focus on evaluating the information which is readily available. Therefore, our taxonomy is classified according to features and characteristics directly observable at the target. Our feature set is predicated upon the result of our analysis. Our target-centric taxonomy follows:

1. **Target of Attack**. The system component that is the target of an attack. Specifically, these are the Network Protocol Stack, the Kernel, Applications and other components such as modems.

2. **Means of Attack**. The method that was used by the attacker as is manifested at and experienced by the target. This category includes input validation errors (buffer overflows, boundary condition errors, etc.), exploits and configuration errors.

3. **Consequences of Attack**. The end result of the attack. This category includes: Denial of Service, unauthorized user access, unauthorized root access and a loss of confidentiality.

4. **Location of Attack**. The location of the attacker. Indicated by whether the attacker is connected via the network or local host.

5 From Taxonomies to Ontologies: The case for ontologies

In [16], Ning et. al propose a hierarchical model for attack specification and event abstraction using three concepts essential to their approach: *System View, Misuse Signature* and *View Definition*. Their model is based upon a thorough examination of attack characteristics and attributes. However, their model is encoded within the logic of their proposed system. Consequently, it is not readily interchangeable and reusable by other systems.

Similarly, the Intrusion Detection Working Group of the Internet Task Force has defined the *Intrusion Detection Message Exchange Format Data Model* (IDMEF) [5] to describes a data model to represent information exported by IDS’s and by individual components of distributed IDS’s. Although the IDMEF specification states: “... the Intrusion Detection Message Exchange Format is intended to be a standard data format that automated intrusion detection systems can use to report alerts about events that they deem suspicious” it also specifies the architecture of an Intrusion Detection System and models some attacks. IDMEF uses the Extensible Mark-up Language (XML) [23] to encode the data model, consequently, due to XML’s limitations, the data model is not contained within the XML declarations but rather in the logic of how the particular IDS interprets the XML declarations.

Because IDMEF is specified in an XML Document Type Definition (DTD) [7] it does not convey the semantics, relationships, attributes and characteristics of the objects which it represents. Moreover, XML does not support the notion of inheritance.

In commenting on the IETF's IDMEF, Kemmerer and Vigna [12] state *"it is a but a first step, however additional effort is needed to provide a common ontology that lets IDS sensors agree on what they observe"*.

According to Davis et. al [6] knowledge representation is a surrogate or substitute for an object under study. In turn, the surrogate enables an entity, such as a software system, to reason about the object. Knowledge representation is also a set of *ontological* commitments specifying the terms that describe the essence of the object. In other words, *meta-data* or data about data describing their relationships.

Frame Based Systems are an important thread in knowledge representation . According to Koller, et al., [13] Frame Based Systems provide an excellent representation for the organizational structure of complex domains. Frame Based Languages, which support Frame Based Systems, include RDF, and are used to represent ontologies. According to Welty et. al [24] at its deepest level an ontology subsumes a taxonomy. Similarly, Noy and McGuinness [17] state the process of developing an ontology includes arranging classes in a taxonomic hierarchy.

The relationship among data objects may be highly complex, however at the the finest level of granularity, the *Knowledge Representation* of any object may be represented as an *RDF* (Resource Description Framework) statement [3] which formally defines the RDF model as:

1. A set called *Resources*.
2. A set called *Literals*.
3. A subset of Resources called *Properties*
4. A set called *Statements*, where each element is a triple of the form:
   {*sub, pred, obj* }
   Where *pred* is a member of Properties, *sub* is a member of Resources,
   and *obj* is either a member of Resources or a member of Literals.

Figure 1 illustrates the basic RDF model.

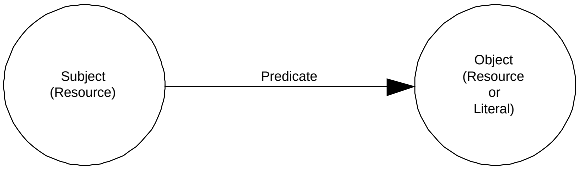

Figure 1. RDF Graph

Additionally, the relationship between a set of objects may be described graphically (as in Figure 3), as a series of *N-triples*, or by an RDF statement.

While RDF defines a model for describing relationships among objects in terms of properties and values, the declaration of these properties and their corresponding semantics are defined in the context of RDF as an RDF schema (RDFS) [20]. In applying RDFS to the problem of intrusion detection the power and utility of RDFS is not simply in representing the attributes of the attack, but rather the in the fact that we can express the relationships between collected data and use those relationships to deduce that the particular data represents an attack of a particular type.

Moreover, specifying an ontology decouples the data model representing an intrusion from the logic of the intrusion detection system. The decoupling of the data model from the IDS logic, specifying it as an ontology, enables non-homogeneous IDS's to share data without a prior agreement as to the semantics of the data. To effect this sharing, the ontology is made available and if the recipient does not understand some aspect of the data it obtains the ontology in order to interpret and use the data.

Ontologies therefore, unlike taxonomies, provide powerful constructs that include machine interpretable definitions of the concepts within a specific domain and the relations between them. In our case the domain is that of a particular computer or a software system acting on the computer's behalf in order to detect attacks and intrusions. Ontologies may be utilized to not only provide IDS's with the ability to share a common understanding of the information at issue but also further enable the IDS with improved capacity to reason over and analyze instances of data representing an intrusion. Moreover, within an ontology characteristics such as cardinality, range and exclusion may be specified and the notion of inheritance is supported.

5.1 Target Centric Ontology

Figure 2 presents a high level graphical illustration of our target-centric ontology that is built upon our taxonomy. An ellipse is used to denote a subject and object while an arc represents the predicate (relationship). Note the addition of the node labeled *Input* which is a superclass of the taxonomic items *Component, Means* and *Location*. Accordingly, an intrusion is **comprised** of some input **resulting** in some consequence, while the input is **directed** towards a a system component, **received** from some location and **causes** some means of by inducing some system behavior. Figure 3 presents our complete ontology in graphical form. Instances of data are represented at the leaves of the graph.

IDMEF, in contrast to an ontology represented by RDF, must work within the constraints imposed by XML, which only provides a syntax for communicating that an attack of a particular type has occurred. IDMEF does not directly contribute to or facilitate the detection and reasoning process. Specifically, once the attack has

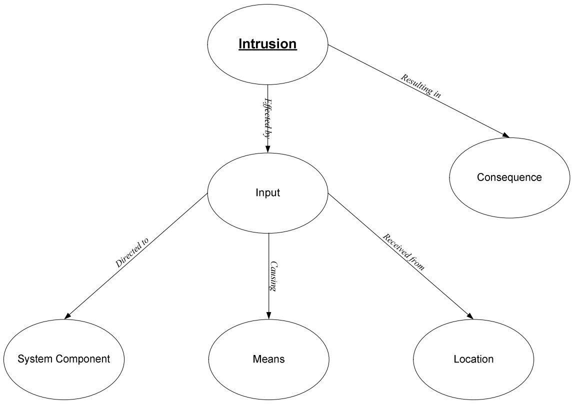

Figure 2. High Level Illustration of the Target-Centric Attack Ontology

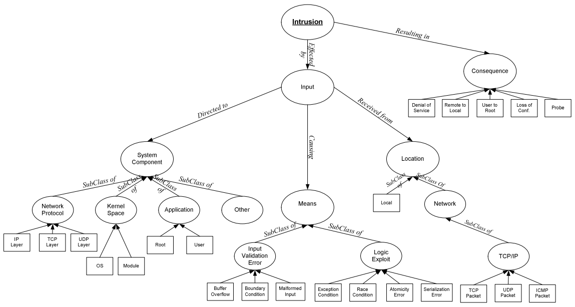

Figure 3. Graphical Presentation of the Target-Centric Attack Ontology

been detected, and its type, source and target identified, IDMEF only provides a format for communicating information concerning the event. How this information is interpreted and used is solely dependent upon the meaning imposed by the receiver of the information – which may or may not be the same as was intended by the originator of the communication. This is not the case with an ontology. The benefit of the ontology is that everyone that uses the ontology imparts the same semantic meaning on instances of the ontology. Moreover, an ontology is easily extensible as new attack types can be added as subclasses.

5.2 Example Attack Scenario

As previously stated the power of an ontology, as specified in *RDFS*, is not in the taxonomic representation of an attack, but rather because RDFS expresses the the data objects and the relationships (semantics) between the instances of those data objects. In turn this enables all reasoning engines that share the ontology to infer that an attack of a certain type has occurred. An additional and important benefit is that the data model is decoupled from, and external, to the programmatic (i.e.: logic) representation of the IDS and is available to other IDSs through the use of a Uniform Resource Identifier (URI). We use the *Mitnick* attack to illustrate the utility of our ontology.

5.2.1 The Attack

The Mitnick attack is multi-phased consisting of a Syn/Flood attack, TCP sequence number prediction and IP spoofing. The attack incorporates yet another attack, Syn/Flood, to effect a denial of service attack on a specific host that has a trust relationship with target of the attack. In the following example **Host B** is the ultimate target and **Host A** is trusted by **Host B**. The attack is as follows:

1. The attacker initiates a Syn/Flood attack against **Host A** to prevent **Host A** from responding to **Host B**.

2. The attacker then attempts to open multiple TCP connections to the target, **Host B** in order to be able to predict the values of TCP sequence numbers generated by **Host B**.

3. The attacker then pretends to be **Host A** by spoofing **Host A**'s IP address and sends a Syn packet to **Host B** in order to establish a TCP session between **Host A** and **Host B**.

4. Because its input queue is full due to the half open connections caused by the Syn/Flood attack, **Host A** cannot send *RST* message to **Host B** in response to the Syn message sent by the attacker purporting to be from **Host A**/.

5. Using the calculated TCP sequence number of **Host B** (recall that the attacker did not see the Syn/ACK message sent from **Host B** to **Host A**) the attacker sends an *Ack* with the predicted TCP sequence number packet in response to the *Syn/Ack* packet sent by **Host B**.

6. **Host B** is now in a state where it believes that a TCP session has been established with a trusted host **Host A**. The attacker now has a one way session with the target, **Host B**, and can issue commands to the target.

5.2.2 Detecting the Attack

Consider an environment of distributed intrusion detection services where the specific IDS architecture (component type, name, function, etc.) is abstracted by the

SHOMAR framework [22]. Moreover, consider that all components of the architecture use our ontology which not only specifies the data model for attacks.

Suppose that an IDS has either "learned" or has been presented with an instance of an ontology characterizing normal behavior during TCP connection establishment (i.e.: *Three Way Handshake*). It is important to note that this "normal behavior" instance of our ontology expresses the temporal relationship between the receipt of a *Syn* the transmission of an *Ack/Syn* and the receipt of the *Ack* establishing the connection as well as the ordering of TCP packet and fragment numbers.

Suppose that some system **Host A** under observation by the IDS has Σ pending TCP connections in time Δ, where Σ and Δ represent quantitative and temporal thresholds specified in the ontology. Furthermore, suppose that the system has responded with *Syn/ACk* messages but has failed to receive the *Ack* completing the handshake. Referring to Figure 3 where the leaves of the tree labeled *TCP Layer, TCP Packet* and *Denial of Service* represent specific instances of data, the IDS now has objects representing the system condition as follows:

- Multiple instances of TCP messages received from remote location(s) and received by the Protocol Stack of the target system.
- Half Open TCP connections.
- A degradation of resources becuase the connection queue for that port is full.

This information representing the system condition will result in **Host A**'s IDS reasoning engine inferring that system's condition is a specific instance of our ontology defining a *Syn/Flood* attack that has been previously asserted into our knowledge base. Information about this instance may now be made available to all other IDS within the *Shomar* framework. Specifically, a message to all IDS in the coalition stating the IP Address::Port Number of **Host A** is the target of a denial of service attack starting at some specific time. Due to the shared ontology each and every IDS in receipt of this message will have a clear understanding of its meaning and implication.

Now suppose that **Host B** has experienced several connection attempts (that were an attempt to determine its TCP sequence numbers) wherein it immediately responded with *RST* messages. As this behavior is aberrant facts about it are asserted into **Host B**'s knowledge base. Now suppose that **Host A** has either established or is about to establish a connection with IP Address::Port Number of **Host B** as reported being involved in the denial of service attack. As our ontology defines an instance where the consequence of a denial of service attack is that any communications established with the target of the attack are themselves the target of a *Mitnick* attack, the IDS operating on behalf of **Host A** will reason that it is also the target of an attack.

This example demonstrates the semantic power expressed by the ontology, specifically that it conveys the implications that one sequence of events (the *Syn/Flood* attack) may have on another set of events.

6 Conclusion and Future Work

We have analyzed vulnerability and intrusion data derived from CERT advisories and NIST's ICAT metabase resulting in the identification of the components (network, kernel-space, application and other) most frequently attacked, the means of attack, the consequences of the attack and the location of the attacker. Our analysis shows that non-kernel space (non operating system) applications, running as either root or user, are the most frequently attacked and are attacked remotely. The most common means of attack are exploits other then buffer overflows and other forms of deliberately malformed input data. According to CERT advisories issued in response to severe vulnerabilities, *root* access is the most common consequence of an exploit whereas the ICAT data shows *denial of service* to be the most common consequence.

Our analysis was conducted in order to identify observable and measurable taxonomic characteristics of computer attacks and intrusions. Accordingly, we developed a taxonomy characterized by *System Component, Means of Attack, Consequences of Attack* and *Location of Attacker*. We have stated the case for replacing simple taxonomies with ontologies for use in IDS's and have presented an initial ontology specifying the class *Intrusion*.

We have produced a *target-centric* intrusion ontology that is based upon our *a posteriori* taxonomy. The ontology is represented in RDFS and instances of the ontology are represented in RDF. Out ontology is available at: http://security.cs.umbc.edu/Intrusion.rdfs. We have converted our ontology into N-Triples and have asserted it into a Prolog knowledge base and use Prolog to reason over our rules and assertions to determine the the cause of a given state, which Prolog deductively determines to be a Syn/Flood attack.

Although we have presented our target-centric ontology in terms of RDF, this does not preclude the use of DAML+OIL (DARPA Agent Mark Up Language and Ontology Interface Layer) [9]. DAML+OIL builds on RDF and RDF Schema, extending these languages to include richer modeling primitives.

Currently, we are in the process of identifying unique attributes and characteristics of the identified attack types.

References

[1] Julia Allen, Alan Christie, William Fithen, John McHugh, Jed Pickel, and Ed Stoner. State of

the Practice of Intrusion Detection Technologies. Technical Report 99tr028, Carnegie Mellon - Software Engineering Institute, 2000.

[2] M.S. Chen, J.-S. Park, and P. S. Yu. Efficient data mining for path traversal patterns. *IEEE Trans. Knowledge and Data Engineering*, 10(2):209–221, April 1998.

[3] W3C The World Wide Web Consortium. Resource description framework (rdf) model and syntax specification, February 1999.

[4] R. Cooley, B. Mobasher, and J. Srivastav. Web Mining: Information and pattern discovery on the World Wide Web. In *Proc. IEEE Intl. Conf. Tools with AI*, pages 558–567, Newport Beach, CA, 1997.

[5] D. Curry and H. Debar. Intrusion detection message exchange format data model and extensible markup language (xml)document type definition. draft-ietf-idwg-idmef-xml-07.txt, June 2002. expires December 19, 2002.

[6] Randall Davis, Howard Shrobe, and Peter Szolovits. What is knowledge representation? *AI Magazine*, 14(1):17 – 33, 1993.

[7] XML Schema Working Group. XML Schema. http://www.w3c.org/XML/Schema, 2000.

[8] Biswaroop Guha and Biswanath Mukherjee. Network Security via Reverse Engineering of TCP Code: Vulnerability Analysis and Proposed Solutions. In *IEEE Networks*, pages 40 – 48. IEEE, July/August 1997.

[9] J. Hendler. DARPA Agent Markup Language. http://www.daml.org, 2000.

[10] A. Joshi, C. Punyapu, and P. Karnam. Personalization and asynchronicity to support mobile web access. In *Proc. Workshop on Web Information and Data Management, 7^{th} Intl. Conf. on Information and Knowledge Management*, pages 17–20, November 1998.

[11] Lalana Kagal, Jeffrey Undercoffer, Anupam Joshi, and Tim Finin. Vigil: Enforcing Security in Ubiquitous Environments . In *Grace Hooper Celebration of Women in Computing 2002*, 2002.

[12] Richard A. Kemmerer and Giovanni Vigna. Intrusion detection: A brief history and overview. *Security and Privacy a Supplement to IEEE Computer Magazine*, pages 27 – 30, April 2002.

[13] Daphne Koller and Avi Pfeffer. Probabilistic Frame-Based Systems. In *Proceedings of the Fifteenth National Conference on Artifical Intelligence*, pages 580 – 587, Madison, Wisconsin, July 1998. AAAI.

[14] John McHugh. Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. *ACM Transactions on Information and System Security*, November 2000.

[15] O. Nasraoui, R. Krishnapuram, and A. Joshi. Relational clustering based on a new robust estimator with application to web mining. In *Proceedings of the North American Fuzzy Information Society Workshop International World Wide Web Conference*, pages 705–709, New York City, 1999.

[16] Peng Ning, Sushil Jajodia, and Xiaoyang Sean Wang. Abstraction-based intrusion in distributed environments. *ACM Transactions on Information and Systems Security*, 4(4):407 – 452, November 2001.

[17] Natalya F. Noy and Deborah L. McGuinnes. Ontology development 101: A guide to creating your fisrt ontology. Stanford University.

[18] O.Zaiane and J. Han. Webml: Querying the world-wide web for resources and knowledge. In *Proc. Workshop on Web Information and Data Management, 7^{th} Intl. Conf. on Information and Knowledge Management*, pages 9–12, 1998.

[19] G. Piatetsky-Shapiro. *Discovery, Analysis, and Presentation of Strong Rules*. AAAI Press, Menlo Park, CA, 1991.

[20] RDF. Resource description framework (rdf) schema specification, 1999.

[21] Jeffrey Undercoffer, Filip Perich, Andrej Cedilnik, Lalana Kagal, and Anupam Joshi. Centarusu2: A Secure Infrastructure for Service Discovery and Delivery in Pervasive Computing. *MONET: Special Issue on Security*, 2002.

[22] Jeffrey Undercoffer, Filip Perich, and Charles Nicholas. Shomar: An open architecture for distributed intrusion detection services. Technical Report TR CS-02-14, University of Maryland, Baltimore County, September 2002. available at: http://security.umbc.edu/pubs/shomar.pdf.

[23] W3C. Extensible markup language.

[24] Chris Welty. Towards a semantics for the web. Vassar College, 2000.

[25] M. Wood and M. Erlinger. Intrusion detection message exchange requirements. draft-ietf-idwg-requirements-08, August 2002.