J. Shanghai Jiaotong Univ. (Sci.), 2013, 18(5): 554-562
DOI: 10.1007/s12204-013-1439-5

Ontology-Based Model of Network and Computer Attacks for Security Assessment

GAO Jian-bo[1] (高建波), *ZHANG Bao-wen*[1*] (张保稳), *CHEN Xiao-hua*[2] (陈晓桦), *LUO Zheng*[3] (罗 铮)
(1. School of Information Security Engineering, Shanghai Jiaotong University, Shanghai 200240, China;
2. General Office, China Information Security Certification Center, Beijing 100020, China;
3. Information Classified Security Protection Evaluation Center, Third Institute of Ministry of Public Security of China, Shanghai 201204, China)

Abstract: With increased cyber attacks over years, information system security assessment becomes more and more important. This paper provides an ontology-based attack model, and then utilizes it to assess the information system security from attack angle. We categorize attacks into a taxonomy suitable for security assessment. The proposed taxonomy consists of five dimensions, which include attack impact, attack vector, attack target, vulnerability and defense. Afterwards we build an ontology according to the taxonomy. In the ontology, attack related concepts included in the five dimensions and relationships between them are formalized and analyzed in detail. We also populate our attack ontology with information from national vulnerability database (NVD) about the vulnerabilities, such as common vulnerabilities and exposures (CVE), common weakness enumeration (CWE), common vulnerability scoring system (CVSS), and common platform enumeration (CPE). Finally we propose an ontology-based framework for security assessment of network and computer systems, and describe the utilization of ontology in the security assessment and the method for evaluating attack effect on the system when it is under attack.
Key words: security assessment, formal analysis, taxonomy, ontology, attack effect
CLC number: TP 309 **Document code:** A

0 Introduction

With the rapid growth of the Internet, attacks are no longer limited in computers alone. They have created a global threat, causing great damages in individuals, communities and national security. Attacks are becoming more sophisticated, distributed and thus spread very fast, even in a matter of seconds. It is necessary to find and classify those attacks. So we need to know the attacks. And the first step in understanding attacks is to classify them into a taxonomy based on their characteristics. A taxonomy classifies attacks into well defined and easily understood categories. Such classification can be used for performing a systematic security assessment of a system. Much work on attacks taxonomy has been done recently. An introduction of them can be found in Ref. [1]. Howard and Longstaff[2] classified attacks based on the attack process. According to Howard's methodology, Alvarez and Petrovic[3] proposed a taxonomy of Web attacks suitable for efficient encoding. Hansman and Hunt[4] categorized network and computer attacks to improve security. And Simmons et al.[5] proposed an attack taxonomy to identify and defend against cyber attacks. We synthesize their work and propose our taxonomy of attacks suitable for security assessment.

Applications of semantic Web, knowledge base (KB) and ontologies in information system are popular recently. A knowledge base is a special kind of database for knowledge management. It provides a means to collect, share, search and utilize information. Here we concern description logic (DL) KB whose core of knowledge representation systems is description logic language. An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary. And ontology provides powerful constructs that include machine interpretable definitions of the concepts within a specific domain and the relations between them. As Raskin et al.[6] argued in their paper

Received date: 2012-09-07
Foundation item: the National Basic Research Program (973) of China (No. 2010CB731403), the Information Network Security Key Laboratory Open Project of the Ministry of Public Security of China (No. C09603), and the Shanghai Key Scientific and Technological Project (No. 11511504302)
∗**E-mail:** zhangbw@sjtu.edu.cn

that a security ontology could organize and systematize all the security phenomena such as computer attacks and support attack prediction, it has been applied in information system for different purposes. For example, Beitollahi and Deconinck[7] analyzed countermeasures against distributed denial of service (DDoS) attacks. Simmonds et al.[8] built a security attack ontology and made clearly understanding of the linkages between different components of a network security system. Wang et al.[9] built an ontology for vulnerability and proposed an ontological approach to computer system security. Ontology has been widely used in information security, and we believe it can be used in security assessment too.

We combine ontology with security assessment. A good taxonomy benefits the ontology construction and security assessment; ontology is machine readable and can make inferences, querying and consistence checking; ontology specifies semantic relationships between diverse concepts, and information related with the target can be gathered easily and quickly; ontology shares a common understanding of structured information, and can be shared among different agents to solve interoperability problems. In this paper, we classify attacks in five dimensions; they are attack impact, attack vector, attack target, vulnerability and defense. We reorganize contents in each dimension and try to make them up to date. We implement our ontology into Web ontology language (OWL), and make inferences by OWL reasoners and DL query languages. Finally, we give a framework for ontology-based security assessment, and illustrate the use of ontology for evaluating the attack effect on the system by a use case.

1 Related Work

1.1 Related Work in the Area of Taxonomies

Howard and Longstaff[2] described attacks as the following process. By using a tool, attackers exploited vulnerabilities in a target and attack for an unauthorized access. They organized a taxonomy of attacks, including five dimensions: attackers, tools used, access, targets chosen, and results achieved. The same idea was later used in Alvarez's classification of Web attacks[3]. They proposed a taxonomy with eight dimensions: entry point, vulnerability, service, action, input length, target, scope and privileges. We use some of Howard and Alvarez's ideas in the first and fourth dimensions of our taxonomy.

Hansman and Hunt[4] proposed four taxonomies of attacks based on four different dimensions of classification covering network and computer attacks. The four dimensions are: attack vector used to classify the attack, target of the attack, vulnerability base on common vulnerabilities exposures (CVE) or criteria from Howard's taxonomy, payload or effects involved. They mentioned the need of future research on correlation between attacks within the taxonomy and the utilization of KB. We try to use KB to analyze attacks by building an attack ontology, and describe relationships between different components in the five dimensions.

Venter and Eloff[10] provided a taxonomy of information security technologies. Their goal was to provide knowledge about security technologies. They divided security technologies into two categories: proactive and reactive. The proactive (reactive) part is subdivided into nine (seven) subcategories. Meanwhile, proactive and reactive technologies are classified by their level of interaction: network, host or application level. We use their taxonomy for reference in our classification of defense.

1.2 Related Work in the Area of Ontologies

Simmonds et al.[8] built a security attack ontology. They improved understanding of the relationship between different components included in a network security system. The classes of his network security attacks ontology are: access, actor, attack, impact, information, intangible, motive, outcome, systems administrator and threat. Main properties include: "assessing", "causing loss of", "gaining", etc. And their relations are presented as follows. An actor has his motive, and uses threat to implement attack; if the attack is succeeded, then the actor gains information and outcome, causing lose of the system. And systems administrator will report access, impact and outcome of the system.

Herzog et al.[11] used OWL to develop an ontology for information security. The ontology includes the classic components of risk assessment: assets, threats, countermeasures, vulnerabilities and their relations to each other. Figure 1 shows a simplified overview of the security ontology. An asset is connected to the concept vulnerability through "hasVulnerability" relation. An asset is threatened by threats and protected by countermeasures. A countermeasure is also an asset; it protects security goal and asset by defence strategy. We adopt some of Herzog's ideas in our proposed ontology, especially in the attack vector, defense and attack impact.

Wang et al.[9] presented an ontology focused on

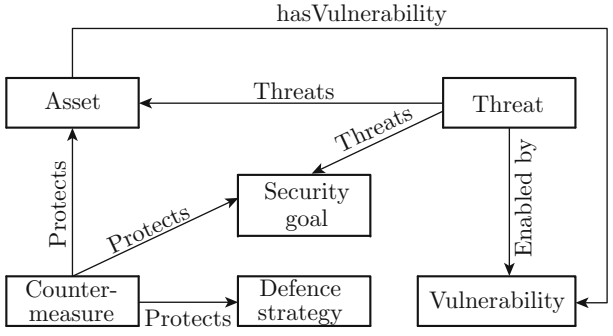

Fig. 1 Overview of the security ontology

vulnerability management which includes all the vulnerabilities published by national vulnerability database (NVD). Top level concepts of the ontology include vulnerability, IT_product, attacker, attack, consequence and countermeasure. Concretely, a vulnerability that exists in an IT_product can be exploited by an attacker, and the attacker conducts an attack with the objective of compromising the IT_product and causing a consequence. Countermeasures can be used to protect the IT_product through mitigation of the vulnerability.

2 Taxonomy of Attacks

The proposed taxonomy uses dimensions for classification. Dimensions are a way of classifying attacks as a whole. Figure 2 provides an overview of the taxonomy.

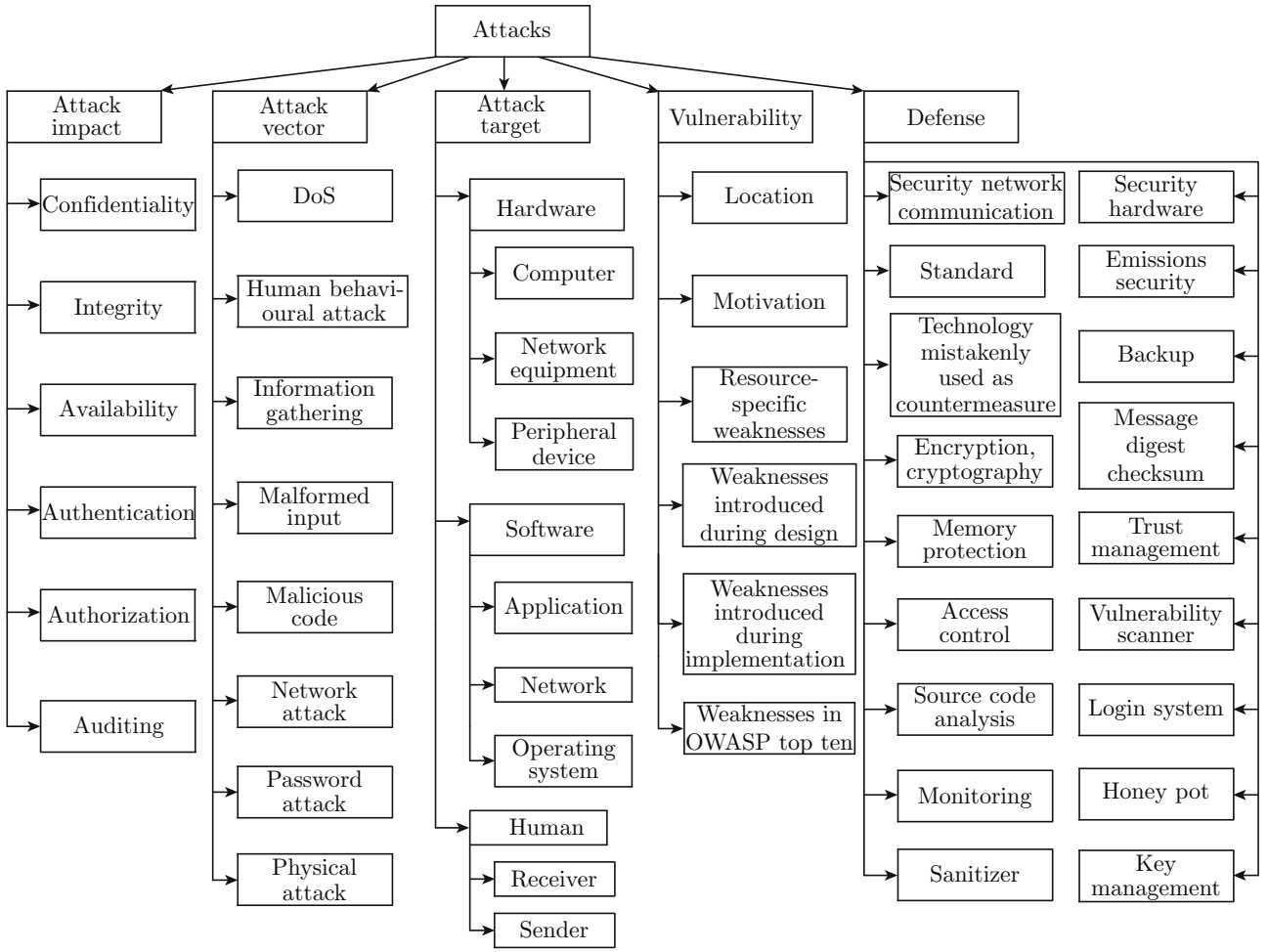

Fig. 2 Overview of attack taxonomy

Our taxonomy proposes five dimensions for attack classification, because these dimensions are major factors in attack effect evaluation. The attack impact depicts what kind of security property an attack will influence. While the attack vector describes how attacks reach their targets. The attack target is the object of an attack. The fourth dimension, vulnerability, means the vulnerabilities in the targets exploited by an attack. Meanwhile the purpose of security assessment is to defend attacks and assure security of the systems, so we include defense as one of the attack dimensions. And Herzog's classification of countermeasure is accepted in our defense dimension. With these dimensions, our taxonomy is suitable for security assessment.

2.1 Attack Impact

We use security property to model the impact of threats. Those properties include confidentiality, integrity, availability, authentication, authorization and auditing. The following are details about them.

Confidentiality is to prevent the information of systems from being disclosed or revealed to unauthorized entities. The information may be contained in all types of files and in database. Protection of confidentiality forms the cornerstone of information security in today's corporations. The most frequent attack against confidentiality is dictionary attack, scanning attack and password sniffing. Integrity is to prevent information of systems from interruption, interception, modification

and fabrication. In most cases, integrity is defined as the ability to guard against information modification. Availability is to guarantee that users are able to access to information and resources. The main attack against availability is denial of service (DoS) and DDoS. Without warning in advance, the computing and communication resources of a system can easily be exhausted by a DDoS attack in a short time. So the availability of the system is influenced. Authentication is the act of confirming the truth of an attribute of a datum or entity. In information security, this might include confirming the identity of a person, or assuring the computer program trustable. The common way to implement authentication is to use passwords or biometric recognition, like fingerprint recognition, iris recognition and voice recognition. While buffer overflows, SQL injection and cross site scripting are the most common authorization attacks. The auditing presents record of security relevant information to the system administrator, which can be used to evaluate the attacks.

2.2 Attack Vector

An attack vector is defined as a path by which an attacker can gain access to a host. It is the most important characteristic of the attack. This definition also includes vulnerabilities, as it may require several vulnerabilities to launch a successful attack. Sometimes it is hard to classify an attack into any one of attack vectors, such as blended attacks, because those attacks use several means to reach their targets. However, they can be expressed in ontology easily as we describe in Subsection 3.3. The following are major attack vectors we classify in network and computer systems.

DoS attacks An attack prevents legitimate users from accessing or using a host or network. They are divided into: stopping service and exhausting service. And then stopping service is subdivided into: process killing, system reconfiguration, process crashing, and malformed packet. Exhausting service is subdivided into: packet floods, forking process, and filling up the whole system.

Human behavioral attacks It is a damage caused by or related with human behavioral. They are subcategorized into: dumpster diving, inappropriate system use, social engineering, and unintentional file sharing.

Information gathering attacks They are attacks in which no physical or digital damage is carried out and no subversion occurs, but in which important information is gained by the attacker, possibly to be used in a further attack. They are subdivided into: eavesdropping, exploiting implementation, mapping, scanning attacks, security scanning, sniffing, and traffic analysis. And again eavesdropping is subdivided into: active eavesdropping, and passive eavesdropping.

Malformed input They are attacks caused by invalid input in order to cause buffer overflow or bypass the access control of the system. If successful, the threat may lead to the additional threats of malicious code or usurpation. It is subdivided into: buffer overflow, code injection and format string attack. Buffer overflow is subdivided into: Heap, Lib and Stack. Code injection is subdivided into: cross site scripting, PHP injection, SQL injection, and shell injection.

Malicious code It is a software program used for malicious attacks. It is subdivided into: Backdoor, Rootkit, Spyware, Trojans, Viruses, and Worms. Viruses are subdivided into: boot viruses, file viruses and network viruses. Keylogger is a subclass of the Spyware.

Network attacks They are attacks focused on attacking a network or the users on the network by manipulating network protocols, ranging from the datalink layer to the application layer. They are subdivided into: bypassing intended controls, distributed, negative acknowledgement (NAK) attacks, spoofing, Web application attacks, and password attacks. Among them, spoofing is subdivided into: IPAddressSpoofing, man in the middle, pishing, replay, and session hijacking. Web application attacks are subdivided into: cookie poisoning, database attacks, hidden file manipulation, and parameter tampering.

Password attacks They are attacks aimed at gaining a password. They are subdivided into: brute force, dictionary attack, and combination of above two.

Physical attacks They are attacks based on damaging physical components of a network or computer.

2.3 Attack Target

We adopt Hansman's idea of classification and add human into the attack target. First it is divided into hardware, software and human, and then hardware is subdivided into: computer, network equipment and peripheral devices. Software is subdivided into operating system, application, and network; operating system includes Windows family, Unix family and MacOS family; application includes server and user; network includes protocols. The idea of flavor and version of software is inherited. Human is divided into receiver and sender.

2.4 Vulnerability

We use the common weakness enumeration (CWE) as the taxonomy scheme of the vulnerability concept. CWE is a software assurance strategic initiative cosponsored by the National Cyber Security Division of the U.S. Department of Homeland Security. It has the following categories: location, motivation, resource-specific weaknesses, weaknesses introduced during design, weaknesses introduced during implementation, and weaknesses in the Open Web Application Security Project (OWASP) top ten. Location is subdivided into: code, configuration, and environment. Motivation is subdivided into: inadvertently introduced weakness and intentionally introduced weakness. Resource-specific weaknesses are subdivided into: weaknesses

that affect files or directories, weaknesses that affect memory, and weaknesses that affect system processes.

3 Ontology of Attacks

Our ontology is built on the base of the taxonomy in Section 2 and ontologies constructed by other researchers. Before we introduce our ontology of attacks, we need to make some related concepts clear. They are KB, DL, and OWL. They include components and application of ontology as well as methods for building it in this section.

3.1 KB, DL and OWL

A KB is a special kind of database for knowledge management. Its knowledge representation (KR) system is based on different logic languages, including propositional logic, first predicate logic, and description logic, etc. When description logic is used as the base of KR system, we get DL based KB. It comprises two components, TBox and ABox. The TBox introduces the "terminology", i.e., the vocabulary of an application domain, while the ABox contains assertions about named individuals in terms of this vocabulary.

Although there are many languages for building ontology, we choose OWL. On the one hand, OWL is based on XML which is popular in the Web, and this means that more tools are available for editing, handling, and documenting the ontologies. On the other hand, KR paradigm of OWL is DL which is useful in automatic classification and reasoning, and DL is very suitable for construction of ontology.

3.2 Ontology Overview

Main components of an ontology are:

(1) classes which represent concepts;

(2) relations which represent an association between concepts;

(3) functions, special case of relations in which the nth element of the relation is unique for the $n-1$ preceding elements;

(4) formal axioms, model sentences that are always true;

(5) instances which represent elements or individuals in an ontology.

An ontology can be used in many ways. For example, it can be used as a KB, a basis for developing software, and a tool for information search. Although currently there is no standard method for ontology development[12], there are several ways and rules for building it[13]. For example, we can reuse old ontology, which is available in the domain. Another way for building ontology is to use available taxonomy in the domain. Since we have classified the attacks and there are many researches about attack ontologies, we combine the two ways. We build our ontology according to the DL knowledge engineering methodology described by Baader et al.[14] and the design criteria for ontologies proposed by Gruber[15].

3.3 Our Ontology of Attacks

According to the taxonomy of attacks and ontologies mentioned above, we build an ontology of attacks. Our ontology is implemented into OWL. So concepts are implemented as classes, relations are implemented as properties, and axioms are implemented as restrictions. There are two types of properties (relations): object properties and datatype properties. Object properties are defined as relations between instances belonging to different classes, Datatype properties are relations between instances of classes and literals. Because the KB of OWL is DL, we use DL to describe the ontology. Concept in DL has the same meaning as class in OWL. Role describes binary relation between concepts. And individual represents instance of class.

Top level concepts of the ontology include attack impact, attack vector, target, vulnerability, and defense. Specifically, a vulnerability that exists in a target can be utilized by an attack, compromising the target and causing the lose of security property. Defense can be used to protect the target through mitigation of the vulnerability, and Herzog's countermeasure is adopted in our defense ontology. Following is the description of the concepts and their relationships in our ontology model.

3.3.1 *Attack Impact*

Attack impact ontology depicts the security properties of targets threatened by attacks. There are six main security properties: confidentiality, integrity, availability, authentication, authorization, and auditing. They are important properties in security assessing. The ultimate goal of attacks is not the attack target itself, but the information contained in or expressed by it, in other words, to destroy the security properties provided by the target. For example, DDoS attack exhausts or stops the resources of network or host, and prohibits the legitimate use of services. The attack target is host or network, but the goal is to prevent the legitimate user from using the resources provided by the target.

3.3.2 *Attack Vector*

Attack vector ontology describes concepts about main means by which the attack reaches its target and associates attack vector with concepts in all sub-ontologies. An attack threatens security property and target, while defense protects them. For example, buffer overflow threatens the integrity of data on volatile media, and boundary checking protects the integrity of data on volatile media, so boundary checking can protect integrity of data by thwarting buffer overflow attack. We use "enabledByVulnerability" to express an attack enabled by one or more vulnerabilities, use "hasAttackVector" to associate attack with attack vector, and use "ifSuccessfulLeadsToThreat" to depict relations between attack vectors.

After attacks as well as their hierarchical

classification and relations about attack concepts in the ontology have been defined, we need to define formal axioms. And they are usually embedded in concept or role (relation) definitions. For example, the definition of Rootkit: it is a malicious code, and if successful, the attack may cause other malicious code attack, and threaten integrity of host. The following is the DL description of above definition:

$$\begin{aligned}&\text{maliciousCode(Rootkit)}\cap\\&(\forall\ \text{ifSuccessfulLeadsToThreat.MaliciousCode})\cap\\&(\forall\ \text{threaten.(Host}\cap\text{Integrity)}).\end{aligned}$$

To complete our attack vector ontology, the last step is to populate it with individuals (instances). We also integrate evaluating index into the data type property of attack vector concept. For example, we consider SQL slammer as an instance of worm according to the definition: SQL slammer is a computer worm that caused a DoS on some Internet hosts and dramatically slowed down general Internet traffic. And if we study the technical detail of SQL slammer attack, we find if it succeeds, then it will cause UDP (user datagram protocol) flood DoS attack. This kind of relation can be described as

$$\begin{aligned}&\text{worm(SQL slammer)}\cap\\&(\forall\ \text{ifSuccessfulLeadsToThreat.(DoS}\cap\text{UDP)}).\end{aligned}$$

We give another example to illustrate how to express the blended attack in our ontology model. The Mitnick attack is multi-phased, consisting of DoS attack, TCP sequence number prediction and IP spoofing. We classify it as the instance of DoS, TrafficAnalysis, and IPAddressSpoofing at the same time. It is expressed as follows:

$$\begin{aligned}&\text{DoS(Mitnick)}\cap\text{TrafficAnalysis(Mitnick)}\cap\\&\text{IPAddressSpoofing(Mitnick)}.\end{aligned}$$

Sometimes, an attack instance has several nick names. SQL slammer has other names include Sapphire Worm, W32.SQLExp.Worm, DDOS.SQLP1434.A, SQL_HEL, W32/SQLSlammer and Helkern, and in ontologies these assertions can be made by making the different names as equivalent classes.

3.3.3 *Target*

According to the classification of target described above, we get a whole picture of targets and their hierarchical structure. Then we define some properties to depict relationships between target and classes in each subontologies. We use "hasVulnerability" to express a target has one or more vulnerabilities, use "threaten" to associate attack vector with target, use "protect" to depict relations between targets and defense, and use "reside" to sketch relations between different targets. For example, IIS6.0 is resided on the Windows server 2003, and Windows server 2003 resides on a home computer.

Then we populate it with instances. At first sight, every concept in Hansman's taxonomy of attack target is a class. So we follow the rules mentioned by Noy and Mc-Guinness[12] to decide whether a concept is an instance or a class, it depends on what the potential applications of the ontology are or what granularity should the ontology need. For example, if an attack threatens Windows XP, then Windows family is a class, and Windows XP is an instance. But if the attack only threatens Windows XP with service pack 2, then Windows XP is a class, and Windows XP SP2 is an instance.

3.3.4 *Vulnerability*

Vulnerability refers to security flaws, defects, or mistakes in software that can be used by a hacker to gain access to a system or a network. We use property "hasVulnerability" and "isVulnerabilityOf" to associate vulnerability with target and use "enabledByVulnerability" to depict vulnerability exploited by attacks. For example, SQL slammer utilizes CVE-2002-0649 to attack MS SQL server 2000, if the attack succeeds, it will cause UDP packet flooding DoS and lose of availability of network. Such description can be formalized as

$$\begin{aligned}&\text{worm(SQL slammer)}\cap\\&(\forall\ \text{enabledByVulnerability.CVE-2002-0649})\cap\\&(\forall\ \text{threaten.(MS SQL server 2000)}\cap\\&(\forall\ \text{ifSuccessfulLeadsToThreat.(DoS}\cap\text{UDP)}\cap\\&\forall\ \text{threaten.(network}\cap\text{availability)}).\end{aligned}$$

We use CWE as the base to build our vulnerability ontology, and populate it with CVE. Because NVD also integrates common vulnerability scoring system (CVSS) as impact metrics for CVE vulnerabilities, so the CVSS score for each CVE vulnerability in NVD is included in our ontology.

4 Framework for Security Assessment

Traditional security assessment method includes vulnerabilities discovery and risk assessment. In this paper, we propose an ontology-based framework to evaluate attack effect. Through the evaluation of attack effect, we can find out the discrepancy of the system's performance before and after attack happens. The following is introduction of assessing model and the method for assessment.

4.1 Our Model for Security Assessment

Figure 3 shows the ontology-based framework for security assessment. The left side of the framework is the ontology we built. Evaluating index is datatype property of arbitrary instance of attack target concept. The right side is security assessment model; from bottom

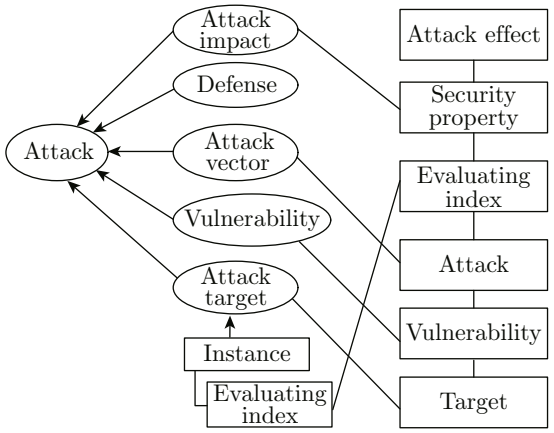

Fig. 3 Ontology-based security assessment framework

to top it is been divided into six layers: target, vulnerability, attack, evaluating index, security property, and attack effect. Except for the attack effect layer, other layers have relationships with one of the ontology classes or properties of instances. The first step to evaluate the security of the target is to use vulnerability scanning tools to find vulnerabilities of the target, and then use ontology to get what attacks will be enabled by the vulnerabilities, after that evaluating index reflecting the attack effect is obtained by querying the ontology. After we get the evaluating index, analytic hierarchy process (AHP)[16] method is used to evaluate the attack effect. Figure 4 describes the details of the AHP model for security assessment.

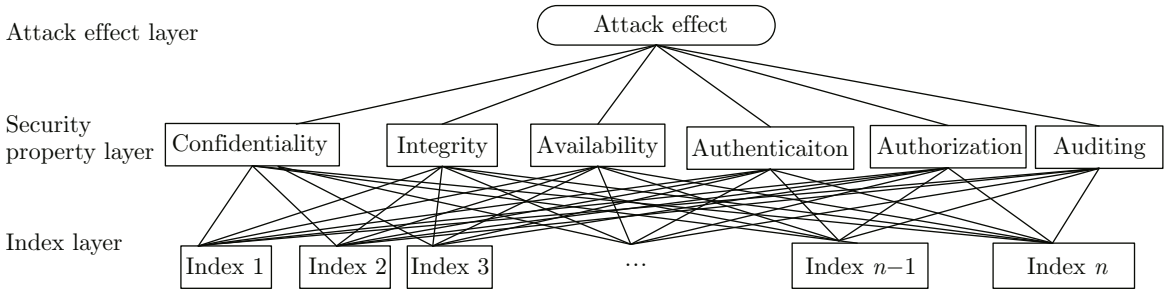

Fig. 4 Attack effect assessing model for a node

The model has been divided into 3 layers. The top layer means result of attack effect assessment. The result of assessment is a number between 0 and 1. When the attack effect is 0, it means attacks have no or subtle influence on the target; when it is 1, that means a strong influence on the target. The middle layer is the "attack impact" mentioned in our ontology, while the third layer "index layer" represents the factors of the assessment. It can be extracted from various security databases (NVD, snort rules database, etc.) through data mining; common evaluating index includes: CPU consumption, memory consumption, network delay, network delay jitter, network or node throughput, packet-drop, update cycle of routing table, system server response time, and system recover time, etc.

We use AHP[16] method to measure attack effect. It can be calculated by

$$z = \sum_{i=1}^{n} w_i I_i, \tag{1}$$

where z represents attack effect on the system, w_i represents the weight of each evaluating index, I_i is normalized value of evaluating index in a measuring, and n is the number of evaluating indexes.

4.2 Feasibility Analysis of Our Approach

Our approach is divided into five major phases identified throughout the evaluating process, i.e., establishment of security ontology for the security assessment, vulnerability scanning, querying and reasoning by the ontology, assessing the weight of each security property and evaluating index by AHP method, and measuring value of evaluating index.

We discuss now viability of each phase in the approach.

Directed by Noy' guide and according to the security ontology built, we get a whole picture of the ontology for security assessment. All concepts in five dimensions of attack ontology can be got in various database and taxonomies of related researches. For example, we collect vulnerability information in CVE, CWE and CVSS. We get target information in common attack pattern enumeration and classification (CAPEC). We extract evaluating index in NVD, or snort rules database.

Meanwhile many vulnerability scanning tools [17] can be used to determine what threats and vulnerabilities exist in computer and network systems; many methodologies for identifying vulnerabilities in the systems exist. Examples of such tools are automated vulnerability detection system (AVDS), Nessus and core impact, etc.

Once an ontology for the research area is built, it can be used for reasoning and querying as Herzog described.

And ontologies have the ability to reason and query by letting a reasoner (FaCT, Racer, Pellet, etc.) infer subsumption relationships between concepts.

Through the AHP method, the weights of security property and evaluating index can be calculated. The AHP approach pairwise compares factors (in our experiment, they are elements' weights with respect to different security properties) influencing the ultimate result, and calculates subjective evaluation of elements' weights based on experts' judgments and opinion. After we get the weights of evaluating indexes and security property, attack effect can be calculated by Eq. (1).

Measuring of evaluating index was studied in earlier researches. For example, Hu et al.[18] designed a model of DoS attack effect evaluation. It shows the viability of calculating the DoS attack effect. In fact the scheme of evaluation could be expanded to a more general attack effect evaluation system.

4.3 Case Study

The following example illustrates the utility of our ontology within a security assessment system and how to use AHP method to calculate the attack effect.

Suppose we are interested in assessing security of a personal computer with IIS6.0 and Windows 2003 SP2 operating system installed on it. To achieve this, we use vulnerability scanning tools to find the vulnerabilities in the system. Then we get several vulnerabilities, we choose vulnerabilities discovered in recent years with high CVSS score, such as CVE-2011-3414, CVE-2010-1256 and CVE-2011-0654. After that we query the ontology what attack will be enabled by the vulnerabilities mentioned above:

(enabledByVulnerability some CVE-2011-3414) or
(enabledByVulnerability some CVE-2010-1256) or
(enabledByVulnerability some CVE-2011-0654).

So we obtain attack vector: BufferOverflow, CodeInjection and DoS; then we associate the attacks with related evaluating index: CPU consumption, memory consumption, and system server response time. After that, we use AHP method to assess the attack effect on the system. The AHP model is implemented using the tool "SuperDecisions". In the example, the security property layer includes two elements: availability and authorization (the properties are threatened most by the attacks). Suppose the weights of availability and authorization evaluated by the experts are 0.6 and 0.4, respectively. Next the elements in evaluating index are pairwise with respect to each of the security properties. The judgments and the derived priorities are shown in Tables 1—3.

After we get the weights of elements of evaluating index, Eq. (1) is used to calculate the attack effect on the system:

$$z = 0.211\,4I_1 + 0.205\,7I_2 + 0.582\,9I_3.$$

Table 1 Element's weights with respect to availability

Weight			Priority/%
CPU consumption	Memory consumption	System server response time	
1	2	1/2	28.57
1/2	1	1/4	14.29
2	4	1	57.14

Table 2 Element's weights with respect to authorization

Weight			Priority/%
CPU consumption	Memory consumption	System server response time	
1	1/3	1/6	10
3	1	1/2	30
6	2	1	60

Table 3 Overall priorities for elements of evaluating index

Evaluating index	Priority/%		Overall priority/%
	Availability	Authorization	
CPU consumption	28.57	10	21.14
Memory consumption	14.29	30	20.57
System server response time	57.14	60	58.29

IT security assessment is an explicit study to locate IT security vulnerabilities and risks. The goal of a security assessment is to ensure that necessary security controls are integrated into the design and implementation of a project. From the process of calculating attack effect, we believe that the evaluation of attack effect benefits the security assessment in the following two ways.

Firstly, because our method uses vulnerability scanners to find vulnerabilities in the system, and then utilizes our ontology to reason relationships between vulnerabilities and attacks, thus the two dimensions in five of attacks interconnect with each other. The use of ontology can categorize vulnerabilities found in the system, so we can find quickly the tree of vulnerabilities.

Secondly, the attack effect reflects the impact of exploitation of a potential weakness of the system. The more attack effect on the security property of the system, the more impact caused by exploitation of the vulnerability. Risk calculation is given as

$$R = \sum_i P_i D_i, \tag{2}$$

where R is the system risk, P_i means the probability of

occurrence of ith weakness, and D_i means the damage caused by the ith weakness. So the evaluation of attack effect is helpful to risk assessment.

5 Conclusion

In this paper, we propose an ontology-based framework for evaluating attack effect. More specifically, the framework benefits security assessment of the system by measuring the attack effect on it. The ontology is the basis of our framework, which provides security information needed in the whole measuring process. While AHP method is used to calculate the weights of security property and evaluating index which are important factors in security assessment. Differing from the traditional security assessment method: vulnerabilities discovery and risk assessment, our approach assesses the security of systems by evaluating the attack effect on the system. Attack effect is the changes of the system's performance before and after attack. The modification of the system's performance is more suitable for reflecting security status. The better the attack effect is, the worse security the system has. In addition, through feasibility analysis of our approach, the proposed framework is viable.

References

[1] Igure V M, Willams R D. Taxonomies of attacks and vulnerabilities in computer systems [J]. *IEEE Communications Surveys*, 2008, **10**(1): 6-19.

[2] Howard J D, Longstaff T A. A common language for computer security incidents [R]. California: Sandia National Laboratories, 1998.

[3] Alvarez G, Petrovic S. A taxonomy of Web attacks suitable for efficient encoding [J]. *Computer & Security*, 2003, **22**(5): 435-449.

[4] Hansman S, Hunt R. A taxonomy of network and computer attacks [J]. *Computers & Security*, 2005, **24**(1): 31-43.

[5] Simmons C, Ellis C, Shiva S, et al. AVOIDIT: A cyber attack taxonomy [R]. Memphis: University of Memphis, 2009.

[6] Raskin V, Hempelmann C F, Triezenberg K E, et al. Ontology in information security: A useful theoretical foundation and methodological tool [C]//*Proceedings of the 2001 Workshop on New Security Paradigms*. New York: NSPW, 2001: 53-59.

[7] Beitollahi H, Deconinck G. Analyzing well-known countermeasures against distributed denial of service attacks [J]. *Computer Communications*, 2012, **35**(11): 1312-1332.

[8] Simmonds A, Sandilands P, Ekert L V. An ontology for network security attacks [C]//*Proceedings of Second Asian Applied Computing Conference*. Kathmandu, Nepal: AACC, 2004: 317-323.

[9] Wang J A, Guo M M, Camargo J. An ontological approach to computer system security [J]. *Information Security Journal: A Global Perspective*, 2010, **19**(2): 61-73.

[10] Venter H S, Eloff J H P. A taxonomy for information security technologies [J]. *Computers & Security*, 2003, **22**(4): 299-307.

[11] Herzog A, Shahmehri N, Duma C. An ontology of information security [J]. *International Journal of Information Security and Privacy*, 2007, **1**(4): 1-23.

[12] Noy N F, Mc-Guinness D L. Ontology development 101: A guide to creating your first ontology [R]. Stanford: Stanford Knowledge Systems Laboratory, 2001.

[13] Blanco C, Lasheras J, Fernandez-Medina E. Basis for an integrated security ontology according to a systematic review of existing proposals [J]. *Computers Standards & Interfaces*, 2011, **33**(4): 372-388.

[14] Baader F, Calvanese D, Mc-Guinness D L, et al. Description logic handbook: Theory, implementation and application [M]. Cambridge, UK: Cambridge University Press, 2003.

[15] Gruber T. Towards principles for the design of ontologies used for knowledge sharing [J]. *International Journal of Human-Computer Studies*, 1995, **43**(5-6): 907-928.

[16] Vargas L G, Dougherty J J. The analytic hierarchy process and multicriterion decision making [J]. *American Journal of Mathematical and Management Sciences*, 1982, **19**(1): 59-92.

[17] Holm H. Performance of automated network vulnerability scanning at remediating security issues [J]. *Computers & Security*, 2012, **31**(2): 164-175.

[18] Hu Ying, Xian Ming, Xiao Shun-ping. Design of a DoS attack effect evaluation system [J]. *Computer Engineering & Science*, 2005, **27**(2): 15-22 (in Chinese).