Attack Graph Generation with Implementation in Network Security

TAO ZHANG, MING-ZENG HU, XIAO-CHUN YUN, YONG-ZHENG ZHANG
Research Center of Computer Network and Information Security Technology
Harbin Institute of Technology
P.O.Box 320, No.92, West Da-Zhi Street, Harbin, 150001
CHINA

Abstract: - As an important method to analyze the security states of computer network, the generation of network attack graph is a hot topic in this domain. After analyzing network vulnerabilities, linking relation between devices and the characteristic of attack, the model of network security states is built, and the generating algorithm of attack graph is implemented. The experiment validates the prototype of generating tools of network attack graph.

Key-Words: - Network Security; Security Analysis; Attack; Attack Graph

1 Introduction

The rapid growth of the Internet influences the economy, politics, culture and many aspects of the society. The deeper and wilder the internet applications is, the more obvious and more complex the computer and network's security problems are. Hackers and virus can find more ways to launch attack with the development of the network technology.

As an important aspect of network security, evaluating the computer security through the analysis of the system information is very important and could protect us form being hacked. This article presents a method to generate attack graph in order to analyze network security. The organization of this article is as follows. Section 2 introduces the model for network security. Section 3 presents the attack graph generating method and the experiment is given in section 4. A conclusion is provieded in Section 5.

2 Model for Network Security

Since security analysis mainly aims at current computer network, it needs a simple、flexible and complete model to reduce complexity of system states space. In this paper we build a model for the network security analysis referencing to the attack character of privilege escalation that attack brings [1] and some approaches used [2] [3] [5]. The related conceptions and definitions are shown as follows.

2.1 Computer and Network

The devices on the network are the basic elements of information system, for example, computers, routers, switches and the like. We use a set, $H=\{h_1, h_2, \ldots h_m\}$, to represent these devices, and $h_i(i=1, 2, \ldots m)$ represents a single network device.

A host on the network is represented by a tuple *(HOSTID, OS, SVCS, VULS)*. *HOSTID* is the unique identifier of host on the network, it can be the IP address or host name. *OS* is the type and version of operation system. *SVCS* is the list of network service types with respective network port numbers which describes the services on host and the information on service monitor ports. *VULS* is the host computer vulnerability list which may include the security bug information of installed software or environment misconfigure information, and is presented by its Bugtrap ID (BID).

2.2 User Privilege

In actual implementation environment of host computer, the system visitors can be classified according to the capability to access the system resource. A lot of researchers have described on this direction [1]. This article proposes to rearrange the visitors and so the possible privilege can be classified according to user's roles, described in Table 1.

Table 1 Classes of Privilege

Privilege class	Role description
ROOT	System administrator, managing all system resources.
USER	Any general system user, which is created by administrator.
ACCESS	Remote visitors which may access network services, can communicate with services and scan system

2.3 Connecting Relation between Devices

The Internet is structured based on TCP/IP protocol family, and the current computer networks are generally based on this protocol. TCP/IP protocol family includes a lot of protocols which are on different layers. According to this technology principle, the connections of network devices distribute on different layers. Ritchery has analyzed the host connectivity for network security [7]. In the paper these connection relations can be expressed as a set, and then the connection relations between two devices can be a sub-set of this set.

To assume the connection relations set between host and devices is $Protocol= \{pro_1, pro_2, \ldots pro_n\}$, $pro_i(i=1, 2, \ldots n)$ which presents a connection relation. The connection relation sets are shown as Table 2.

Table 2 Onnection Relation Sets

Protocol Layer	Link Relation Example
Application Layer	TCP（UDP）_PORT_SERVICE_APPLICATION
Translation Layer	TCP（UDP）_PORT
Network Layer	ICMP_MESSAGE Type
Data Link Layer	ARP，HUB_SNIFF，SWITCH_SNIFF

The connection relations between hosts are represented by a triad *(HSRC, HDST, Protocols)*. *HSRC* represents source host. *HDST* represents destination host. *Protocols* are a sub-set of connection relations sets between source host and destination host. When there is no relation between source host and destination host, *Protocols* is empty set. When the source host is the same as destination host, the connection relation is local connection, at this time, *Protocols = {localhost}*.

2.4 Security Requirement and Attack Object

The security requirements guaranteed by using a series of security strategies are the users' demand for integrality, usability, confidentiality of system information. They have independent, complementary and interrelated relationship each other. Independence means that each security attribute can be evaluated as an individual security quality, and there is no conflict between them. Being Complementary means that each security attribute emphasizes on different users' requirements, and they supplement to each other, so they can integrate to reflect the full-scale security requirements. Interrelation means that if an attacker can break a certain security attribute, at the same time he has the ability to break other security attributes, and the system vulnerability which influences a security attribute can also influence the other security attributes in different levels.

By analyzing the different types and levels of security requirements, we can find the attack methods which are used to destroy this security attribute. After synthetically analyzing current attack methods and results, the privilege of object systems which attacker can get is the main factor. This article analyzes how to protect the administrator's privilege of system devices as instantiation of security requirements and security attributes.

2.5 Attack Rule

Basically, using a vulnerability to attack can be seen as a map from a set of preconditions to a set of results. So an attack can be represented by a two-tuples *Attack_rule=(Preconditions, Postconditions)*, in which *Preconditions* is the preconditions set, *Postconditions* is corresponded results set .

The preconditions set includes four elements which is represented as *Preconditions={Src_privilege, Dst_privilege, Vuls, Protocols}*. *Src_privilege* represents the lowest privilege which attacker should have on the host where the attacks are launched. *Dst_privilege* represents the highest privilege which attacker should have on the object host. *Vuls* represents the vulnerability which the attack rule depends on. *Protocols* describe the needed connection relation between attack host and object host.

The results set include three elements which is represented as *Preconditions={Rslt_privilege, Rslt_protocols, Rslt_vuls}*. *Rslt_privilege* describes the privilege which attacker can get on object host after an attack is successfully completed. *Rslt_protocols* is the network protocols set which is added by attacks. If the attacked host can use the network protocols in this set to access a host on the network, the current attacking host can get the ability to access this host. If the attack rule doesn't influence the current network connection relations, *Rslt_protocols* will be an empty set. When *Rslt_*

protocols={all}, this represents that the current attacking host can get the attacked host's total ability to access the object network. *Rslt_vuls* is the newly added vulnerability set on attacked host after attack is successfully implemented, and it describes the dependent relation between vulnerabilities.

According to described above, the attack rule can be represented as: *Attack_rule=({Src_privilege, Dst_privilege, Vuls, Protocols}, {Rslt_privilege, Rslt_protocols, Rslt_vuls})*.

3 Generating Attack Graph

Synthesizing the attacker's starting point and object, host information and network topology information, the based-graph description represents the threat to security of information system, and it is called attack graph. To analyze the network security, based on the analysis of network security incidents and attacker's actions, we make assumptions as follows:

Assumption 1: The attacker has the powerful attack ability, namely, attacker who knows the vulnerability well in system has the ability to attack these vulnerabilities.

Assumption 2: The attacker is sophisticated so that he doesn't launch an attack to get the privilege that he has possessed.

We use the network states to present the model information of network described in Section 2. This system use a forward-search, breadth-first and depth-limited (attack steps limited) attack route producing algorithm to find the attack routes, then utilize the tools Graphviz [8] to generate attack graph. The algorithm to produce attack route is described as follows:

```
Algorithm：NAG_Generate(M)
Input：Model information of network for security M
Output：Attack route clue
1.  Set up the initial network state--init_state;
2.  Add init_state into state_queue;
3.  while (state_queue not empty && deepth<Max_deepth)
4.      cur_state ß Get_State(state_queue);
5.      if (M specified attack object && attack object have achieved)
6.          continue;
7.      set up host_queue that include hosts these have link protocols with the host attack launched;
8.      while ( host_queue not empty )
9.          host ß Get_Host(host_queue);
10.         set up protocol_queue these could be used access other host by the host launched attack;
11.         while( protocol_queue not empty )
12.             protocol ß Get_Protocol(protocol_queue);
13.                 set up the attack_rule_queue corresponding with protocol;
14.                 while( attack_rule_queue not empty)
15.                     attack_rule ß Get_Attack_Rule(attack_rule_queue);
16.                     attack the host according to attack_rule;
17.                     if (attack successfully && attacker's privilege escalated on the host)
18.                         generate new network state new_state;
19.                     if (new_state don't appeared before it)
20.                         add new_state into state_queue;
21.                     output attack route clue;
```

When each attack depends on the previous attack on attack route, the attack route is called minimal attack route. In comparison with the method used [2] [3] [5], our method can directly find all minimal attack route in the 17th step of the above algorithm. At the same time, in attacker's point of view, breadth-search guarantees the creating of all the attack routes.

4 Experiments

We validate the prototype system in the environment of the laboratory network to find all of the attack routes which the attacker may choose.

4.1 Network Environment

In our environment, the network topology is shown as Fig.1. The firewall separates the internal network from external network. There are four computers on the internal network, and attacker's host is IP0 on the external network. The host information on the network is shown as Table 3. The firewall allows HTTP and SSH packets to enter the internal network for communication with IP4, but interdicts other packets. In the internal network, the

connection relation won't be controlled by firewall, so it can be assumed that internal host can connect with any remote server.

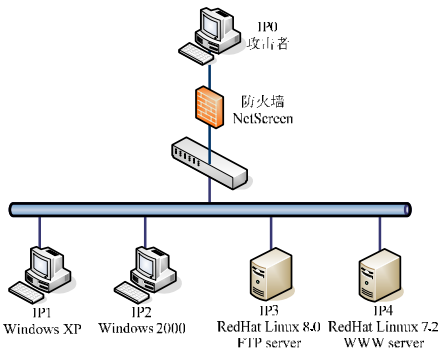

Fig. 1 Network Topology

Table 3 Host Description

HOSTID	OS	SVCS	VULS
IP1	windowXP	{Netbios Remote procedure call, Netbios name}	{10108, 10181}
IP2	window2000	{Netbios session}	{10212}
IP3	Red Hat Linux8.0	{ftp , ssh}	{6410}
IP4	Red Hat Linux7.2	{http , ssh }	{10201, 10212}

According to network vulnerability analysis in SANS TOP 20 in recent years, our system has 183 attack rules including 8 rules associated with the experiment network. In our experiment, we assume that the attacker will attack the internal network using host IP0. At the initial state, we set that the attacker has the highest privilege ROOT on IP0. However, on the other host he just has the lowest privilege ACCESS. And in this experiment, we don't limit attacker's objects and the maximal number of attack steps.

4.2 Results Analysis

After implementing of attack graph generating tools, the attack graph which is got is shown as Fig.2, and it includes all possible attack routes. In Fig.2, the node presents the network states under attack, and the directed edge presents attack action adopted by attacker. The red node presents the network states that the attacker can get a ROOT privilege on some host system when the attacker translates into the current states from the former states, and the yellow node presents that the privilege got on some host only is USER when the attacker translates into the current states from the former states.

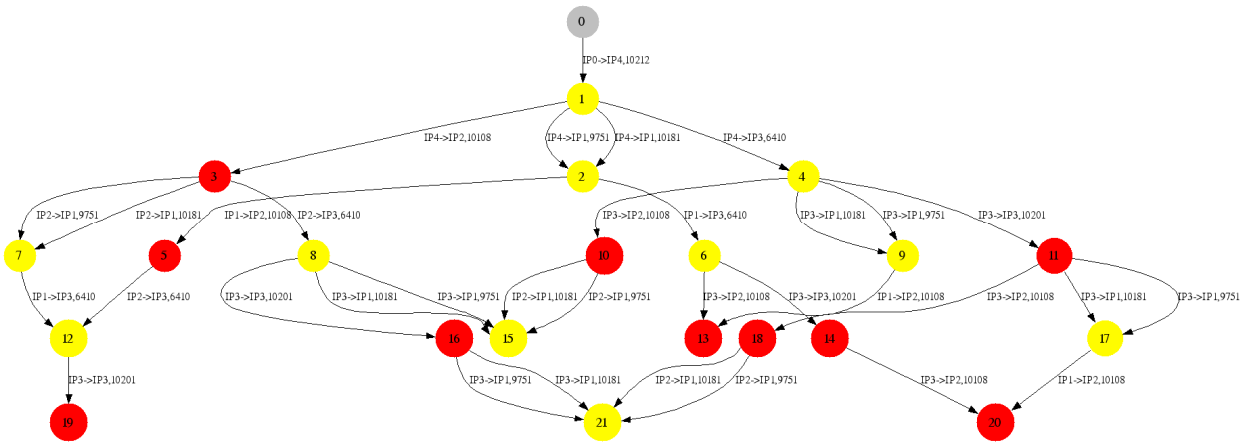

Fig. 2 Attack Graph

According to the analysis of the entire attack route in Fig.2, although we set static rules to protect host IP1, IP2 and IP3, the possibility of attack to them from an attacker still exist. The privileges which the attacker can obtain on the different hosts are shown as Table 4, and there are some attack routes which can implement these attack purposes. For example, the attacker can obtain the host IP2 privilege ROOT by nine different attack routes, and the one of these attack routes have three attack steps:

Step 1: The attacker uses vulnerability 10212 and attacks host IP4 from host IP0 (state 0 **à** state 1);

Step 2: The attacker uses vulnerability 10181 and attacks host IP1 from host IP4 (state 1 **à** state 2);

Step 3: The attacker uses vulnerability 8205 and attacks host IP2 from host IP1 (state 2 **à** state 5);

Table 4 Possible Result Privilege on Each Host

Host	Top Privilege Class
IP1	USER
IP2	ROOT
IP3	ROOT
IP4	USER

From the experiment results above, we believe

that the attack graph's generating tools can dynamically simulate network attacks from attackers, and provide the possible attack steps the attacker adopts and describe the harm to network. Using results analysis, security administrator can understand the network security states and make some decisions to strengthen the network security.

5 Conclusion

The tools to generate attack graph are designed and implemented, and the experiment indicates the method is usable and effective. Many related research should be done in the future, the results from network scan tools should be used in the tools. The generating algorithm should be optimized and the method to analyze attack graph should be further studied.

Acknowledgements

This paper is supported by the pre-research project for national defense under the grant No.41315.7.1. Some implementations are done with the help of some members in our research center.

References:

[1] Zhang Yong-zheng, Yun Xiao-chun, A New Vulnerbality Taxomomy Based on Privilege Escalation, *2004 Sixth Internatioan Conference on Enterprise Information Systems Proceedings*. Porto: INSTICC 2004.

[2] L.Swiler, C.Philips, D.Ellis, and S.Chakerian. Compter-Attack Graph Generation Tool. *In Proceeding of the DARPA Information Survivability Conference & Exposition II*, Anaheim, California, June 2001.

[3] P.Ammann, D. Wijesekera, S. Kaushik. Scalable, Graph-Based Network Vulnerability Analysis. *In Proceedings of CCS 2002: 9th ACM Conference on Computer and Communications Security*, Washington, DC, November 2002.

[4] Steven Noel, Sushil Jajodia. Managing attack graph complexity through visual hierarchical aggregation. *In Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, Washington, DC, 2004.

[5] Sheyner, J.Haines, S.Jha, R.Lippmann and J.Wing. Automated Generation and Analysis of Attack Graphs. *In Proceedings of IEEE Symposium on Security and Privacy, Oakland, California*, May 2002.

[6] Oleg Sheyner. Scenario Graphs and Attack Graphs. Ph.D Thesis at Carnegie Mellon University, April 2004.

[7] R.Ritchey, B.O'berry and S.Noel. Representing TCP/IP Connectivity for Topological Analysis of Network Security. *In Proceeding of 18th Annual Computer Security Applications Conference*, Las Vegas, Nevada, December 2002.

[8] Graphviz. http://www.graphviz.org/pub/graphviz/ARCHIVE/graphviz-1.12-1.i386.rpm. 2005.