Transactions D:
Computer Science & Engineering and
Electrical Engineering
Vol. 17, No. 2, pp. 133–148
© Sharif University of Technology, December 2010

Ontological Classification of Network Denial of Service Attacks: Basis for a Unified Detection Framework

A. Varshovi[1] and B. Sadeghiyan[1,*]

Abstract. *In this paper we introduce the notion of a detection framework to facilitate the reasoning and cooperation process of detection and response systems. The presented framework defines four dimensions as requirements to be satisfied: "What to detect", "Where to inspect", "How to decide", and "How to alert". The first dimension tries to unify the understanding of the problem between systems. The second will introduce detection features and parameters. The third dimension exactly states how intelligent systems or expert knowledge should be deployed, while the task of the fourth is to unify the alert and message exchange format. To address the "What to detect" aspect of our framework, we have considered a network denial of service and have presented an ontology which relates three taxonomies of DoS attacks, each from a different point of view: Attack Consequence, Attack Location and Attack Scenario. For scenario based taxonomy, we present a decision tree-like structure, which can be used as a base for attack detection. All these taxonomies are then related to each other in an ontology. An implementation of this ontology using Web Ontology Language (OWL) might help IETF's IDMEF to construct a base for a more accurate alert correlation.*

Keywords: *Availability; Denial of service; Detection framework; Ontology; Taxonomy.*

INTRODUCTION

Availability is defined as one of the basic components of computer security. According to a definition by Bishop [1], availability refers to the ability to use the information or resource desired. Because an unavailable system is at least as bad as no system at all, subverting the availability of a system has always been one of the major goals of attackers and intruders. A specific type of malicious activity, called a "Denial of Service" (DoS) attack, threatens the availability of systems and a so called division, "Network DoS", seriously targets services in today computer networks. Although there are several definitions of the problem, they all agree that DoS is an activity with the goal of preventing the access of legitimate users to a specific service or set of services. Recent DoS activities are performed in a distributed manner to multiply attack power and hide the attacker's identity.

Although most DoS attacks are based on very simple logic, there is no perfect defense against many types of these attacks. The heart of the problem lies in the characteristics of the Internet protocols. These protocols have been designed to meet performance and functionality issues and leave other important aspects, like security, to end parties based on the end-to-end principle. As a result, a subverted party would exploit a protocol to initiate a DoS attack toward other parties. Recently, prevention of DoS attacks through protocol redesign or active mechanisms based on game theory have been proposed [2,3]. The main question to ask about these category of solutions is the overhead and the effect on functionality regarding implementation in a real environment.

Conventionally, the task of attack detection is the responsibility of "Intrusion Detection Systems" (IDS), which may be stand alone systems and appliances or integrated sub-systems of secure gateways, firewalls, or even network devices such as routers and switches. Intrusion detection systems are designed based on two

[1] *Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, P.O. Box 15914, Iran.*

[*] *Corresponding author. E-mail: basadegh@ce.aut.ac.ir*

Received 27 September 2009; received in revised form 29 May 2010; accepted 7 June 2010

major approaches: anomaly and misuse detection. “Anomaly Detection” systems use a model of the normal behavior of the environment and interpret deviations of this intended model as an attack. In contrast, “Misuse Detection systems” essentially model misuse behavior. They usually define misuse patterns or signatures of attack events and try to find a match between them and current events of the environment. Each of these approaches has its own benefits and drawbacks, but both of them face serious problems in the detection of DoS attacks. The majority of DoS attacks use normal network traffic and this makes it really hard, sometimes impossible to distinguish between attacks and normal events. So, neither a normal profile nor a misuse pattern can be ideally set as a base to detect these kinds of attack. Misuse detection systems have a chance to detect certain network traffic generated by known attack tools. On the other hand, anomaly detection may detect violation of network traffic limitations modeled through specific statistical thresholds. It is considered that a “Hybrid approach”, combining both misuse and anomaly approaches, to be a better choice.

The knowledge behind the detection of DoS attacks is obtained from two sources: “Expert Knowledge” and “Intelligent Decision Support Systems”. Expert knowledge is based on observing experienced attacks and the study of published material that is presented via general statements or sometimes detailed signatures. In the other approach, intelligent systems analyze collected sensory data and try to build the desired detection models, hence, making them more sophisticated and biased toward the data observed. In other words, each solution is limited to its special environment and to the view of its designers. This concludes to a heterogeneous selection of detection features, classification of attacks and response mechanisms. As another issue, the ambiguity and diversity behind the understanding of the problem makes the correlation of different systems and building a cooperation framework difficult. This brings the necessity of a unified Detection Framework, to the front, the lack of which seems to be the main reason for this divergence.

In this paper, we present the notion of a Detection Framework to address the problems explained above. As one of the major components of the framework we propose ontology of DoS attacks. The proposed ontology includes a scenario-based taxonomy. This taxonomy is an effort to distinguish between various scenarios that may be deployed in order to launch an attack. Meanwhile, it provides a scheme from which corresponding decision support models may be derived. It is also obvious that this agreement on the problem can significantly improve the design of correlation mechanisms. The idea behind this research can be extended to cover other types of attack and is the subject of our future work.

Therefore, this paper does not propose or advocate any specific detection or defense mechanism. Instead, we present a framework to address four main problems in the detection of and defense against DoS attacks:

1. To build a classification scheme that guides intelligent or expert knowledge-based detection systems.
2. To enable detection systems to exchange information through a unified view of the problem.
3. To facilitate the process of alert correlation by unifying an understanding of the problem in a heterogeneous environment.
4. To present an ontology of DoS attacks and related scenarios which covers known and unknown attacks.

The rest of the paper is organized as follows. First, we summarize the related works. Important definitions and assumptions as the base of this study are explained before introducing the notion of detection framework. The task of ontology and taxonomy in a detection framework is described after, and three taxonomies of DoS attacks are presented. Introducing our ontology of DoS attacks and the problem of integrating all concepts of this domain into a single taxonomy is the subject of the next section. As an illustration, we consider the case of a p attack and explain how the framework assists the detection process. We also provide information about our future work to extend this framework while the last section concludes the paper.

RELATED WORKS

Although the concept of a unified detection framework has not been stated before, there are several pieces of research that target different dimensions of our framework.

The IDMEF data model [4] was designed by IETF to provide a standard representation of alerts in an unambiguous fashion, permitting the relationship between simple and complex alerts to be described. This data model can be one of the components of a detection framework in order to make it possible for systems to share their captured events. But without any agreement about the the subject itself, the exchanged information might be confusing or useless.

Recent RFC 4732 [5] on “Internet Denial-of-Service Considerations” is an attempt to unify basic definitions and concepts behind the problem. This RFC does not introduce a practical approach to build a knowledge-base for unified understanding and detection of this type of attack.

The framework presented in [6] on a flexible intrusion detection, and response framework for active networks is an effort to build a response scheme for detection and response systems. The work by Hess et al. is beyond the scope of this paper, which just focuses on the detection process. The presented concept can be merged with our work in order to produce a detection-response framework. Introducing this conjunction can be a subject for future research.

The aforementioned works are steps towards satisfying the need for a unified framework. The IDMEF only targets the way that systems generate an alert. The RFC 4732 talks mainly about definitions. The latter, by Hess et al., focuses on the response framework, not how the systems should reason or understand the problem. Kemmerer and Vigna [7] comment that "IDMEF defines the format of alerts and an alert exchange protocol. Additional effort is needed to provide a common ontology that allows sensors to agree on what they see. Without this common way of describing the involved entities, sensors will continue to disagree when detecting the same intrusion". The same idea has been stated in [8] as: "... that the intrusion detection community would benefit greatly from a shared alert model that extends the current IDMEF work with semantic information and a common attack naming scheme. An ontology for intrusions is a prerequisite for true interoperability between different IDSs."

The need for a detection framework has been stated briefly in several pieces of researches. Allen et al. discuss the necessity of establishing a commonly accepted framework for intrusion detection [9]. Cheung, Lindqvist and Fong in [10] discuss that "knowledge representing attack scenarios needs to be modeled, preferably in a way that is decoupled from the specifics of a particular correlation technology". We address this specific requirement in our ontology when we propose our scenario based taxonomy of DoS.

Moreover, scalable intrusion detection design and cooperative defense strategies, like the one presented in [11], consist of a stage at which each defense node detects attacks locally using a variety of existing IDS tools. It is obvious that a heterogeneous understanding of attacks makes it impossible to build this cooperation framework.

We believe that the ontology proposed in [12] is the closest to our idea of a detection framework; however, the presented target centric ontology talks only about general aspects of intrusion and does not specify a certain view of them. As a result, in the case of a detected attack, the ontology would comprehensively say what the relationship of the attack to other components of the problem domain (e.g. system components and consequence of attack) would be. It is obvious that as long as we are using ontologies with mixed types of relation, the conclusion is not a precise definition of what the attack itself is. Moreover, this ontology can just be deployed after detection of an event. This work mainly considers the exchange of knowledge between systems, which is only one of the roles of a detection framework.

Besides the above mentioned, there are some other works which do not directly try to propose a framework. They analyze the problem in such a way that leads to a high level classification of the DoS attacks. We give a summary on this line of research in the following sections of the paper. Some of them propose taxonomies of DoS attacks and others define the DoS more formally.

There is no doubt that a hierarchical structure, such as a taxonomy or its more detailed sibling, a decision tree, has the ability to define an attack precisely with several levels of hierarchy. Such a hierarchical structure not only results in more accurate detection but also the means to select detection features. The variety of DoS classification views or schemes makes it nearly impossible to develop a single taxonomy that covers them all. We will take a look at the presented taxonomies relating DoS attacks in subsequent sections.

DEFINITIONS AND ASSUMPTIONS

The first step to defend against DoS is to define the DoS itself. Besides several definitions of DoS attacks, a variety of studies has been presented, each with its special view of the problem. This has led to various understandings of the characteristics of DoS attacks and the confusion of how to defend against them. In other words, all the research toward building a defense system against DoS attacks lacks agreement on a common yet effective view of DoS. In this section, we express our definitions and assumptions, besides the problem to be addressed in the next sections.

Definitions

Here, we define DoS and Network DoS as two basics of our discussion. So, first we review several definitions for DoS.

Yu and Gligor [13] define that "users with high priorities for resource use are authorized to deny a service to lower priority users". Based on this definition, DoS may not always be a malicious activity. For example, an administrator of a system can legitimately deny access of users to services provided by that system.

Amoroso [14] states that "A denial of service attack is assumed to have taken place when access to a computer or network resource is intentionally blocked or degraded as a result of malicious actions taken by another user". He explains this definition as follows: "the DoS [Denial of Service] threat will be defined to occur when a service associated with a maximum waiting time (denoted MWT) is requested by a user at time 't' and is not provided to that user by the time (t+ MWT)".

According to [15], the definitions given by Gligor and Amoroso are based on the concept of access control, and using access control policies to defeat an attack is not a straightforward approach.

The CERT coordination center defines a DoS attack as "the prevention of authorized access to a system resource or the delaying of system operations and functions" [16].

Bishop defines DoS as a threat, stating that it is a long-term inhibition of service, a form of usurpation, although it is often used with other mechanisms to deceive [1].

The intention behind an attack is the subject of some discussion. According to [1], the aspect of availability that is relevant to security is when someone may deliberately arrange to deny access to data or to a service by making it unavailable. In contrast, Howard believes that intentional or unintentional assaults on availability can be assumed as DoS attacks [17].

We define the DoS as "any intentional or unintentional action or assault on the availability of a service in order to deny access of any number of users to that service or any other dependent services". We also explain other requirements of this definition in the assumptions section that follows. We define the Network DoS to be "any DoS attack launched through network traffic by the use of network protocols, with the target to be of any level".

Assumptions

The following assumptions are held through our discussions:

1. We assume that DoS can be the primary or sometimes non-primary goal of an attack.
2. The same as [1], we assume that the cause and result of attacks are important, not the intention underlying them. If delay or denial of a service compromises the security of the system, or when it is part of a sequence of events leading to compromising a system, then we consider it an attempt to breach the security of the system. The attempt may not be deliberate; indeed, it may be the result of environmental characteristics rather than the specific actions of an attacker.
3. We suppose that there is a considerable amount of resources available at servers. This makes it impossible for a single normal attacking machine to exhaust target services. While logically it is not impossible to launch this kind of attack in today's Internet, this does not really result in a DoS attack. So, we limit uni-source attacks to those which deploy only a small number of specially crafted packets, and does not rely on the volume of traffic as a means of denial.
4. Following the model presented by Millen [18], we consider a system as a hierarchy of services, supporting services and atomic services. We explain this matter under the taxonomy of service components in the coming sections.
5. We assume that detection systems use any type of decision support model, ranging from human expert knowledge to intelligent learning systems.

DETECTION FRAMEWORK

As Figure 1 states, we propose that a detection framework should unify the following requirements, which we call "Dimensions" of a framework. All research behind the detection of DoS and especially Network DoS covers a subset of these dimensions. In other words, the presented unified framework brings together all necessary requirements of attack detection in four defined dimensions.

1. What to Detect
   - A formal description of events.
2. Where to inspect
   - A set of features and parameters.
3. How to decide
   - A decision support model.
4. How to alert
   - An alerting data format.

What to Detect

This dimension of the framework is an attempt to unify an understanding of what a DoS attack is. The

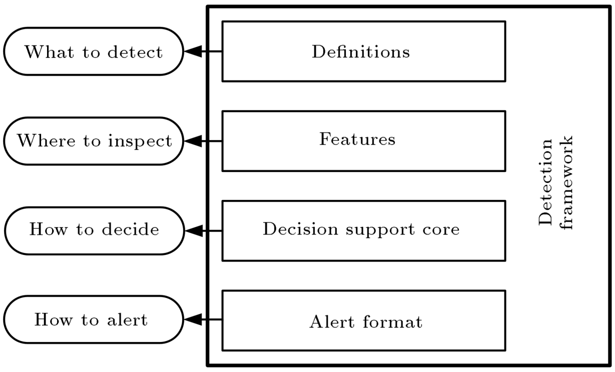


Figure 1. Detection framework.

intended concepts and descriptions to be presented are coded in any form of knowledge representation. In the following sections, we propose an ontology containing several levels of hierarchy, which we believe presents the DoS in a comprehensive manner. Furthermore, the ontology enables detection systems to contribute to the field through their observations. In other words, if an unknown event is categorized as an attack by a system, it can be coded with properties and concepts of the ontology. The event can then be propagated to other detection systems that use the same framework. This highly facilitates the cooperation of these systems.

Where to Inspect

It is of utmost importance in the task of detection to be able to recognize, out of a number of facts, which ones are incidental and which are vital. Specifying a set of features is critical to facilitate the detection of events. In this dimension of the framework, these features should be introduced based on the ontology or any representation provided in previous dimension. The task of this feature selection mechanism is not to strictly define the needed features. It is intended to introduce sufficient concepts, which enable the system to select the most effective features between numerous ones provided by the environment.

How to Decide

Up to this level, we have an understanding of the problem, which in the best case categorizes the events. Furthermore, it presents a set of parameters assigned to each category or class of these events. The question remaining is how do we use these parameters and features to detect defined events? This is where expert systems, intelligent systems, or any decision support model or pattern recognition method are deployed. Note how the framework specifies the exact task of these systems and brings together the domain knowledge and the power of intelligent systems. In other words, we have provided these systems with our knowledge of DoS attacks.

How to Alert

The alerting format addresses the problem of knowledge exchange between detection systems or sensors. As explained before, the work done by IETF, called IDMEF, falls into this dimension of the detection framework.

In this paper, we completely address the first dimension of the framework with an ontology of network DoS attacks.

ONTOLOGY VS. TAXONOMY

As mentioned before, we present an ontology of DoS attacks as a means to satisfy the requirement of “What to detect” dimension of our detection framework. The goal of the presented ontology is not only to provide a formal definition of the problem, but also to present a scheme to be used when a system tries to detect and classify events.

In this section, we first summarize several definitions of taxonomy, ontology, and how they are related to each other. We present several taxonomies of DoS attacks, based on various views of these attacks. The selected views are those which describe the problem at the target side. The importance of this target side definition of DoS concepts is encouraged by sensor placement of current detection systems. The majority of these sensors like IDS, firewalls, application level sensors or so on are located close to the target. It is a reasonable choice as detection features are observable and measurable at the target side or on behalf of the target. We then discuss why it is complicated to combine all these views in a single taxonomy and propose our ontology of DoS attacks to address this problem.

After presenting three taxonomies, we define certain properties to relate them in an ontology. We present the problem of relating these taxonomies in a hierarchical manner via a single taxonomy. This raises the problem of uninterpretable is-a or kind-of relationships, which we will discuss further.

Ontology

An ontology integrates a specific domain of knowledge in a common vocabulary, and provides basic concepts in the domain and relations among them. This is not only useful for others who need to share information, but also for intelligent systems through machine-interpretable definitions. According to [19], there are several reasons why someone would want to develop an ontology:

- 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 operational knowledge.
- To analyze domain knowledge.

An ontology consists of “concepts” in the domain of discourse, which are sometimes called “classes”. Various features and attributes to describe a concept are coded in the ontology using “properties”, which are called sometimes, “slots or relations”. It is important to arrange concepts of an ontology in taxonomic order containing classes and subclasses as much as possible. Subclasses represent concepts that are more specific than their parent class. The next step is to relate generated taxonomies using properties to form a complete ontology. We keep this approach as a road map during the development of our ontology in the following sections.

Taxonomy

According to Marriam-Webster, a taxonomy is a classification system where the classification scheme conforms to a systematic arrangement into groups or categories, according to established criteria. The Oxford Dictionary defines taxonomy to be a branch of science concerned with classification and a scheme of classification.

Borrowed from [12], Glass and Vessy contend that taxonomies provide a set of unifying constructs, so that the area of interest can be systematically described, and aspects of relevance may be interpreted. The overarching goal of any taxonomy, therefore, is to supply some predictive value during analysis of an unknown specimen, while the classifications within the taxonomy offer an explanatory value. Simpson also defines a taxonomic character as a feature, attribute or characteristic that is divisible into at least two contrasting states, and used for constructing classifications. He further states that taxonomic characters should be observable from the object in question [12].

A taxonomy is contained within an ontology. In other words, as stated in [19], an ontology at its deepest level, subsumes a taxonomy. Taxonomy brings the essence of the object, while ontology provides sufficient meta-data to define relationships. As discussed by Landwehr et al. [20], a taxonomy is not simply a neutral structure for categorizing specimens. It implicitly embodies a theory of the universe from which those specimens are drawn. It defines what data are to be recorded and how like and unlike specimens are to be distinguished. In creating a taxonomy of computer program security flaws, we create a theory of such flaws, and if we seek answers to particular questions from a collection of flaw instances, we must organize the taxonomy accordingly.

Taxonomies of DoS Attacks

Mirkovic and Reiher have presented a taxonomy of DDoS attacks and defense mechanisms [21]. The presented attack taxonomy consists of several views to classify DDoS species, and that is the point which makes it remarkable compared to other proposed taxonomies. The taxonomy has been claimed to be complete in covering those attacks that have not yet appeared, but realistic potential threats that would affect current defense mechanisms.

Several aspects of Mirkovic and Reiher's taxonomy should be discussed here. First, the taxonomy does not combine the presented views. In other words, it is not completely possible to combine every two views in order to categorize DoS attacks. Some of the views are victim side, like impact and rate dynamics, while some others are related to the attack nature or the characteristics of attack networks. Second, the presented taxonomy is a means by which to summarize and classify knowledge about DDoS attacks, but the level of abstraction and views to the problem makes it unsuitable for setting as a classification scheme for detection systems. Third, the claimed prediction capability of the taxonomy is the subject of only one of the categories, "En Route Spoofed Source Address", and this prediction capability has its roots in the talent of the authors, not the completeness of the taxonomy.

Douligeris and Mitrokotsa have presented a modified and reduced version of the Mirkovic and Reiher's taxonomy [22]. The only difference is in a branch called "Classification by exploited vulnerability" where flood attack amplification attack, protocol exploit attack and malformed packet attack have been classified. In our point of view, these categories are not based on vulnerabilities, but on the characteristics of attacks.

The taxonomy by Specht and Lee [23] is unique in the sense that it not only classifies the attacks based on high level characteristics, but also observes some levels of detail, like protocols used and some information regarding the scenario of the attack. The taxonomy lacks a complete domain study and a systematic approach to address the problem. Just a few attacks have been presented, while several more have not been considered. Again, the proposed taxonomy cannot be a knowledge-base for IDS classification tasks, and seems to be a knowledge classification, like the two previously explained works.

Lough has tried to develop a taxonomy of computer attacks concerning wireless networks [24]. He summarizes characteristics, features and attributes of a taxonomy. Among all attributes, we believe that, ideally, a taxonomy should bring the capability of predicting future trends into the domain of knowledge. On the other hand, an ontology in a conceptual point of view only presents the current state of the domain. As we accept that an ontology consists of taxonomies, the whole structure inherits the predictive attribute from the taxonomy. In addition, by including several taxonomies in an ontology, it is possible to define detection features and relations between events. This decouples the data model that defines the problem from the logic and decision making process of an intrusion detection system.

In a project by the Technion Computer Networks

Lab. [25], DDoS attacks are classified into bandwidth-throughput attacks, protocol attacks and software vulnerability attacks. Based on this classification, they have also developed a detection system; however, how they have selected detection features, and how the presented classification guides their detection system has not been explained.

In the SRI Computer Abuse Methods Model proposed by Neumann and Parker [26], DoS is not explicitly depicted. They believe DoS is the result of the abuse methods they have tried to categorize.

These two latter works and other misuse models, such as Anderson's penetration matrix [27] or Jayaram and Morse's network security taxonomy [28] are efforts to define the problem not to classify the species. Meanwhile, they do not focus on DoS and speak about computer misuse and attacks in general. Although these cannot be employed to precisely detect a certain type of attack, through them the presented knowledge is a valuable source to analyze the problem.

TOWARD A TAXONOMY OF DOS ATTACKS

To form an organization for DoS attacks; either taxonomy or ontology, we need to select the concepts of the domain to be considered and included in the structure. We select these concepts based on our domain study. In addition to the previous works mentioned in related works and taxonomy sections, around 1500 DoS attack codes are examined and studied. We have used CVE records [29], NVD entries [30] and Snort DoS rules [31], as a source for our attack study. Our process of constructing these taxonomies consists of two steps:

1. Considering all the concepts related to DoS attacks based on a domain study.
2. Arranging naturally related concepts in a class-subclass hierarchy to form a taxonomy.

After that, we try to relate or connect these taxonomies using properties. As a convention, we have capitalized the first letter of class names. This is necessary to distinguish between a class name and a general usage of a specific word. Properties inside an ontology are depicted in italic for the same reason.

We have considered three target side views to network DoS to develop our taxonomies: Consequence of attack, Location of attack and Scenario of attack. We have avoided presentation of other views, which cannot be interpreted by means of target side features and parameters. The three chosen views cover all aspects of attacks that are detectable at the target. In simple words, they explain the characteristics of attacks regarding,

1. Network protocols and traffic through the scenario of attack.
2. How the attack effects its target service through the consequences of the attack.
3. Whether it is distributed or not through the location of the attack.

We describe the developed taxonomies in the following sub-sections and figures. Note that in all the figures, an arrow shows a relation. If the relation is not explicitly labeled, it denotes an 'is-a' or 'kind-of' relationship.

Consequence

"What is the result of a DoS attack?" has been the topic of discussion in much research, each trying to define this subject from a special point of view.

Fallah categorizes DoS attacks to be resource allocation, resource destruction and alteration or destruction of configuration information [15]. We use a modified version of their view in our ontology with the view, consequence of attack, as stated by [12]. It needs to be noted that some experts believe this view is the impact of an attack. The impact of an attack might be service shutdown or degradation, a view which has been partly covered in the work done by Howard.

Howard classifies DoS attacks into destruction process degradation, storage degradation and shutdown [17]. Again, the view of the problem is a combination of the impact and approach of the attack. Moreover, regarding our requirement of taxonomy as part of DoS ontology, the level of abstraction is somewhat ambiguous in this classification. Process and storage are not at the same level of abstraction, and there is no subject specified for destruction and shutdown.

We classify the consequence of DoS to be Service Destruction, Service Exhaustion and Service Control. In destruction, the service completely stops, and there is no way to serve the clients, even with a low quality of service. It is equivalent to shutdown, as described by Howard, and happens when all the necessary supporting services, as defined by Millen, have stopped working. Exhaustion refers to the time when service is logically functional but cannot respond to legitimate users as the result of a massive number of requests in its queue. Flooding attacks in any layer are a good and classic example of this category. When controlling a service, the attacker has the liberty to destruct, exhaust, degrade or control the sequence of service activities. Controlling a service may be the result of remote or local activity. Fluctuating rate attacks are a kind of remote service control, as the attacker releases or exhausts the service on his own decision. Sometimes, control of one service is a way to deny another service. For example, an attacker may target a web service vulnerability and by obtaining local access to this service can easily deny related web-based applications. The reverse approach is also possible. In a Mitnick attack, the attacker denies the target service and redirects all client requests to his side, by masquerading his server machine as the original, thus controlling the whole service.

Location

From a location point of view, DoS attacks may be Local or Remote. Local attacks are launched after access to the service or underlying components are granted or via a bogus application or malicious code. In other words, local DoS attacks are often the result of some predecessor attacks. In contrast, remote attacks usually use a communication infrastructure, and target every component of the service using specially crafted network packets, application requests or massive amounts of traffic. Remote attacks have been the subject of classification under the concept of Distribution. We classify them as Uni-Source and Multiple-Source.

Traditional DoS attacks were launched via a single attacking machine. The growth of hardware and software technologies has concluded to a significant increase in the memory space and processing power of current computers and network devices. Therefore, attacking to deny a service via a single machine is nearly impractical. Uni-Source attacks often make use of a small number of special packets that exploit a vulnerability in the target OS or Application. We say that it is assured that Uni-Source attacks deploy semantic scenarios. The traditional Teardrop recently announced IPSec ESP attacks, and the recent Apache 2.0 HTTP server Mod_Cache vulnerability are examples of this category. Note that there are some other attacks that are classified as semantic, which depend on a large volume of requests, such as the TCP Synflood. These types of semantic attack often exploit a certain characteristic of communication or application level protocols.

Multiple-Source attacks, which use brute-force as their scenario, are classified into DDoS (Distributed DoS) and DRDoS (Distributed Reflected DoS). In a DDoS attack, an attacker compromises a number of other network plugged machines and forms a DoS attack network to send the attack traffic simultaneously to the victim from various machines. In order to hide his identity, the attacker may use several layers of indirection between his machine and attack agents called Zombies. The zombies are controlled by Master machines that may communicate with the attacker directly or through another layer of indirection, sometimes called a Stepping Stone. The famous Synflood attack is the best example for reference to this subclass.

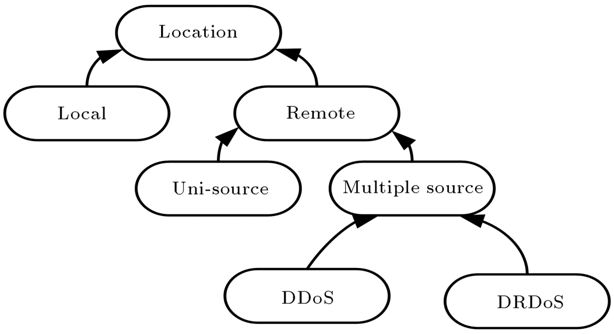


Figure 2. Taxonomy of attack location.

Another effective technique that allows for the amplification of the attack from a single source is a DRDoS attack. The attacker would simply craft packets with the return address of the intended victim, and send those packets to the broadcast address of the reflector network. These packets would effectively reach all available and responsive hosts on that particular network and elicit a response from them. Since the return address of the requests was forged by the victim address, the response would be sent to the victim in the form of a large volume network flow. The early versions of the DRDoS attack used ICMP Echo or UDP Echo services (Smurf, Fraggle and Papasmurf), but every responding protocol, like TCP SYN or DNS replies, are exploited to initiate this kind of DoS attack. Figure 2 summarizes this taxonomy of the attack location.

Scenario

This aspect of DoS attacks has been considered in the work by Mirkovic and Reiher under a branch named “exploited weakness”. We call this classification “Scenario of attacks” as in our previous work [32], and relate it to more precise and detailed classification of attack scenarios under network protocol taxonomy in the following section. In a semantic scenario a specific bug or vulnerability in every service component is the subject of exploitation. The protocol suite implemented inside the OS kernel and bogus applications are specific subjects of exploitation. Brute force attacks are performed using a massive amount of legitimate network flow. There is often no logical weakness in the target of this kind of attack, but as the Internet has been designed to provide a functional and easy to use environment, some legitimate uses are the subject of attackers exploits.

Scenario: Network Protocol

In this specific taxonomy, named a network protocol, we move toward a decision-tree-like structure and present more details about the attacks. McHugh sug-

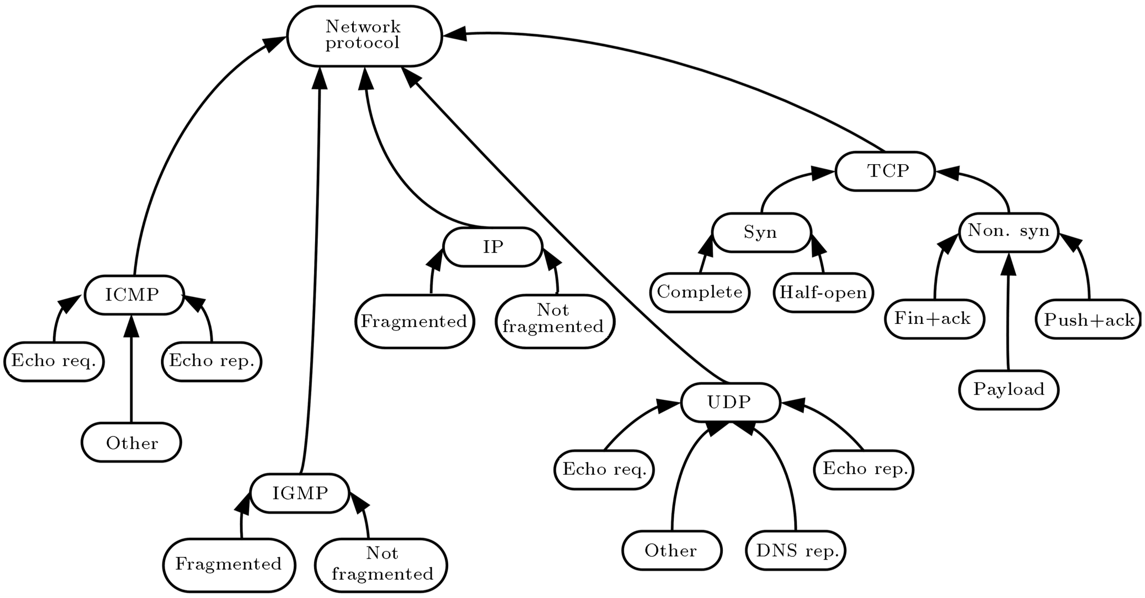


Figure 3. Taxonomy of network protocols based on DoS scenarios.

gests classifying attacks based on a protocol layer [33] and, likewise, Guha and Mukherjee believe that an analysis of each layer of the TCP/IP protocol provides a foundation for an attack taxonomy [34].

As we are proposing this framework for network DoS, other scenarios and means that comprise local attacks are not covered here. To the best of our knowledge, no more scenarios have been discovered, neither in the literature, nor in our experiments. Remote or network DoS – if we exclude bombing and physical attacks – have no means other than network protocols to reach their target. As in Figure 3, several protocols have been selected to form the taxonomy of network protocols based on DoS scenarios. It is obvious that the presented taxonomy is not a complete taxonomy of network protocols and has been tailored to match the results of our domain study.

IP level attacks may use fragmented packets to target OS vulnerabilities, such as Teardrop and Syndrop attacks. A similar scenario is possible and has been observed with IGMP fragmented packets. In addition, an attacker may use complete IP packets to launch a brute force attack in order to deplete the links and target network. The latter is also the subject of several IGMP, ICMP and UDP based attacks, as illustrated in subclass IGMP Not Fragmented, and also ICMP_Other and UDP_Other, which include scenarios in which an attacker uses various ICMP or UDP ports as the destination ports of attack traffic.

ICMP-based attack scenarios have two more major subclasses. The first subclass uses massive Ping or ICMP Echo requests, such as Pingflood or Ping of death. The other subclass redirects echo replies in a DRDoS approach. The same happens using UDP echo requests and UDP echo replies. Moreover, DNS as a responsive protocol has currently been the subject of DRDoS attacks.

The majority of today's network DoS attacks make use of TCP packets. In a famous scenario, a large number of SYN packets are sent toward the target. The target stores the state of this half-open connection and waits for initiators to complete the connection, something that never happens, and the protocol buffer is exhausted. As a result, there will be no room to accept connections from legitimate users. It is also possible to open a huge number of complete TCP connections to overwhelm the ability of the target to accept more connections. These two important scenarios have been classified in the SYN subclass of TCP protocol scenarios.

Non Syn TCP packets are also used as a means of network DoS attacks. The most crowded category includes attacks that target specific application bugs or higher level services, which are classified as payload attacks. FIN+ACK packets are redirected to the target when a responsive TCP protocol is used to launch a DRDoS attack. PUSH+ACK packets force the system to empty its data buffer before it has been filled. In a distributed fashion, the system will be overloaded because of the buffer reference processing overhead, while in a normal TCP connection, the system will not refer to emptying the buffer until it is almost full.

AN ONTOLOGY OF DOS ATTACKS

To develop an ontology of DoS attacks and relate these three taxonomies, we need to define more concepts of the domain. These additional concepts help us to present the needed relations in a comprehensive manner. As we are talking about the Denial of Service, we need to present the root concepts of Service and DoS in our ontology.

The Concept of Service

In trying to construct a framework to detect DoS, it is inevitable that we should clearly define the service itself. We define 'Service' as an abstract concept. As stated by Millen [18], a service is supported by resources or underlying services. It is obvious that each activity that targets the denial of a specific service, needs to shutdown or alter the necessary supports. In order to fully analyze the problem of DoS attacks, we need to consider these supporting resources that are called Service Components. It is possible to rely only on a single class, Service Component, and define every DoS as an act to alter this component. This brings forward the ambiguity of what specific component or resource is the final target of the attack. We know that a DoS may target a wide range of resources from physical ones like server rooms and physical links to high level supporting services, such as web or file services. Without classifying service components to several finer subclasses, it is not possible to analyze the problem and classify the attacks in a way that is valuable for a response system. The major reason behind classifying service components is to find out the target of the attack.

Hautio and Weckstrom have tried to categorize the targets of DoS attacks [35]. The targets of attack are divided into a network infrastructure, host system hierarchy, protocols and software. The network protocol attacks are then subdivided into IP based, ICMP based and TCP based attacks. The work done by Hautio and Weckstrom is somehow close to one level of our view of the problem, but lacks other levels of hierarchy to explore DoS attacks.

To address the requirements explained above, we classify service component into three subclasses: Communication Infrastructure, Host and Application. Although this classification is somehow similar to the work by Hautio and Weckstrom, we have tried to solve the ambiguity of dividing network protocols and software by developing another related taxonomy of network protocols. We do not name service component taxonomy as being the target of DoS attacks directly, but relate this taxonomy to being the target of DoS attacks using a property in our ontology.

All logic media are denoted in the Communication Infrastructure. The protocols up to the transport layer of the OSI model are responsible for transferring information and data from client to server and vice versa. Reasonably, all the upper layer functions are classified into an Application subclass. It is worth saying again that we do not regard the physical aspects of the medium involved in the delivery of service, although they can be classified using the proposed ontology. Moreover, network and communication devices, such as routers and switches, fall into the Host category of our service component taxonomy, as they are constructed on hardware and OS.

We define a host to be a combination of the operating system and hardware. It is obvious that OS includes a part of the communication infrastructure, as some portions of the OSI protocol suite are implemented in the OS kernel. We have separated OS to make it possible to distinguish between attacks that target OS vulnerabilities. These attacks mainly try to lock the system using network protocols that are part of the communication infrastructure. This is presented via the "Targets" property between DoS and the service component and "Implements" the property connecting OS to network protocols. The intent in introducing the hardware subclass to the host class is to complete the organization of the ontology, and bring the possibility of further development of this work to add physical DoS attacks. As another reason, we have observed several attacks that target the hardware of a host by trying to execute a chain of NOP instructions. These instructions put an x86 Intel machine in the trap state and lock up the CPU of a host.

The Concept of DoS

This is the major concept of the domain of knowledge. In a taxonomy, such a concept becomes the root of the class hierarchy. In an ontology, this is the node that can only be organized with outgoing properties. However, in the following sections, as we introduce the concept of service, the ontology becomes more reasonable by adding an incoming property to the concept of DoS. The class of DoS which is a subclass of Attack itself, is a one class taxonomy in our work and does not have any direct subclasses. This can be seen in other presented taxonomies of DoS.

In the taxonomy by Mirkovic and Reiher [21], the class of DDoS Attack Mechanisms does not have any direct subclass, and classification is only possible when different views are described, for example classification by exploited weakness (EW branch of the taxonomy). Moreover, choosing a plural noun for a class name (Attack Mechanisms) is not reasonable, because the concepts in a domain of knowledge are not plural in nature. Again, we emphasize that DoS has no subclass of its level of abstraction. This can be seen in every presented taxonomy of DoS.

Several taxonomies are related to each other using multiple properties or slots to form our DoS ontology, as illustrated in Figure 4. As discussed earlier, by using an ontology, we can easily relate multiple views

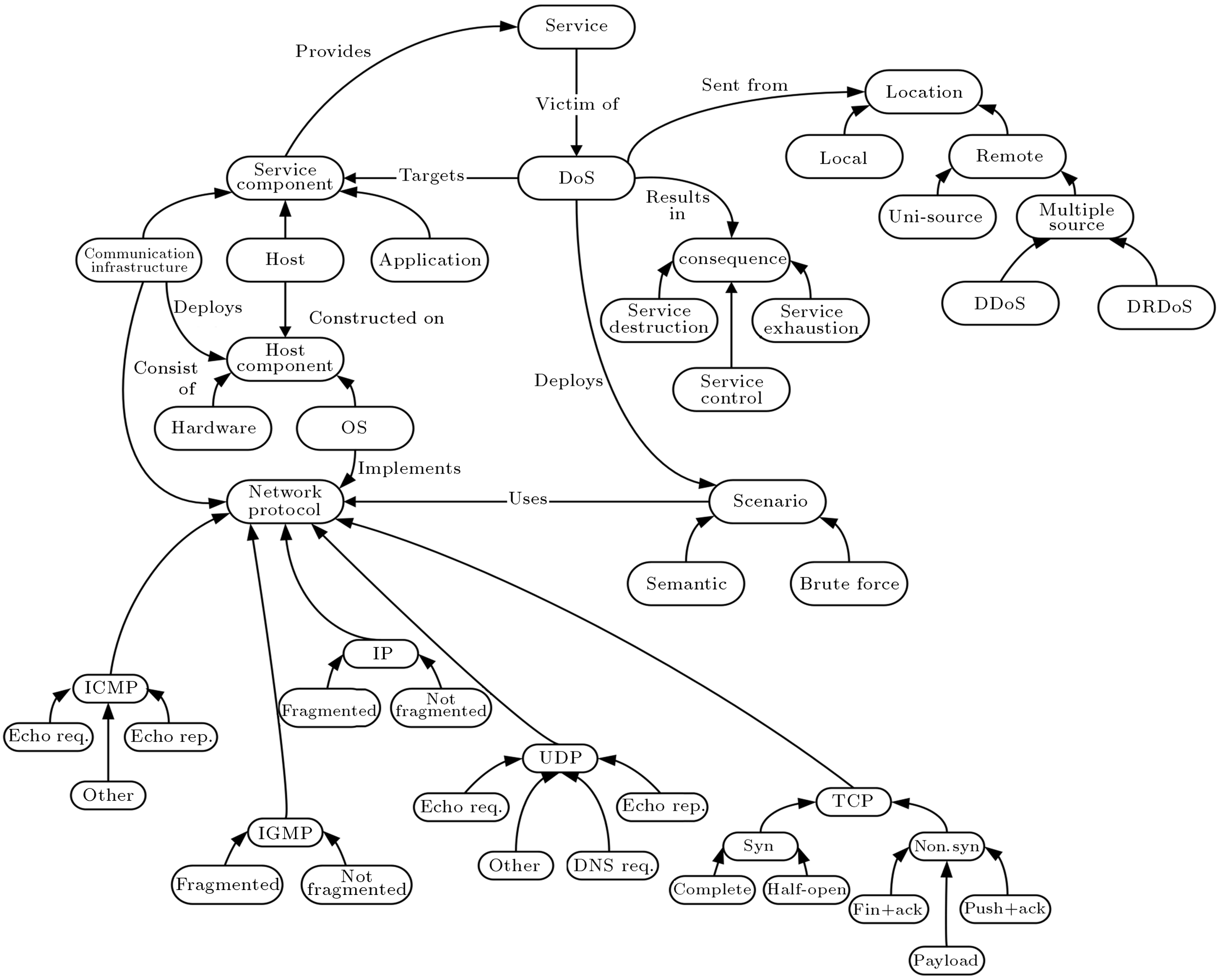


Figure 4. Ontology of network DoS.

or aspects of one major concept. A detailed description of these relations follows.

As a DoS attack targets computer services, we have related “Service” to be the “Victim” of DoS. Obviously, this property creates good potential for developing ontology by another property, “Targets”, which completely relates the concept of DoS to the Service Component taxonomy that “Provides” the Service itself. In other words, a DoS attack may target every subclass of Service Component and, in this way, the related service becomes a victim of the attack. As depicted in Figure 4, a subclass of the Service Component, Host, is “Constructed on” components, Hardware and OS, which can be targets of a DoS attack [5].

In the previous sections, we classified DoS attacks to be remote or local, and also remote attacks to several other sub classes. This location-based taxonomy of DoS attacks has been related to DoS by a “Sent from” property where famous DDoS and DRDoS attacks are located.

The result of a DoS attack on the target side, as fully discussed previously, has been addressed via a “Results in” property. The consequence of a DoS attack may be one of the Service Destruction, Service Control or Service Exhaustion subclasses.

The heart of the ontology is where we introduce the scenario that a DoS attack “Deploys”. Relating to our previously presented scenario-based taxonomy of DoS, we have tried to provide a decision-tree-like classification of DoS scenarios. The operating system Implements the whole protocol suite in a host. A scenario which may be semantic or brute force, “Uses” network protocols as the means of attack. We described this before. Note that the Communication Infrastructure -a subclass of the Service Component- “Consists of” ‘Network Protocols’. This does not mean that DoS attack scenarios that use network protocols, target network infrastructures only. This ambiguity should be relaxed here, and this special property “Consists of” has only been presented to satisfy the complete structure of the ontology and relations between presented concepts.

DISCUSSION: NETWORK DOS TAXONOMY

As explained before, a taxonomy is a hierarchical structure of classes or concepts. This hierarchy relates the concepts via a special property, 'is-a' or 'kind-of'. In organizing a single taxonomy of DoS attacks, we should relate various aspects presented in previous section using this special property. Previously, we stated that the concept of DoS would be the major concept in a taxonomy of this domain of study. Moreover, we need to exclude the concept of service, as there is no way to relate this concept to DoS via is-a property. As depicted in Figure 5, starting from the major concept -DoS- we need to switch between views of the problem and develop the taxonomy. Note that to make the related terms reasonable, we have used new terms to express various views: Target, Approach and Distribution point to Service Component, Consequence and Location, respectively.

It is obvious that target of attack is-not-a DoS. As we want to organize the concepts in a single taxonomy, we need to redefine them. The best way here is to use a certain rule for labeling the objects. At every level of hierarchy, objects are labeled using one of the terms related to that certain level, concatenated by the label of the previous level.

Labels of each level are of the following sets:

TargetOfAttack={Communication Infra., Host-OS, Host-Hardware, Application},

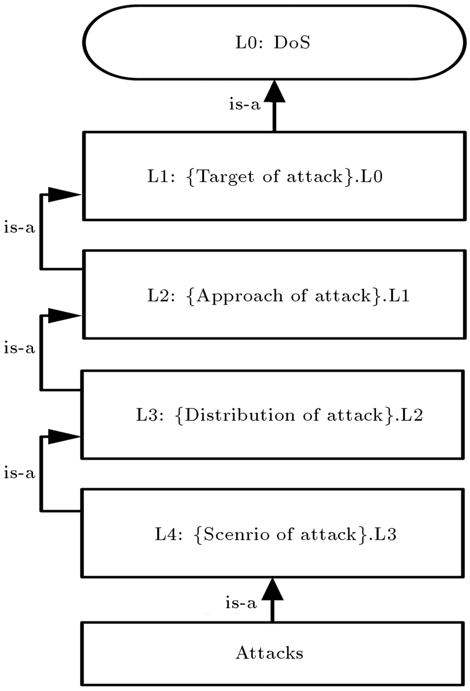

L0: DoS
is-a
L1: {Target of attack}.L0
is-a
L2: {Approach of attack}.L1
is-a
L3: {Distribution of attack}.L2
is-a
L4: {Scenrio of attack}.L3
is-a
Attacks

Figure 5. Taxonomy of network DoS.

ApproachOfAttack={*Destruction,Exhaustion,Control*},

DistributionOfAttack={*Local,Distributed,Reflected*}.

For scenario of attack, we propose a combination of Scenario and Network Protocol taxonomies presented in previous sections, where BF and SE terms denote brute-force and semantic, respectively:

ScenarioOfAttack={IP-BF-NotFragmented,IP-SE-Fragmented,ICMP-BF-EchoReq, ICMP-BF-EchoRep, ICMP-BF-Other,IGMP-SE-Fragmented,IGMP-BF-NotFragmented,UDP-BF-EchoRep,UDP-BF-Ech oRep,UDP-BF-Other,UDP-BF-DNS,TCP-BF-SYN-Full,TCP-SE-SYN-Half,TCP-SE-NonSYN-PL, TCP-SE-NonSYN-FinAck,TCP-SE-NonSYN-PushAck}

As an example, consider a distributed Synflood attack. Using this labeling mechanism, we obtain the following code:

(TCP-SE-SYN-Half).(Distributed).(Exhaustion).
(Host-OS).(DoS)

This approach applies to all instances. As we are trying to redefine relationships that are not naturally 'is-a', it is required to use a coding scheme as described above. In contrast, an ontology with the capability of containing any type of relation or property makes this presentation more simple and straightforward.

ILLUSTRATION: PHARMING ATTACK

To better explain the task of detection framework and especially, our ontology in action, we consider a scenario of a Pharming Attack as an illustration. In a pharming attack, the attacker redirects requests of a target server to another fake server under his control, especially in the case of a web service. Here, we suppose a scenario where the attacker accomplishes the pharming attack through subverting related DNS servers. As shown in Figure 6, in normal cases, the client forwards his name resolution request (what is the address of the server) to DNS1, and in a recursive configuration, DNS1 sends a request to DNS2 asking for the address. The address of the server is resolved via DNS2 and a reply containing the address on demand is sent back to DNS1, which recursively sends it back to the client. This way, the client is able to connect to the server. Now, suppose that we have detection sensors installed at each party: client, server, DNS1 and DNS2. We consider a correlation engine, which includes a rule to detect a pharming attack based on the logic illustrated via the pseudo code in Figure 7.

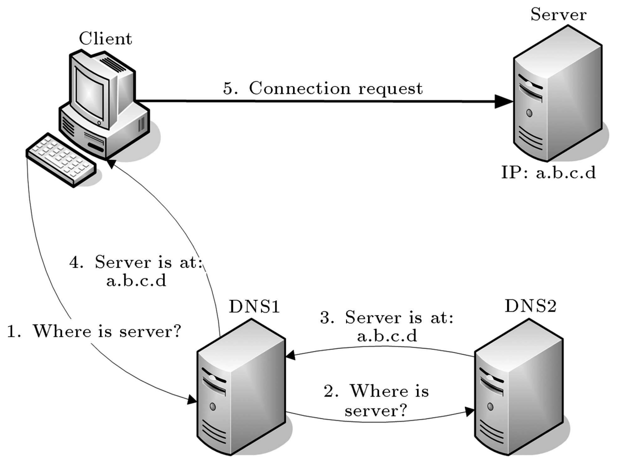


Figure 6. Recursive name resolution-normal condition.

The pharming attack is performed in the following steps as shown in Figure 8:

1. The attacker launches a UDP flood attack on port 53 of DNS2.
2. The attacker tries to guess the Nonce Sequence of DNS1.
3. The client asks for the address of the server from DNS1.
4. DNS1 recursively requests DNS2 for the address of the server, but as DNS2 is under a flooding attack, it does not respond.
5. Using the guessed nonce sequence, the attacker responds to the DNS1 request by spoofing his source address with the address of DNS2, and reporting the address of the server to be e.f.g.h, which is the address of a fake server (the server under control of the attacker).
6. DNS1 sends back e.f.g.h as the address of the server to the client.
7. The client connects to the fake server.

A detection system somewhere in the Internet detects a new attack called Nonce Guessing. This attack may be launched on a DNS to discover the nonce sequence used by the server. The information of this new attack is propagated, using the facilities of the detection framework, to other detection systems over the Internet as follows:

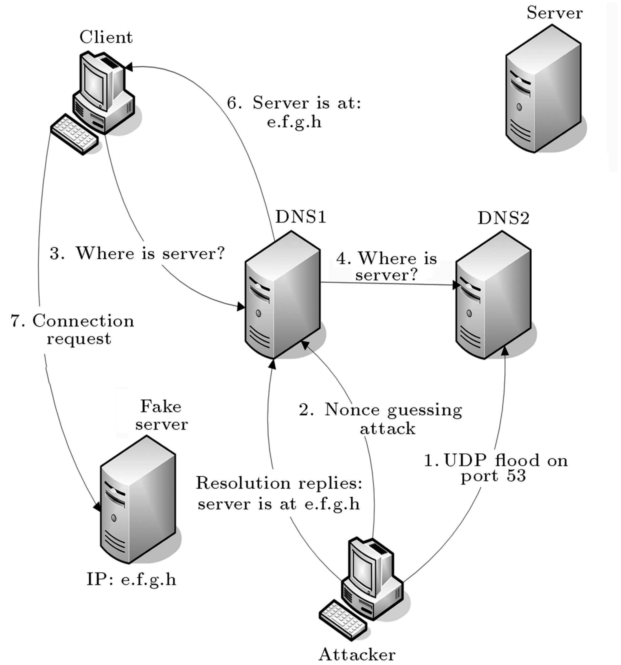
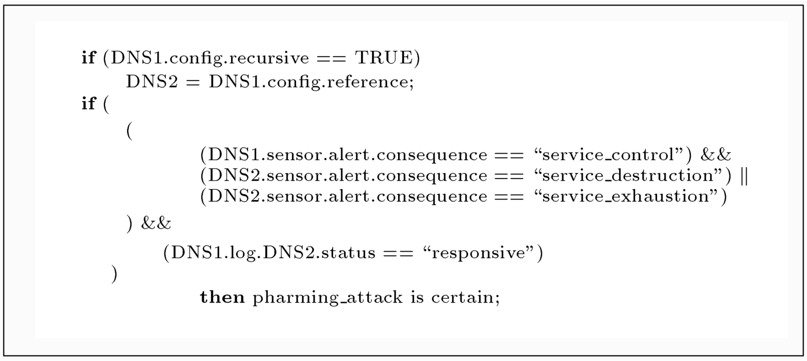


Figure 8. Pharming attack.

- The Nonce Guessing attack is detected and its properties are clarified using the proposed ontology as shown in Figure 9.
- By locating the attack in the ontology, several features like Protocol and Type of Packet are implicitly defined. Moreover, IDMEF provides the capability of reporting values of selected features inside the alarm. We suggest that a feature exchange solution for the unification of this part of the framework would be a better choice. For example, an extension to support linguistic variables as the value of features brings the opportunity for each sensor to interpret the value regarding its environment. In the case of



```
if (DNS1.config.recursive == TRUE)
    DNS2 = DNS1.config.reference;
if (
    (
            (DNS1.sensor.alert.consequence == “service_control”) &&
            (DNS2.sensor.alert.consequence == “service_destruction”) ||
            (DNS2.sensor.alert.consequence == “service_exhaustion”)
    ) &&
        (DNS1.log.DNS2.status == “responsive”)
   )
            then pharming_attack is certain;
```

Figure 7. Pharming attack detection-recursive scenario.

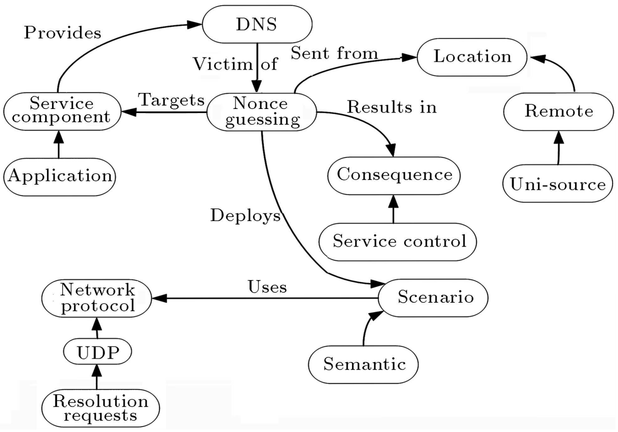


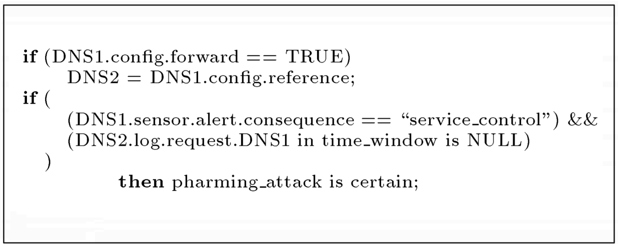
Figure 9. Nonce guessing attack-ontological view.

our scenario, the sensor which detected the attack reports a "high number of DNS requests" in the "time window" besides a certain "payload pattern".

- The decision making process is the dimension of the framework, which every sensor should address on its own. For example, in a fuzzy sensor which interprets the term high by the use of MBF's (Membership Functions), the reported features can be quickly transformed to a detection rule as illustrated in Figure 10.

In this way, the detection sensor on DNS1 detects a Service Control activity called a 'Nonce Guessing attack' using the information received over the Internet from other sensors. Moreover, an UDP flood attack, which is classified as a Service Exhaustion, is detected on DNS2. By the aid of these alerts and access logs of DNS1, the correlation engine can detect the pharming attack. In addition, if the client sends its connection states to the correlation engine, not only does the certainty of detection increase, but it is also possible to find the fake server.

The condition can be extended to cover scenarios in which the attacker attempts a DNS cache poisoning on DNS1 to insert his own address as the fake address of DNS2. In this way, he can easily control access to the server. Considering the DNS cache poisoning attack as a service control activity, the scenario can be detected easily via simple correlation rules, such as the one illustrated in Figure 11.



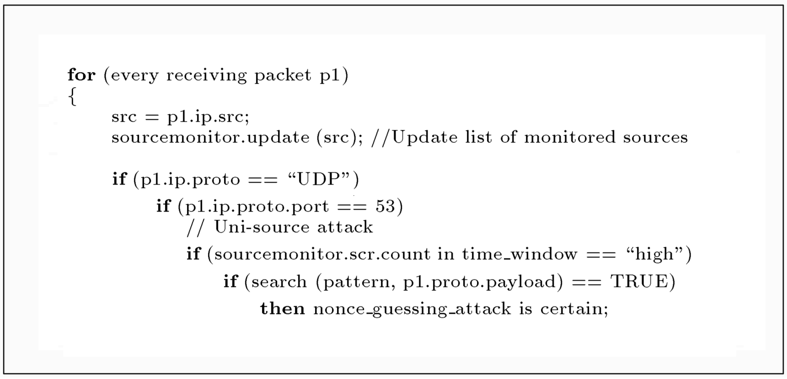
```
if (DNS1.config.forward == TRUE)
      DNS2 = DNS1.config.reference;
if (
      (DNS1.sensor.alert.consequence == "service_control") &&
      (DNS2.log.request.DNS1 in time_window is NULL)
   )
          then pharming_attack is certain;
```

Figure 11. Pharming attack detection-forward scenario.

FUTURE WORK

In another work, we have implemented the proposed ontology using OWL in a Protege environment. This ontology can be the base of reasoning about the network DoS in detection sensors. In the case of new attacks, via various features at the target side, every taxonomy included in the ontology concludes to a certain subclass. This makes it possible not only to alert more precisely but also to inform other sensors and systems in cooperation about all characteristics of a certain attack. Moreover, the scenario-based taxonomy included in the ontology facilitates the process of feature selection for attack detection. Using this taxonomy, we have selected 11 out of 41 KDD 99 Data set features, and by means of a fuzzy rule base and Fuzzy C-Means algorithm, it was possible to detect almost all the DoS session records with a reasonable false positive rate [32].

As a future work, we are going to provide a feature selection base for each of the taxonomies included in the ontology. This satisfies the next requirement of a detection framework; where to inspect. This ontology and corresponding features can be integrated using a XML format in the IDMEF. Prelude IDS gives a good



```
for (every receiving packet p1)
{
      src = p1.ip.src;
      sourcemonitor.update (src); //Update list of monitored sources

      if (p1.ip.proto == "UDP")
            if (p1.ip.proto.port == 53)
                // Uni-source attack
                if (sourcemonitor.scr.count in time_window == "high")
                      if (search (pattern, p1.proto.payload) == TRUE)
                            then nonce_guessing_attack is certain;
```

Figure 10. Nonce guessing attack detection.

opportunity to implement this framework using the Prelude library and XML. Moreover, it is common for various IDS implementers to label similar information differently, because IDMEF effort is mostly concerned with syntactic rules. The presentation of our ontology is one step forward in solving this issue for DoS attacks and making the correlation process straightforward. Extending the ontology in order to cover other types of computer attack, and integration with the correlation framework presented in [36] are also other potential subjects for future research.

Conclusion

With the rapid growth of computer networks, Gigabit Ethernet, 10G connections and other newly presented trends of these technologies, Denial of Service remains as an issue for the availability of services. Without running toward a total solution based on the cooperation of detection, defense and log generation systems, the issue cannot be resolved. In this research, we presented the concept of a Detection Framework to facilitate the cooperation of detection systems. As the heart of this framework, we presented an ontology of DoS attacks, which includes a classification scheme based on attack scenarios to guide intelligent systems. This framework can be extended to other major types of computer attack and, by incorporating other previously presented works, especially IETF's IDMEF, the most important requirements of the framework are addressed. This idea of a detection framework greatly enhances the task of alert fusion and reduces the need for an error prone alert normalization and preprocessing in a correlation engine, as described in [36]. In other words, this framework in a way provides meta-data to be used by several components of a correlation system. Although other requirements of the framework have not been addressed in this research, several experimental studies are evidence of the effectiveness of our presented ontology.

ACKNOWLEDGMENT

This research has been partly supported by Iran Telecommunications Research Center (ITRC).

REFERENCES

1. Bishop, M., *Computer Security: Art and Science*, Addison Wesley Professional (2002).
2. Mahimkar, A. and Shmatikov, V. "Game-based analysis of denial-of-service prevention protocols", in *Proceedings of the 18th IEEE Workshop on Computer Security Foundations*, USA, pp. 287-301 (2005).
3. Agah, A., Asadi, M. and Das, S.K. "Prevention of DoS attacks in sensor networks using repeated game theory", In *Proceedings of the International Conference on Wireless Networks (ICWN)*, USA, pp. 29-36 (2006).
4. IETF "RFC 4765: The intrusion detection message exchange format (IDMEF)", Network Working Group (2007).
5. IETF "RFC 4732: internet denial-of-service considerations", Network Working Group (2006).
6. Hess, A., Jung, M. and Schafer, G. "FIDRAN: a flexible intrusion detection and response framework for active networks", *8th IEEE International Symposium on Computers and Communication*, Turkey, pp. 12-19 (2003).
7. Kemmerer, R.A. and Vigna, G. "Intrusion Detection: A brief history and overview", *Security and Privacy a supplement to IEEE Computer Magazine*, **35**(4), pp. 27-30 (2002).
8. Kruegel, C., Valeur, F. and Vigna, G., *Intrusion Detection and Correlation: Challenges and Solutions*, Springer Science and Business Media Inc. (2005).
9. Allen, J. et al. "State of the practice of intrusion detection technologies", *Report CMU/SEI-99-TR-028, Carnegie Mellon-Software Engineering Institute*, Networked Systems Survivability Program (2000).
10. Cheung, S., Lindqvist, U. and Fong, M.W. "Modeling multistep cyber attacks for scenario recognition", in *Proceedings of the Third DARPA Information Survivability Conference and Exposition (DISCEX III)*, USA, pp. 284-292 (2003).
11. Zhang, G. and Parashar, M. "Cooperative defense against DDoS attacks", *Journal of Research and Practice in Information Technology*, **38**(1), pp. 69-84 (2006).
12. Undercoffer, J. et al. "A target-centric ontology for intrusion detection", in *Proceedings of IJCAI Workshop on Ontologies and Distributed Systems*, Mexico, pp. 47-58 (2003).
13. Yu, C. and Gligor, V. "A specification and verification method for preventing denial of service", *IEEE Transactions on Software Engineering*, **16**(6), pp. 581-592 (1990).
14. Amoroso, E., *Fundamentals of Computer Security*, Prentice Hall (1994).
15. Fallah, M.S. "A formal description of availability and denial of service in computer networks", PhD Thesis in Engineering Faculty of Tarbiat Modarres University (TMU), Iran (2002).
16. CERT Coordination Center, *Trends in Denial of Service Attack Technology*, USA (2001).
17. Howard, J.D. "An analysis of security incidents on the internet", PhD Thesis in Carnegie Mellon University, USA (1997).
18. Millen, J.K. "A resource allocation model for denial of service", in *Proceedings of the IEEE Symposium on Security and Privacy*, USA, pp. 137-147 (1992).

19. Noy, N.F. and McGuinnes, D.L. "Ontology development 101: A guide to creating your first ontology", Technical Report, Stanford University, available at: http: //protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

20. Landwehr, C.E. et al. "A taxonomy of computer program security flaws", *ACM Computing Surveys*, **26**(3), USA, pp. 211-254 (1994).

21. Mirkovic, J. and Reiher, P. "A taxonomy of DDoS attack and DDoS defense mechanisms", *ACM SIGCOMM Computer Communications Review*, **34**(2), USA, pp. 39-53 (2004).

22. Douligeris, C. and Mitrokotsa, A. "DDoS attacks and defense mechanisms: classification and state-of-the-art", *Computer Networks*, **44**(5), USA, pp. 643-666 (2004).

23. Specht, S.M. and Lee, R. "Distributed denial of service: taxonomies of attacks, tools, and countermeasures", In *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems*, International Workshop on Security in Parallel and Distributed Systems, USA, pp. 543-550 (2004).

24. Lough, D.L. "A taxonomy of computer attacks with applications to wireless networks", PhD Thesis in Faculty of the Virginia Polytechnic Institute and State University, USA (2001).

25. Schechner, T. et al. "DDoS project final report", Technical Report, Technion Faculty of Electrical Engineering, Computer Networks Lab (2000).

26. Neumann, P.G. and Parker, D.B. "A summary of computer misuse techniques", In *Proceedings of the 12th National Computer Security Conference*, pp. 396-407 (1989).

27. Anderson, J.P. "Computer security threat monitoring and surveillance", Technical Report, J.P. Anderson Co., USA (1980).

28. Jayaram, N.D. and Morse, P.L.R. "Network security-a taxonomic view", in *Proceedings of European Conference on Security and Detection (ECOS97)*, UK, pp. 124-127 (1997).

29. *Common Vulnerabilities and Exposures*, available at: http://cve.mitre.org

30. *National Vulnerability Database*, available at: http://nvd.nist.gov

31. *Snort: The De-Facto Standard for Intrusion Detection*, available at: http://www.snort.org

32. Varshovi, A. "Fuzzy intrusion detection systems: Enhancement through classification of DoS attacks", PhD Thesis in Computer Engineering and IT Department of Amirkabir University of Technology, Iran (2004).

33. McHugh, J. "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 Systems Security*, **4**(4), pp. 407-452 (2001).

34. Guha, B. and Mukherjee, B. "Network security via reverse engineering of TCP code: Vulnerabiliy analysis and proposed solutions", in *Proceedings of the IEEE Infocom'96*, USA, pp. 603-610 (1996).

35. Hautio, J. and Weckstrom, T. "Denial of service attacks", (1999), available at:http://www.hut.fi/u/tweckstr/hakkeri/DoS paper.html

36. Valeur, F. et al. "A comprehensive approach to intrusion detection alert correlation", *IEEE Transactions on Dependable and Secure Computing*, **1**(3), pp. 146-169 (2004).

BIOGRAPHIES

Ali Varshovi was born in 1980. He received his B.S. in 2002 in Computer Engineering (Software) from the University of Isfahan, and his M.S. in 2005 in Computer Architecture from Amirkabir University of Technology on fuzzy intrusion detection systems. Since then, he has been active in Internet security research especially on Denial of Service. He is also active in web application security and software exploitation. His research interests include Intrusion Detection, Web Security and Fuzzy Logic.

Babak Sadeghiyan was born in 1961. He received his B.S. in 1985 in Electrical (Electronics) Engineering from Isfahan University of Technology, and his M.S., in 1989 in Electronics Engineering from Amirkabir University of Technology, Tehran, Iran. He received his Ph.D. in 1993 in Computer Science from University College at the University of New South Wales, Australia, on the design of secure hash functions. Then, in 1993, he joined the Computer Engineering and IT Department of Amirkabir University of Technology where he is still continuing his academic activities, and is currently an Associate Professor of the department. His research areas of interest include all aspects of Cryptology and Information Security, more specifically.

Dr Sadeghian has contributed to Design and Analysis of Cryptographic Algorithms, Cryptographic Protocols, Hardware Implementation and Cryptanalysis of Cipher Systems, Network Security and Intrusion Detection Systems. He has pioneered the academic study of intrusion detection systems in Iranian universities since 1996. He is author of one book, 14 research journal papers and more than 100 conference papers.