Automatic Generation for Penetration Testing Scheme Analysis Model for Network

Lu Shen, Xiaoyan Liang, Yang Bo, Chunhe Xia
State Key Laboratory of Virtual Reality Technology and System
Beijing Key Laboratory of Network Technology
School of Computer Science and Engineering, Beihang University
Beijing, China
{lusknight, Liangxiaoyan, boyang }@cse.buaa.edu.cn, xch@buaa.edu.cn

***Abstract*—The existing penetration testing systems need to rely on the professional skills in the testing process, so they increase the penetration testing of human resources and costs. While it also reduces the efficiency of the test, it increases the test cycle. This paper presents a new penetration testing scheme by an automatically generated method. Through penetration testing scheme description, we establish an automatic generation system about penetration testing scheme, and the automatic generation penetration testing scheme for network analysis model (AGPTSAM) is established, and then the termination of the AGPTSAM is proved by the pushdown automaton. The prototype system of AGPTSAM is implemented. The penetration testing scheme could provide guidance and basis for the implementation of the penetration testing system, and shorten the test cycle and the human resources cost. This method is verification of effectiveness by experimental which works on the implementing platform.**

Keywords-Penetration Testing, Scheme, Automatica Generation;

I. INTRODUCTION

Penetration testing is security testing in which assessors mimic real-world attacks to identify methods for circumventing the security features of an application, system, or network. It often involves launching real attacks on real systems and data that use tools and techniques commonly used by attackers [1]. It can help validate and confirm the effective configuration of an organization's firewalls and its intrusion detection systems, and "test" its information security and response capability [2]. Penetration testing which is from the perspective of an attacker gives the tested side of information security assessment. Through a penetration testing about business, organization or sector, we can get their information security polices and network vulnerability, and give possible solutions and remedies to enhance network security. Existing penetration tests are mainly based on network security experts' experience and ideas to carry out the work, but it takes a lot of human resources and causes extensions of the penetration testing cycle. The common penetration testers cannot properly participate in the core work to the testing. This paper presents a solution for network penetration testing method of automatically generated. It mainly divides the penetration testing scheme and the implementation of the system, so it enables the people use penetration testing system not only independent of the network security experts. Penetration testing schemes automatically generated model are established. Then we validate the computing termination by the pushdown automaton. The experimental results show that this method can automatically generate penetration testing scheme to conduct the implementation of penetration testing.

Section II presents the recent research status of penetration testing and penetration graph. Section III establishes the analysis model about penetration testing scheme automatically generated. We analyze the model of the penetration testing process which exploits the vulnerability sequence. And then we get the penetration diagram and analyze the penetration testing path. From the penetration testing path we generate a series of test cases for the final penetration testing. Then we give the proof about the analysis model. Section IV gives the network penetration testing scheme for automatic generation of prototype system. Section V gives a specific experiment. Finally, a conclusion is presented.

II. RELATED WORK

Many researchers have reported the model and the standard about penetration testing. In 2008, the National Institute of Standards and Technology (NIST) published guidelines for information security testing techniques [1]. The policy proposed a four-stage model of penetration testing, such as planning, discovery, attack and reporting. In the attack phase, dynamic feedback information is transferred to the discovery tools. Through this process the testers can gain greater access iteratively. Canada's Information Technology Advisory Committee published a White Paper on penetration testing that is a series of activities undertaken to identify and exploit security vulnerabilities[2].Germany's Federal Office of Information Security proposed a five-stage penetration testing: preparation phase, reconnaissance stage, analysis information/risks phase, active intrusion attempts phase, final analysis phase [3].

In the penetration testing process, we need to model the mimicking attack and the attack path. And thus we perform penetration testing to provide guidance. Modeling penetration testing methods are divided into the following four categories: (1) based on Petri net modeling of penetration testing scheme, (2) based on attack net modeling of network penetration testing, (3) based on attack tree modeling of penetration testing scheme, (4) based on attack

978-0-7695-4501-1/11 $26.00 © 2011 IEEE
DOI 10.1109/ICCIS.2011.102

graph modeling of penetration testing scheme. Dahl OM, who use the interval timer Colored Petri Nets (Interval Timed Colored Petri Nets) model of penetration testing scheme [4]. It is strong that the hierarchical attack structure is description by Petri net. Petri net state of the system as a collection of attributes, and the set of attack moves as the set of transitions. Petri net can be used to establish the network attack graph, and then get the penetration path [5]. If we use penetration attack tree to establish the model, we need to define node in the attack tree and define the relationship between attack node [6]. Based on attack-graph, penetration testing scheme need consider the size of attack graph that is generated in the network and the method of the generation of the graph [7]. Based the attack graph, we can solve the problem about the penetration testing. However, we must modify the node in the attack graph. It is penetration graph that we propose. We can get the penetration path from which we analyze the edge and node in the penetration graph, and we can get the penetration testing case from the edge in the penetration graph.

It is always use the attack graph to generate the penetration graph. Cynthia Phillips first gives us the a graph-based system for network-vulnerability analysis. It concludes that an attack-graph analysis could work well for modeling enterprise-level (commercial or military) network risks [8].Laura P.Swiler show us a a tool that generates attack graphs. Each node in the attack graph represents a possible attack state. Edges represent a change of state caused by a single action token by the attacker or unwitting assistant [9]. To solve the scalable attack graph, many people propose the attribute-based attack graph [10-14]. In the attribute-based attack graph, node indicates the system attribute and atomic attack. Directed edge indicates the cause and effect between the nodes. Attribute-based attack graph is scalable method which can be used in the large size network. But the attribute-based attack graph indicates the attack path implicitly. The AGPTSAM for network which is established will follow the basic concept and activity in the section III.

III. AGPTSAM FOR NETWORK

In this chapter, we describe the analyze model of AGPTSON (Automatic Generation for Penetration Testing Scheme Oriented on Network). Firstly, we introduced the concepts of AGPTSONAM, and then, we describe the compute model of AGPTSONAM, and proved the validation for the termination by the state transition of AGPTSAM.

A. The definition of basic concepts

In this section, we explained the basic concepts related in this paper and put emphasis on the concept division of AGPTS analyzing activities and the formal description of those basic activities. This is to prepare for constructing the model.

Definition 1(Penetration testing case). The Penetration testing case is a 4-tuple $(goal, enviroment, input, expection)$ .Where $goal$ is the host's privilege which is uncertain in the penetration testing process, $environment$ is the vulnerability environment information, $input$ is the testing measure and $expection$ is the penetration expected result that can be certain privilege or the relation between the hosts.

Definition 2(Penetration testing time). The penetration testing time is a 2-tuple $\langle begin, end \rangle$. Where $begin$ is the time when the penetration testing begins, and end is the time when the penetration testing ends.

Definition 3(Penetration testing goal). It is the privilege of the host that is in the penetration testing target.

Definition 4(Penetration testing scheme). The Penetration testing scheme is a 4-tuple $(time, target, goal, process)$. Where $time$ is the penetration testing time, $target$ is the penetration testing target, $goal$ is the penetration testing goal, $process$ is the penetration testing case and the relationship of them.

In the Fig. 1 we can get the model's divided concept.

Definition 5(Plan activity)

$$\begin{cases} \Phi^{plan} = \{\varphi^{identify\_rules}, \varphi^{set\_goals}\}, \\ \varphi^{set\_goals} : pentest\_goal \mapsto penetration\_goal, \\ \varphi^{identify\_rules} : pentest\_rule \mapsto penetration\_rule; \end{cases} \quad (1)$$

Where, Φ^{plan} denotes a set of penetration testing plan activities, φ^{set_goals} and $\varphi^{identify\_rules}$ denotes activities of goal and rule setting. $pentest\_goal$ denotes the penetration goal that is described by the natural language, thus $penetration\_goal$ denotes the penetration goal that is described by the scheme description language. $pentest\_rule$ denotes the penetration rule that is described by the natural language. Thus $penetration\_rule$ denotes the penetration goal that can be used by the graph gentration activity.

Definition 6(Collect and scan information activity)

$$\begin{cases} \Phi^{collect\_scan} = \{\varphi^{feedback}, \varphi^{vul\_scan}\}, \\ \varphi^{feedback} : \{(info_{feedback}, VUL_{feature-database})\} \to \{(network_{info}, test\_case_{result})\}, \\ \varphi^{vul\_scan} : TOPOLOGY \times VUL_{feature-database} \to VUL_{scan}; \end{cases} \quad (2)$$

Where, $\Phi^{collect\_scan}$ denotes a set of the collecting and scanning activities, $\varphi^{feedback}$ and φ^{vul_scan} denotes activities of the target's information collection. $info_{feedback}$ denotes the feedback information, and $VUL_{feature-database}$ denotes the database about character of the

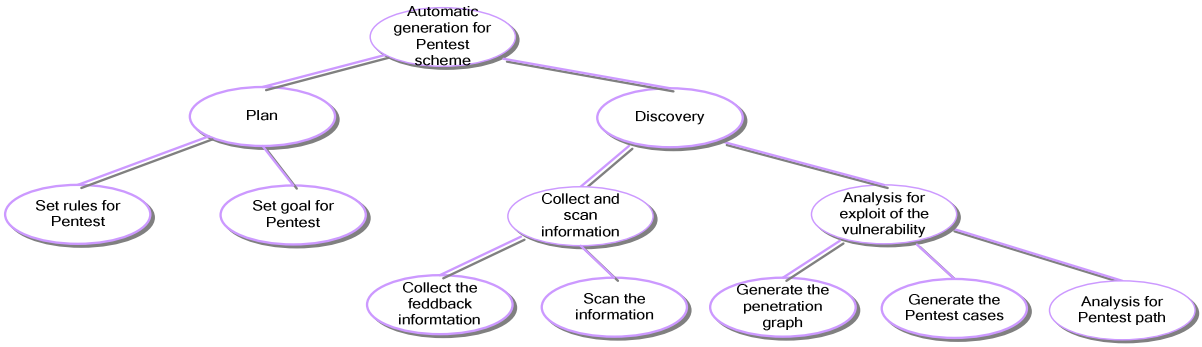

Figure 1. Concept division of AGPTS model for network

vulnerability. $network_{info}$ denotes the information of the network, and $test\_case_{result}$ denotes the result of the test case after it has been executed. VUL_{scan} denotes the vulnerabilities that was scanned.

Definition 7(Analysis for exploit of the vulnerability activity)

$$\begin{cases} \Phi^{vul\_analysis} = \{\varphi^{graph\_generation}, \varphi^{case\_generation}, \varphi^{path\_analysis}\}, \\ \varphi^{graph\_generation} : VUL \times VUL_{exploit\_rules} \to pentest\_graph, \\ \varphi^{case\_generation} : pentest\_graph \times VUL_{exploit\_database} \to TEST\_CASE, \\ \varphi^{path\_analysis} : pentest\_graph \times TEST\_CASE \to pentest\_scheme; \end{cases} \quad (3)$$

Where, $\Phi^{vul\_analysis}$ denotes a set of vulnerability analysis activities. $\varphi^{graph\_generation}$, $\varphi^{case\_generation}$ and $\varphi^{path\_analysis}$ denotes activities of the vulnerability analysis. $pentest\_graph$ denotes the penetration testing graph. $TEST\_CASE$ denotes the testing cases that are generated by the activity. $pentest\_scheme$ denotes the penetration testing scheme.

B. AGPTSAM for network

1) the automation model of AGPTSAM for network

We give pushdown automaton model (AGPTSAM) for computer to identify the input. AGPTSAM is constructed as follows:

$$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F) \quad (4)$$

Where, $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$, $\Sigma = \{a, f, x, t, e\}$, $\Gamma = \{T, G, M, Z_0\}$, $F = \{q_7\}$, q_0 is the initial state. q_7 is final state that the penetration testing scheme generate. q_1 is the state that the penetration testing target is determined. q_2 is the state that the penetration testing goal is determined. q_3 is the state that the feedback information is generated. q_4 is the state that the vulnerability is determined. q_5 is the state that the penetration graph is generated. q_6 is the state that the penetration testing case and penetration testing path is determined. Table I is the symbols in Σ and Γ as follows:

TABLE I. THE EXPLANATION OF SYMBOLS IN Σ AND Γ

Symbols	*The Meaning of Symbols*
a	Network address
f	Vulnerability Character
x	The binary group that consist the feedback and test case result
t	The binary group that consists the precondition and postcondition of the vulnerability
e	Vulnerability exploiting information
T	A stack symbol, denotes a penetration testing target
R	A stack symbol, denotes the setting rules before the penetration testing
G	A stack symbol, denotes penetration testing goal
M	A stack symbol, denotes penetration testing target and goal
Z_0	A stack bottom symbol, no meanings

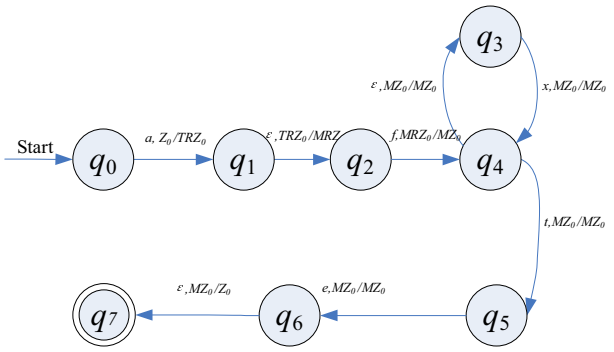

Figure 2. State transition graph of δ in AGPTSAM for network

According to the process of the state transition, the input sequence of AGPTSAM is:

$$INPUT = \{\omega \mid \omega \in (afxte)^n, n = 1, |\omega| = 5\}; \quad (5)$$

The explanations of input symbols can find from Table I.

2) The validation for the termination by the state transition of AGPTSAM

Theorem 1: AGPTSAM is expected to terminate for input, as follows:

$$(q_0, \text{INPUT}, Z_0) \vdash^* (q_7, \varepsilon, Z_0) \qquad (6)$$

Where, $INPUT = \{\omega | \omega \in (afxte)^n, n = 1, |\omega| = 5\}$;

Proof:

According the input $\omega = afxte$ then:

$$(q_0, \omega, Z_0) \vdash (q_1, fxte, TRZ_0) \vdash (q_2, fxte, MRZ_0)$$
$$\vdash (q_4, xte, MZ_0) \vdash (q_3, xte, MZ_0)$$
$$\vdash (q_4, te, MZ_0) \vdash (q_5, e, MZ_0)$$
$$\vdash (q_6, \varepsilon, MZ_0) \vdash (q_7, \varepsilon, Z_0)$$

This is $(q_0, \text{INPUT}, Z_0) \vdash^* (q_7, \varepsilon, Z_0)$, AGPTSAM closes to final state.

IV. THE PROTOTYPE SYSTEM OF AGPTSAM FOR NETWORK

Based on the AGPTSAM for network, this section gives prototype system of AGPTSAM. Fig.1 gives the structure of prototype system.

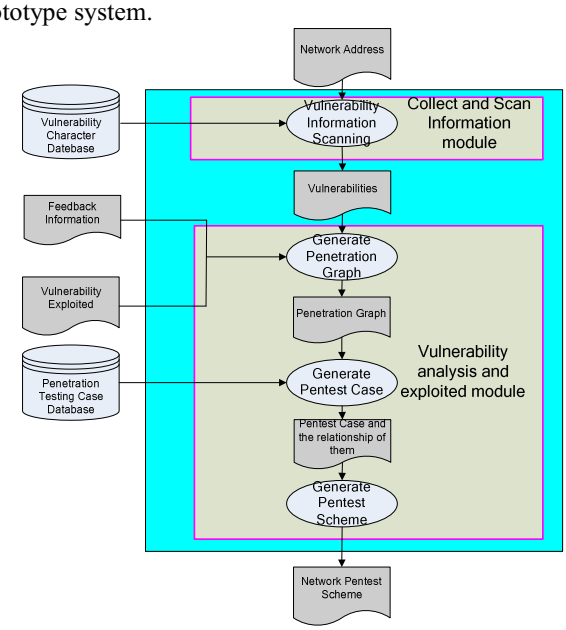

Figure 3. The structure of the prototype system

This prototype system includes two modules: Collect and Scan information and Vulnerability analysis and exploited. In the Collect and Scan information module, we can get the network vulnerability information and the feedback information about the implementation of penetration testing. In the Vulnerability analysis and exploited module, we can get the measure that vulnerability is exploited by and we also can get the relationship between the vulnerabilities. Then the penetration-graph can be got by the vulnerabilities' exploiting relationship. So we analyze the graph and get the penetration testing path and the testing cases in the network.

In the generation of the penetration graph, we use the backward matching method. Because the penetration testing goal is specific, we can take it as the initial node in the penetration graph.

After the generation of the penetration graph, we must detect the cycle path in the digraph. It uses the topological sorting to solve this problem.

V. EXPERIMENT

In this section, we show the implementation process of the prototype system to validate the function of this system. From this experiment, the input of the system is the scope of the IP address.

The experiment has used the topology as follows:

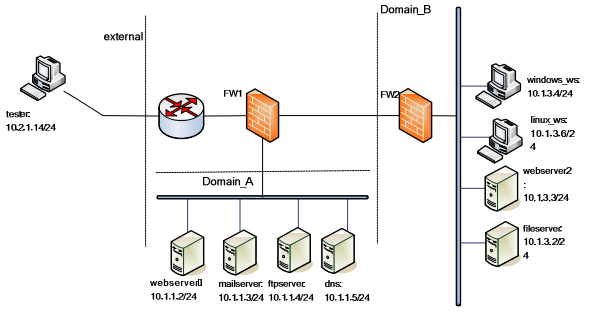

Figure 4. The topology of the experiment

The network has been divided into two subnets by the firewall. The external hosts have just access the subnet A, and have not access the subnet B. But the hosts in the subnet A have access into the subnet B. We suppose that the tester is outside of the firewall. And the penetration testing target is IP address which is from 10.1.1.2 to 10.1.1.5 and from 10.1.3.2 to 10.1.3.6. The penetration testing goal is to gain the root privilege of webserver2.

The penetration rules consist of four kinds. Firstly, we choose the attack tools that must be approved by the tested organization. Secondly, we can't carry out the attack that leads to the business process termination. Thirdly, the scanning activity can't affect the system and network. Fourthly, we must limit the penetration time.

We use Nessus as the vulnerability scanning. And we need write the script using NASL. Its scanning results(vulnerabilities) as follows:

TABLE II. THE VULNERABILITY INFORMATION

Vulnerability ID	Precondition	Postcondition
CVE-2002-0364	(Windows,IIS)	Privilege(Admin)
CVE-2004-0330	(Ftpserver,Serv-U)	Privilege(Root)
CVE-2001-0439	(linux,LICQ)	Privilege(User)
CVE-2003-0542	(Linux,Apache)	Privilege(Root)
CVE-2002-0509	(Linux,Oracle)	Deny of Service

From the four penetration rules, we can't lead to the business process termination, so we give up the vulnerability CVE-2002-0509 and we can't exploit it.

We can get the relation between the hosts. And its results as follows:

TABLE III. THE RELATION BETWEEN HOSTS

Relation	The relation between Hosts
Service	WebService(0,1)
Service	Ftp(0,3)
Connection	LICQ(1,6)
Connection	LICQ(3,6)
Service	WebService(6,7)
Service	Ftp(1,3)

According the method of penetration graph generation, we use natural number named the host in the network. Just like the host web server(10.1.1.2), we name it host 1, and so on. We can get the penetration graph as follows:

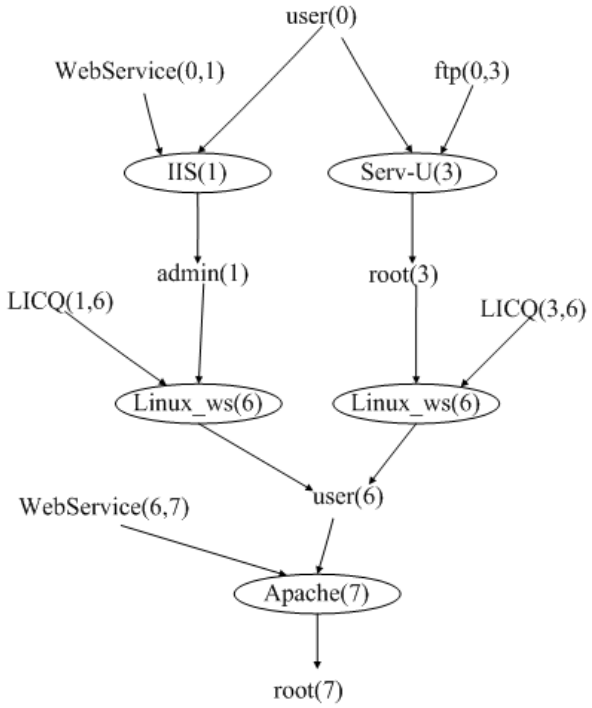

Figure 5. The penetration graph that is generated

Then we get the penetration path from this graph, we can get two paths to the user(6) . They are Path 1:,

$\{\text{user(0),WebService(0,1),IIS(1)}\} \rightarrow \{\text{LICQ(1,6),admin(1),Linux\_ws(6)}\}$,

$\{\text{LICQ(1,6),admin(1),Linux\_ws(6)}\} \rightarrow \{\text{user(6),WebService(6,7),Apache(7)}\}$.

And Path 2:

$\{\text{user(0),ftp(0,3),Serv-U(0,3)}\} \rightarrow \{\text{LICQ(3,6),root(3),Linux\_ws(6)}\}$,

$\{\text{LICQ(1,6),admin(1),Linux\_ws(6)}\} \rightarrow \{\text{user(6),WebService(6,7),Apache(7)}\}$.

And the penetration testing case can be shown as follows:

TABLE IV. PENETRATION TESTING CASE

Order Number	Penetration Testing Case
1	(<IP(10.1.1.2),privilege>,CVE-2002-0364,Overflow,Privilege(admin))
2	(<IP(10.1.1.4),privilege>,CVE-2004-0330,Overflow,Privilege(root))
3	(<IP(10.1.3.6),privilege>,CVE-2001-0439,Overflow,Privilege(user))
4	(<IP(10.1.3.3),privilege>,CVE-2003-0542,Overflow,Privilege(root))

From the table IV, we can get the penetration testing scheme as follows:

Scheme{

Penetration Target: 10.1.1.2-10.1.1.5 AND 10.1.3.2-10.1.3.2.6;

Penetration Goal: the root privilege of Webserver2 (10.1.3.3);

Penetration Time: 20:00-24:00;

Penetration Testing Case Path 1:

{

(<IP(10.1.1.2),privilege>,CVE-2002-0364,Overflow,Privilege(admin));

(<IP(10.1.3.6),privilege>,CVE-2001-0439,Overflow,Privilege(user));

(<IP(10.1.3.3),privilege>,CVE-2003-0542,Overflow,Privilege(root));

}

Penetration Testing Case Path 2:

{(<IP(10.1.1.4),privilege>,CVE-2004-0330,Overflow,Privilege(root));

(<IP(10.1.3.6),privilege>,CVE-2001-0439,Overflow,Privilege(user));

(<IP(10.1.3.3),privilege>,CVE-2003-0542,Overflow,Privilege(root));

}

}

Where, we have two penetration testing paths. Then, we give the scheme to the implement system of the penetration testing. The penetration report will be got.

VI. CONCLUSION

In this paper, we present a new model to generate the penetration testing scheme for the network automatically. The AGPTSAM for network is presented, which consists of the basic activity concepts and the automation model of the AGPTS for network. In the model, the generation of penetration graph and the generation of the penetration testing case are the main idea for the prototype system. And we design the prototype system to verify the model. Based on the experimental verification for the prototype system, the penetration graph can be generated, and it can be used to gain the penetration path. Then from the penetration graph we show the penetration testing cases. To sum up, this paper achieves the goal to figure out the penetration testing scheme generation automatically.

In the future, we will advance the existing penetration testing scheme generation algorithm in order to make it much more appropriate to guide the implement of the penetration testing and we could do more experiments to judge the validity of the penetration testing scheme by the platform of Penetration Testing.

ACKNOWLEDGMENT

This work is supported by two project: The Co-Funding Project of Beijing Municipal education Commission under Grant No.JD100060630 and the National Foundation Research Project.

We would like to express gratitude to Mr. Chao Yuan, Mr. Junshun Hu for many helpful discussions. Detailed comments by all of them greatly improve the paper.

REFERENCES REFERENCES

[1] Karen Scarfone. Technical Guide to Information Security Testing and Assessment. http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf.

[2] The Canadian Institute Of Accountants. Using an Ethical Hacking Technique to Assess Information Security Risk. http://www.provenprivatenetworks.com/cica.

[3] BSI. Study A Penetration Testing Model. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/Penetration/penetration_pdf.pdf.

[4] L. Jehyun, L. Heejo and H. P. In, "Scalable attack graph for risk assessment," in Information Networking, 2009. ICOIN 2009. International Conference on, 2009, pp. 1-5.

[5] J. P. McDermott, "Attack net penetration testing," in New security paradigms, Ballycotton, County Cork, Ireland, 2001, pp. 15-21.

[6] N. Zhu, X. Chen, Y. Zhang, and S. Xin, "Design and Application of Penetration Attack Tree Model Oriented to Attack Resistance Test," in Computer Science and Software Engineering, 2008 International Conference on, 2008, pp. 622-626.

[7] Cuiyin, Zhang lijuan, and Wuhao,"Automatic generation method for penetration test programs based on attack graph," Journal of Computer Applications, 2010, pp. 53-57.

[8] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in New security paradigms, Charlottesville, Virginia, United States, 1998, pp. 71-79.

[9] L. P. Swiler, C. Phillips, D. Ellis, and S. Chakerian, "Computer-attack graph generation tool," in DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01. Proceedings, 2001, pp. 307-321 vol.2.

[10] Xinming Ou,Wayne F. Boyer, and Miles A. McQueen, "A scalable approach to attack graph generation," in Audit and Control ACM Special Interest Group on Security,Association for Computing Machinery. Alexandria, Virginia, USA,2006,pp. 336-345.

[11] Ammann P, Wijesekera D, and Kaushik S. "Scalable, graph-based network vulnerability analysis," in Proc. of the 9th ACM Conf. on Computer and Communications Security. New York, 2002,pp. 217−224.

[12] Jajodia S, Noel S, and O'Beny B, "Topological analysis of network attack vulnerability," in Proc. of the Managing Cyber Threats: Issues, Approaches and Challenge, Netherlands, 2003,pp. 854-890.

[13] Wang L, Noel S, and Jajodia S, "Minimum-Cost network hardening using attack graphs," in Computer Communications,2006,pp. 812−3824.

[14] Noel S, Jajodia S, O'Berry B, and Jacobs M,"Efficient minimum-cost network hardening via exploit dependency graphs," in Proc. of the 19th Annual Computer Security Applications Conf. Las Vegas, USA, 2003, pp. 86−95.