A Scalable, Vulnerability Modeling and Correlating Method for Network Security

Xuejiao Liu[1], Debao Xiao[1], Nian Ma[1], Jie Yu[2]
[1](Institute of Computer Network and Communication, HuaZhong Normal University,China)
[2](Department of Computer Science, National University of Defense Technology, China)
liuxuejiao@gmail.com

Abstract. Nowadays attacks are becoming increasingly frequent and sophisticated, and they are also becoming increasingly interconnected. Recent works in network security have demostrated the fact that combinations of vulnerability exploits are the typical means by which an attacker can break into a network. It is therefore in great need of performing vulnerability analysis to do security analysis first and take the initiative to find hidden safety problems, then plan effective security measures. In this paper, we propose an analysis model, which derives vulnerability analysis functionality from the interaction of three distinct processes: scanning, modeling and correlating. Scanning is served as a significant issue for identifying vulnerabilities. Modeling provides a concise representation for expressing fact base such as host configuration, vulnerability information, and network topology. Moreover, correlating is used to provide a perspective into correlating isolated vulnerabilities in order to construct layered attack graph. Transition rule is presented in scalable design, which enables highly efficient methods of vulnerability correlation algorithm. Finally, a real case study has been described to demonstrate the capability of our model.

Keywords: Network security, scalable, modeling, vulnerability correlation

1 Introduction

Today, with the enhancement of interconnection network and availability of network service, the new means of attack which are direct to the vulnerabilities of system and software are constantly emerging, there has be a growing trend toward vulnerability exploitation in combination. The network is facing increasingly serious security risks. In order to ensure the confidentiality, integrity and availability of computer systems and network, the network administrator needs to do the security analysis first for the system, take the initiative to find hidden safety problems and then plan effective security measures.

There are many potential interactions among multiple hosts and components in a network, such that the configuration and vulnerabilities of one machine will affect the security of others in the network [1]. Generally speaking, many services are perfectly secure when offered in isolation, but when combined with other services they tend to be exploited by attackers to badly compromise the network. There are numerous

examples of such chains. A simple example has been illustrated in [6], that is file transfer protocol (ftp) services and hypertext transfer protocol (http) services hosted on the same machine. If an attacker can use the ftp service to write data to a directory that the web server can read, it may be possible for the attacker to cause the web server to execute a program written by the attacker. For this reason, configuring an enterprise network securely is becoming a daunting task for network administrator.

Even well administered networks are vulnerable to attacks due to the security ramifications of offering a variety of combined services. It is therefore necessary to consider security-related information to protect the network as a whole, including specific vulnerabilities on various hosts in the network, host configuration and connectivity between hosts. More importantly, a significant issue is to design automatic tools that can analyse the configuration of an enterprise network and find potential security vulnerabilities.

Currently, in the field of network security analysis, existing vulnerability techniques are quite effective to scan the single or multiple host vulnerabilities among the target network. However, these tools check all security holes only from isolated perspective, even if a single vulnerability may not appear to pose a significant treat, a combination of such vulnerabilities may allow attackers to reach critical network resources [2]. Our assumption is to model these information after scanning as a general representation of fact, and then based on the scalable transition rule as well as vulnerability correlation algorithm, layered attack graph with probability is construct to identify the critical node of a most possible successful attack.

When analyzing the security of an enterprise network, it is important to consider multi-stage, multi-host attacks. According to *monotonicity property* in [2], where an attacker does not decrease his ability by launching attacks, and hence does not need to relinquish privileges he already gained. That is to say, a determined attacker is not likely to stop at the machine he first compromises, but can be expected to try to penetrate deeper into the network by jumping from one machine to another. Under this assumption, our transition rule is designed based on the the consequence of higher privilege escalated and additional connection established.

The remainder of the paper is organized as follows. In the next section, we summarize the principle of security design goals. An analysis model is then proposed in Section 3. Consequently, in Section 4, we discuss formal definition of fact, transition rules, layered attack graph, and give a detailed case study to illustrate the proposed model. Related work are reviewed in Section 5. Finally, Section 6 concludes the paper.

2 Security Design Goals

Network vulnerability is impossible to be entirely eliminate [6]. This is due to two factors. Firstly, a network to be useful it should offer some services. These services are based on various protocols and implemented in softwares which have not designed perfectly and possibly have some shortcomings of their own. Secondly, a network may likely contain hosts that are misconfigured which can be easily exploited by an attacker.

Just as the saying says "there is no absolute thing in the world", there is no absolute security in the network. Even a seemingly well guarded network is often infiltrated by

a multi-step network intrusion. For security sake, it is therefore important to protect network assets while still allowing the normal functioning to ensure confidentiality, integrity and availability of the network. Thus, network security analysis must achieve the following goals.

Survivability in Service: Nowadays it is difficult to guarantee that any complex piece of software does not contain some flaws [3]. Network security analysis tries to limit vulnerability while still allowing the network to fulfill its purpose and doesn't effect its performance.

Proactive against Attack: Attacks have become more complex than previously and now attempt to exploit multiple vulnerabilities simultaneously while deploying sophisticated mechanisms to hide their malicious payload. Securing the network has never been more challenging than now. Preventing attacks is better than detecting successful attacks. The real problem facing today is not much how to defend against this divers array of attacks - they are emerging in astonishing speed - but how to defend against them in a proactive manner.

Therefore, the motto of security design is pro-action. It is an impending necessity for proactive and survivable architectures that provide fast recovery and convergence stability for networks susceptible to programmed attacks and faults [4]. That is, our main belief is that security risk(and for that matter vulnerabilities) should be handled at an early stage. The goal of network security analysis is to link up individual vulnerabilities using host configuration information in conjunction with network topology, then find the most vulnerable one to take timely measures.

3 Analysis Model

An accurate network security analysis requires a deep understanding of vulnerabilities and their effects on the network components and the knowledge of how these components are inter-related at which point to be exploited by an attacker in a networked system. Existing analysis models have the problems of inadequate capacity of quantitative analysis and lacking for vulnerabilities correlation.

To address these problems, a network security analysis model is proposed here, which can be described in terms of three steps: scanning, modeling and correlating. Each of these three parts is described briefly in the following sections.

In order to analyze the network vulnerabilities, analysis tools should be able to automatically establish the systematic attack scenarios based on the target network vulnerabilities, network services, connection relations and access authorization [5]. Fig.1 demonstrates the analysis model to construct layered attack graph for network security, benefiting from formal modeling and vulnerability correlation.

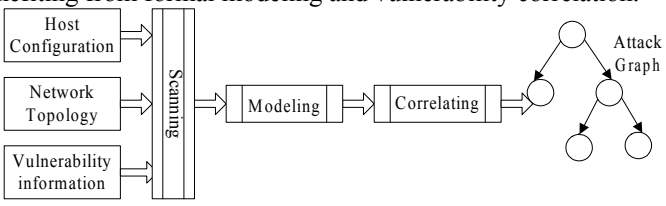

Fig. 1. Network security analysis model

Scanning: Scanning tools and techniques are used in scanning process, determining vulnerabilities of individual host as well as host configuration. Furthermore, understanding network topology (eg. Star, Ring, Fully connected or Partitioned) to specify network accessibility between pair of hosts is critical in considering security analysis.

Modeling: After scanning the network, it may generate a large volume of seemingly disparate information. Modeling process is then used to synthesis and convert these information into a unified representation. In this step we provide a standard representation called fact base for identifying and describing not only network topology but also host and vulnerability attributes.

Correlating: A network vulnerability considered in isolation may not appear to pose a significant threat. But the interdependency of vulnerabilities and the connectivity of a network make such analysis incomplete. To measure the overall security of a network one must first understand the vulnerabilities and how they can be combined to construct an attack.

In this step we perform vulnerability correlation based on the fact base using transition rules. Then an attack graph is produced showing the most vulnerable paths within network. The advantage of static modeling is that system administrators can take advantage of data provided by vulnerability scanners used on their systems and combine it with other known attributes of the network to provide a more complete view of the vulnerability status [7]. While vulnerability correlation algorithm then applies transition rules to static modeling, finding correlation relationship between them. These two approaches can benefit from each other when used together.

4 Vulnerability Correlation

4.1 Fact Modeling

Definition 1 A *fact F* is a triple (*h, v, n*), where (1) *h* is a set of attribute names about host configuration, each with an associated domain of values, (2) *v* is a list of vulnerability information about *h*, (3) *n* is an array of connection matrix used to identify all of the connection relations connected to *h*.

Each fact encodes the knowledge about the hosts within the network and it can be instantiated with any concrete term during modeling process.

4.2 Transition Rule

In order to integrate fact base modeled after scanning process into layered attack graph, a unified representation form is of great need for different types of information in *F*. Thus we design a XML-based rule for regulating the unified pattern representation. We derive transition rule to define logical relations between different

facts. Fact information is static and isolated in nature, additional specification is supposed to give more consideration because dynamic connectivity between the hosts.

Definition 2 A *transition rule R* is defined using XML. Fig. 2 illustrates an example of the transition rule, where it has two XML tags: *fact* and *consequence*. Fact is a starting rule that begins to match the rule if the cve lists on specific host are matched. Consequence refers to the possible impact that vulnerability may result in.

```
<rule id="2" name="WS-FTP Buffer Overflow" probability="0.3">
 <fact cve="CVE-2001-1021,CVE-2007-2213"  host="ANY"
       from="ANY" to="ANY" privilege="ANY">

  <consequence>
    <rule from="1:SRC_IP" to="1:DST_IP" privilege="user"
          addcon="1:SRC_IP;n(1:DST_IP)">
    </rule>
  </consequence>

 </fact>
</rule>
```

Fig. 2. An example of transition rule

The definition of the rule about every attribute is described in Table 1.

Table 1. The attributes of the rule

Attribute	Value	Description
id	decimal	the rule id
name	string	the rule name
probability	<1	the probability of the whole rule
from to	ANY	default value
	IPv4 (x.x.x.x)	IP address
	1:SRC_IP	IP address referenced within the previous rule of SRC_IP
	1:DST_IP	IP address referenced within the previous rule of DST_IP
	n(1:DST_IP)	any host connected to 1:DST_IP
host	h	host configuration information
cve	cve number	a list of vulnerabilities
privilege	none<user<root	privilege access

According to the example of the transition rule, we can see once the cve list is matched, host is padded with *h* in fact, a set of variables such as from, to and privilege are created for that particular vulnerability using *n* in fact. After exploiting the WS-FTP Buffer Overflow, the possible consequence may be that access from 1:SRC_IP to 1:DST_IP will escalate to user and there is another connection relationship from 1:SRC_IP to n(1:DST_IP). Finally probability is assigned to evaluate the reliability of

this specific transition rule. The definition of probability is based on how difficult it is to exploit that vulnerability by the attacker, which depends on empirical knowledge.

4.3 Vulnerability Correlation Algorithm

Definition 3 Given a set of facts F, a set of consequents C, a transition relation $R_t \subseteq F \times C$, then the consequence may be the fact for another consequence, we called a subsequence relation $R_s \subseteq C \times F$, an *layered attack graph* G is the directed graph $G(F \cup C, Rt \cup Rs)$, $F \cup C$ is the vertex set and $R_t \cup R_s$ is the edge set. Figure 3 shows the vulnerability algorithm to construct layered attack graph.

Vulnerability interactions must be considered to ensure a thorough network security analysis, and that interactions produce sequences of events, called attack chains [1].

Definition 4 A layered attack graph may contain several paths from an initial fact to a successful state. For an attack path exploiting the transition rules $r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_n$, suppose the probability for every rule is p_i, i=1,2,…n, then the *attach chain probability* is $A=\{p_1 p_2 \cdots p_n \mid 0<p_i<1\}$

$$A = \prod_{i=1}^{n} p_i$$

What vulnerability correlation do is to generate attack graph. Furthermore, understanding how attackers exploit these vulnerabilities to accrue further capabilities is critical in considering security countermeasures. Fig. 3 illustrates vulnerability correlation algorithm.

INPUT: A set of fact base FB

OUTPUT: Layered attack graph G

1. $C, F, R_t, R_s \leftarrow \varnothing$
2. for each $f \in FB$ {
3. if f matches one of the rule
4. create a consequence node c
   $C \leftarrow C \cup \{c\}$
   $F \leftarrow F \cup \{f\}$
   $R_t \leftarrow R_t \cup \{f \xrightarrow{probability} c\}$
5. look up $f' \in FB$ such that there is rule(c) = fact f' {
   $R_s \leftarrow R_s \cup \{c \xrightarrow{probability} f' \xrightarrow{probability} c'\}$
   $F \leftarrow F \cup \{f'\}$
   $C \leftarrow C \cup \{c'\}$
6. generate an layered attack graph G initialed with f
   }
   }
7. compute every attack chain probability A_i
8. output G with the max probability of attack chain

Fig. 3. Vulnerability correlation algorithm

As is shown in Fig. 3, in the correlating process, it first reads all the transition rules on startup in order to match individual rules. It functionality ressembles a logical tree

cconsisting of "if" and "or" statements, joined together to provide reliable means of identifying attacks or network misbehaviour.

Attack graphs allow the security analyst to assess the true vulnerability of critical network resources, and to understand how vulnerabilities in individual network services contribute to overall vulnerability[14]. The probability index can be used as an indicator to trigger proactive and survivable methodologies to aid fast recovery at the earliest possible stages.

Such attack graphs allow one to see, step by step, the various ways an attacker can incrementally penetrate hosts within a network. Also, it is a layered graph constructed as a result of the relationship between the exploitation of vulnerabilities in different hosts that an attacker may likely carry out in order to reach his final goal.

4.4 A Case Study

In order to validate the effectiveness of the vulnerability correlation algorithm, we construct an experimental environment. Its topology is shown as Fig.4.

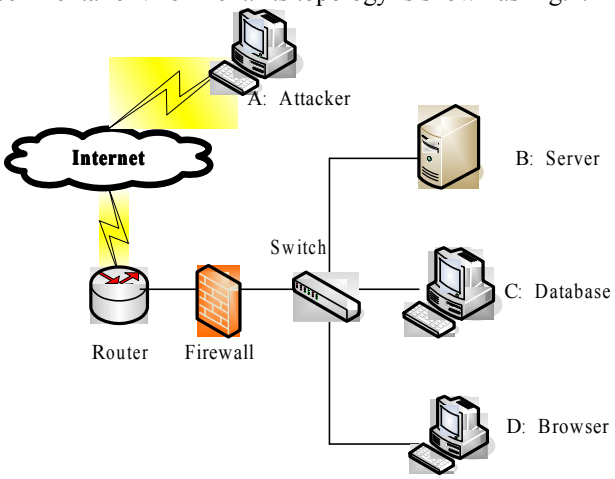

Fig. 4. Experimental network topology

The experimental environment is a switch network, including three host: B, C and D. Host B refers to server running IIS and SSH service. Host C is an important Oracle database running on Linux. Host D is a general browser which can access C. The firewall only allows the exterior hosts to access host B' s via WWW service. The attacker's goal was assigned to gain root privilege of the server.

After scanning and modeling process, the fact information (h and v) of each host is shown as Table 2 and connection modeling is illustrated in Table 3.

There are IIS buffer overflow and sshd buffer overflow existing in host B, which may lead to denial of service or the execution of arbitrary node. Due to another buffer overflow of Oracle in host C, successfully exploiting it an attacker can achieve privilege escalation.

Table 2. Fact (h and v) modeling of the experimental network

H	h	v	Description	Probability
B	122.204.142.1,*	CVE-2002-0150	IIS buffer overflow	0.3
		CVE-2002-0640	Sshd buffer overflow	0.6
C	122.204.142.2, Linux	CVE-2002-1767	Tnslsnr buffer overflow	0.3
D	122.204.142.3, Windows	CVE-2006-2373	SMB Privilege Elevation	0.7

Table 3. Connection relatiship modeling of the experimental network

n	A: Attacker	B: Server	C: Database	D: Browser
A: Attacker	y	80	n	n
B: Server	y	y	y	y
C: Database	y	80	y	y
D: Browser	y	80	y	y

According to vulnerability correlation algorithm as well as transition rule, the layered attack graph is automatically generated as Fig. 5. Each transition in the attack graph represents a specific exploit that an attacker can carry out. Because C trusts D, then D can access C without authenticate and the probability reaches to 0.9 high.

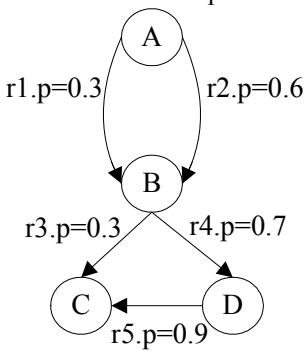

Fig. 5. The layered attack graph

Through analyzing Fig. 5, all the attack chains and their successful probability are shown as Table 4. From that, we can know that A→r2→B→r4→D→r5→C is the most dangerous path than others. So it is necessary to patch for the vulnerability of Tnslsnr buffer overflow in Oracle (CVE-2002-1767) as well as SMB Privilege Elevation (CVE-2006-2373). In addition, strengthening the authenticate relationship from browser D to Database C is also important.

Table 4. Analysis results of attack chain probability

attack chain	probability
A→r1→B→r3→C	0.09
A→r2→B→r3→C	0.18

A→r1→B→r4→D→r5→C	0.189
A→r2→B→r4→D→r5→C	0.378

5 Related Work

Recent works in network security have demostrated the fact that combinations of exploits are the typical means by which an attacker breaks into a network. Analyzing vulnerability for network security is still a new topic in security area, and currently it is rising more and more attentions.

Vulnerability Analysis: Vulnerability analysis techniques perform rigorous examinations to identify system vulnerabilities. There are quite a few vulnerability scanners that are quite effective at what they do - namely identifying vulnerabilities in specific hosts of a target network. Some of the tools are: Nessus [11], System Scanner [10], Retina [12] and etc. However, they do not attempt to identify how combinations of the vulnerabilities on the same host or between hosts on the same network can contribute to the exploitation of a network. Under such circumstances, vulnerability modeling and correlating method is proposed in this paper to improve the operation and efficiency of vulnerability analysis by creating abstract representation conducive to further analysis, chaining together the vulnerabilities uncovered by such tools and thus discover combinations of the exposed vulnerabilities.

Recent advances in vulnerability analysis have yielded the representation of organizing the chains of exploits in the form of graph-based analysis and model checking. While attack graph provides a visual means to represent attack scenarios, and its weakness lies in their construction, as the process is typically accomplished manually. Model checker is served as a powerful inference engine for chaining together network exploits, and suffers from scalability problems. In this paper we revisit the work of graph-based analysis.

Graph-based Analyis: Graph analysis is a common computer science technique to capture steps of a successful system compromise. Researcher have proposed a variety of graph-based approach to represent and generate attack graph (tree, petri net) for analyzing network security [1, 2, 5, 7, 8, 13, 14, 15], including logical attack graph [1], privilege graph [13], exploitation graph[7]and etc. Some of them has addressed scalability [1, 8] and automation [9] problems. Although the precise definitions of attack graph vary by author, there is little attention paid to quantitative analysis in the representation of attack graphs, which results in the attack graph being difficult to use and understand by human beings. Unlike existing approaches whose solutions requires removing the unnecessary nodes and decreasing the scale of the graph, our attack graph is constructed initially with max probability in the attack chain.

6 Conclusion

This paper has proposed a network security analysis model which aims to achieve security design goals, that is to provide proactive against attack and survivability in

service. The layered attack graph is generated method based on transition rule. The scale problem of attack graph is solved by limiting attack steps and max successful probability of attack paths.

Layered attack graph directly illustrate casual relationship among vulnerabilities and therefore have the significant advantage of identifying the critical node by probability attached to the transition rule. Our analysis model will complement the available tools for a comprehensive analysis of the exposed network. The case study has shown that our vulnerability correlation algorithm is very efficient.

References

1. X. Ou, W. F. Boyer, and M. A. McQueen. A Scalable Approach to Attack Graph Generation. In Proceedings of the 13th ACM conference on Computer and Communications Security (CCS 2006). Alexandria, Virginia, USA, October 30-November 3, 2006, 336-345.
2. S. Jajodia, S. Noel, and B. O'Berry. Topological Analysis of Network Attack Vulnerability. In V. Kumar, J. Srivastava, and A. Lazarevic (Eds.), Managing Cyber Threats: Issues, Approaches and Challanges. Kluwer Academic Publisher, 2003.
3. J. Andrews and T. Moss. Reliability and Risk Assessment. The American Society of Mechanical Engineers, 2002.
4. G. Qu, JayaPrakash, Ramkishore, and S. Hariri. A Framework for Network Vulnerability Analysis. In Proceedings of IASTED International Conference Communications, Internet and Information Technology (CIIT 2002). St. Thoams, Virgin Islands,2002, 289-298.
5. D. Man, B. Zhang, W. Yang, and etc. A Method for Global Attack Graph Generation. IEEE International Conference on Networking, Sensing and Control (ICNSC 2008). China, April 6-8, 2008, 236-241.
6. R.W. Ritchey and P. Ammann. Using Model Checking to Analyze Network Vulnerabilities. In Proceedings of the IEEE Symposium on Security and Privacy. Washington, May, 2001, 156–165.
7. W. Li and R. B.Vaughn. Cluster Security Research Involving the Modeling of Network Exploitations Using Exploitation Graphs. In Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops (CCGRIDW 2006), 2006.
8. P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, Graph-based Network Vulnerability Analysis. In Proceedings of the 9th ACM Conference on Computer and Communications Security. Washington, DC, USA, 2002, 217-224.
9. O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M.Wing. Automated generation and analysis of attack graphs. In Proceedings of the 2002 IEEE Symposium on Security and Privacy (S&P 2002), 2002, 273–284.
10. Internet Security Systems, SystemScanner.http://www.iss.net.
11. Tenable Network Security, Nessus. http://www.nessus.org.
12. eEye Digital Security, Retina Network Security Scanner. http://www.eeye.com/html/index.html.
13. M Dacier. Towards Quantitative Evaluation of Computer Security. Ph.D Thesis, Institut National Polytechnique de Toulouse, Decemeber 1994.
14. S. Noel, M. Jacobs, P. Kalapa, and S. Jajodia. Multiple Coordinated Views for Network Attack Graphs. Workshop on Visualization for Computer Security. USA,2005, 99-106.
15. S.J.Zhang, J.H.Li, X.Z.Chen and L.Fan. Building network attack graph for alert causal correlation.Computer&Security.2008,1-9.