BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **13**, Special Issue

Sofia • 2013

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2013-0037

Network Threat Identification and Analysis Based on a State Transition Graph

Huiying Lv, Yuan Zhang, Jie Wang

School of Management, Capital Normal University, Beijing, China
E-mails: lvhy999@163.com yuaner.cn@gmail.com wangjie@cnu.edu.cn

Abstract: *With the rapid popularity of Internet and information technology, local area network is becoming insecure. Along with the improving advantages, security threats are emerging continually and bringing great pressure and challenges. An identification and analysis method for network real-time threats is proposed to accurately assess and master the current network security situation, and thereby preferably guide a dynamic defense. This method recognizes the current threats and predicts the subsequent threats by modeling attack scenarios and simulating attack state transferring. The threat identification model is called Attack State Transition Graph and Real-Time Attack State Graph, which is constructed by an Expanded Finite-State Automata. Based on the former possible threat paths, the state transitions can be illustrated and based on the latter, actually successful threats and threat paths are described. Then a threat identification algorithm is presented based on the above model. With this algorithm, various invalid threats are filtered; current valid threats are obtained by correlating the dynamic alarms with a static attack scenario. Further on, combining the Attack State Transition Graph with a Real-Time Attack State Graph, a possible next threat and a threat path can be identified and an attack target can also be predicted. Finally, the simulated results in an experimental network verify the feasibility and validity of the model and algorithm. This method provides a novel solution to evaluate and analyze the network security situation.*

Keywords: *Network security, threat, attack, state transition, graph.*

1. Introduction

Because of dynamics and openness in cyberspace, the environmental characteristics of an information network are changing ceaselessly, and both the attacker and defender would adjust their strategies, which makes the attack-defence confront situation also constantly changing. When the attacker detects new vulnerabilities and target, he/she will adjust their attack methods and strategy. At the same time, defenders have to timely recognize the attackers' intention and the new threats in order to make an efficient decision. Literature [9] enumerates some common changes of the environmental characteristics:

1. New vulnerability or weakness is found by scanners;
2. New security event is reported by IDS;
3. Add or delete network services;
4. Reconfigure and add fireworks;
5. Modify network configuration parameter.

Therefore, with the changes of the network environment, the security situation and level also transform, the old defensive security measures might become impotent. So in order to improve the dynamic response capabilities, how to identify a real-time threat situation, precisely evaluate and timely respond to a security risk has become an important and urgent issue.

Traditional methods identify and evaluate the network security threat with either alarms correlation analysis or attack graph. In [2], in MIRADOR system Cuppens achieved alarms correlation and threat identification analysis by modelling the attack activities. He extracted a single attack activity to match its precondition and consequence, and then correlated the previous action and subsequent action. In SATA system (Security Alerts and Threat Analysis), in N i n g [5] proposed a new method of learning a multi-stage attack through a mining attack sequence to recognize the attacker's high-level strategies and predict the attack intentions. The method is easy to implement and can be used to detect new multi-stage attack strategies. Literature [12] proposed a network security situation identification and analysis system based on UCLog+. This system can find new threats and analyze the attack source by incident storage, querying and correlation.

In [6] Attack Graph is a predicted attack path by correlating vulnerabilities, threats and connectivity between hosts, and can be automatically generated by model checking algorithms. In N o e l [8] applied adjacency matrix clustering to network attack graphs for threats correlation, prediction and hypothesizing, which have quadratic complexity in the size of an attack graph. In [7] and L i p p m a n n [4] proposed NetSPA based on an attack graph and constructed by analyzing firewall rules and vulnerabilities; the tool can obtain reachability from an already compromised host to the target host, so be used to measure and maintain network security. Literature [11] presented a quantitative threat modelling method based on Attack Path Analysis (T-MAP), which quantifies security threats by calculating total severity weights of relevant attack paths for Commercial off the Shelf (COTS) systems. Literature [10] combined Object-oriented Technique to propose a vulnerability relation model based on an extended time Petri-net, and present a non-

target oriented network threat analysis method based on improved Dijkstra algorithm.

In MIRADOR and SATA, threat identification and analysis is realized by extracting threat behaviours from IDS alarms to construct an attack scenario. These methods seldom consider the influence of changing the network environment and security strategy to threat behaviours. So accuracy and efficiency of threat identification cannot be ensured and subsequent threat and attack intension is difficult to deduce. The systems AG, TVA, NetSPA and T-MAP either only correlate static information including vulnerabilities, network topologies and firework rules, etc., or fail to consider the dynamic nature of the intrusion process, so neither can obtain a real-time threat state, nor dynamically evaluate and analyze threat severity.

In this paper we combine different environment factors, consider both real-time attack behaviours and response behaviours, propose a dynamical identification method for current threat state based on invasion scenario simulating. The method first correlates network topology, vulnerabilities and firewall rules, as well as attack knowledge to generate Attack State Transition Graph in [3], and then by correlating real-time alarms from IDS constructs a Real-time Attack State Graph. This model visually describes attacker's intruding process and state transforming, depicts real-time transition of vulnerability state under the influence of attack actions and response actions. Then a current threat state is identified and perceived by matching the above static scenario and be used to threats estimation and analysis, further to predict a next threat and target. This method also provides an useful evidence and guidance for intrusion response and security decision. Finally, this method is validated in an example network environment.

The rest of this paper is organized as follows. The next section defines the model for threat identification and analysis and other correlative conceptions. Section 3 presents real-time threat identification process and algorithm and also analyzes its efficiency. In Section 4 a representative virtual network environment was given to illustrate the applicability and validity of the above model and corresponding algorithm. Finally, Section 5 concludes the paper and gives future directions.

2. Threat identification model

Network security is not independent and stationary, but is in a dynamic course of attack-defence countermeasure under Internet environment. In a given cyberspace and given time, the behaviours of the attacker and behaviours of the defender interact and then turn the system into a specific state. Security countering is essentially a dynamic game between attackers and defenders. All behaviours of the network system can be considered as state transforming. Therefore, the whole network system can be considered as a finite-state system, of which state transforming can be described by Finite-State Automata.

The network security countermeasure model is defined as an Expanded Finite-State Automata (EFSA), which is described as

(1) $$M_A = (S_A, E_A, S_0^A, \delta_A, \rho, S_F^A, S_S^A),$$

where S_A is a set of security states; E_A is a set of attack actions; S_0^A is a set of initial states; S_F^A is a set of final states, if $s_f \in S_F^A$, then $\forall a \in E_A$, $\delta(s_f, a) \notin S_A$; S_S^A is a set of successful states; $\delta_A : S_A \times E_A \to S_A$ is a set of state transitions; $\rho : S_A \to S_A$ is probability of state transition.

No matter what consummate security protection technology is applied, threats always exist and vary, to completely prevent threats sounds just like walking on water. To correctly and timely master attacker's information is a key for the defender to make a response decision. By simulating the intruding process and constructing a real-time state transition scene, the defender can master the current safety situation, estimate attacker's target and predict potential threats, and then provide an important basis for proactive security defend.

Definition 1. Attack State Transition Graph. Under a given system configuration and security policy, the state transition graph of network security counter measure model $M_A = (S_A, E_A, S_0^A, \delta_A, \rho, S_F^A, S_S^A)$ is called Attack State Transition Graph (ASTG), described as

(2) $$\text{ASTG}_M = (S_A, V_A),$$

where S_A is a set of nodes, the node is defined as a security state and is described with attributes of the attacker; $\forall s \in S_A$, s = {*Host_id*, *Privil_list*, *Vuls_list*, *Conn_list*, *Servs_list*}, of which *Host_id* is the host where the attacker currently situates in, *Privil _ list* $\subseteq$ {*Root*, *Privileged user*, *User*, *Access*, *none*} is a set of all privileges obtained by the attacker, *Vuls_list* is a list of all vulnerabilities in *Host_id*, *Conn_list* is a list of all connections from *Host_id* to other hosts, *Servs_list* is a list of all open network services on *Host_id*.

$V_A \subseteq S_A \times E_A \times S_A$ is a set of directed edges in ASTG$_M$. An edge in V_A is a state transition that arose by a successful threat.

Definition 2. Precondition state and Subsequence state. $\forall v_a = \langle s_i, a, s_j \rangle \in V_A$, $s_i, s_j \in S_A \Leftrightarrow \exists$a $\in E_A$, making $\delta_A(s_i, a) = s_j$; s_i is defined as a precondition state of *a*, denoted by a *pre-state*(*a*), s_j is defined as a subsequence state of *a*, denoted by *post-state*(*a*).

Definition 3. Precede event and Succeed event. In ASTG$_M$, the attack event reaching *s* is defined as a precede event of *s*, all precede events are denoted as a set *precede*(*s*); an attack even starting from *s* is defined as a succeed event of *s*, all succeed events are denoted as a set *post-state*(*s*).

Definition 4. Attack Target. The attack targets can be described by CTL (computational tree logic) [1]:

(3) $$G_{\text{attack}} = \{g_{\text{attack}} = E_F(\varphi)\}, \text{ and } s_{g\text{–attack}} \in S_S^A.$$

Definition 5. Attack Path. In ASTG_M, if $\forall s_0 \in S_0^A$, $\exists \omega \in E_A^*$, $\omega = \{a_1, a_2, \ldots, a_{k-1}\}$, making $\delta(s_0, \omega) = s_f$, $s_f \in S_F^A$, and for all $1 \le i \le k$, $a_i \in E_A$, $s_i \in S_A$, $\delta(s_{i-1}, a_i) = s_i$, we define the attack path as a sequence consisting of finite states and events from s_0 to s_f, whose origin is s_0 and the terminal is s_f, and expressed as

$$\text{path}(s_0, \omega) = (s_0, a_1, s_1, a_2, \cdots, a_k, s_f). \tag{4}$$

Definition 6. Real-time Attack State Graph. Define a new directed graph $\mathrm{RASG}_M = (\overline{S_A}, \overline{V_A})$, where $\overline{S_A}$ is set of nodes including all the security states having appeared, and $\overline{V_A}$ is a set of directed edges including all successful attack behaviors having been dynamically detected.

Our approach depends on an explicit assumption of monotonicity, in essence, the attacker never needs to backtrack to reach the same target. Based on this assumption, ASTG never cycles. ASTG can be generated using an improved depth first search algorithm, which has been provided and validated in our previous research achievements in [3].

Next, based on the constructed state transition graph, correlate real-time threats with the status attribute in ASTG, and correlate those happened and happening activities with the simulated intrusion route, then generate RASG. Furthermore, based on ASTG and RASG, the current security state can be identified, subsequent threats and possible paths can also be predicted, thus attack intension and target can be deduced.

3. Threat identification algorithms

3.1. Occurrence testing function

Set a testing function $P(.)$ = {true, false} to mark occurrences of threat behaviors and states. If a threat behavior or state has occurred or appeared, the value of the testing function is true, otherwise false, what is more, for those threats that repeatedly occur, set a timestamp function $T(.)$ to record the latest moment when the same alerts and attack states are detected. The two functions are defined as:

$$P(a) = \begin{cases} \text{true}, a \in \mathrm{RASG}_M \\ \text{false}, a \notin \mathrm{RASG}_M \end{cases}, \quad a \in E_A,$$
$$P(s) = \begin{cases} \text{true}, s \in \mathrm{RASG}_M \\ \text{false}, s \notin \mathrm{RASG}_M \end{cases}, \quad s \in S_A; \tag{5}$$

$$T(a) \in [0, \text{curenttime}], a \in \overline{E_A}, \ \forall \hat{a}, a = \hat{a} \Rightarrow \hat{a}\_\text{time} \in [0, T(a)],$$
$$T(s) \in [0, \text{curenttime}], s \in \overline{S_A}, \ \forall a \in \text{precede}(s) \Rightarrow T(a) \in [0, T(s)], \tag{6}$$

where currenttime is the present time, $\hat{a}$ is the real-time intrusion alarm.

3.2. Attack matching window

Because some attacks repeat with very high frequency, such as network probes, syn flooding and ping of death, in order to increase the efficiency of matching real-time attacks, set a time interval to correlate the real-time alarms, which is called an Attack Matching Window(AMW) and is represented by $\partial\tau$. All analogical attacks detected in a given $\partial\tau$ may be considered as repeated alarms and negligible for processing. So for any two successive intrusions, there always is

$$|\hat{a}_x\_\text{time} - \hat{a}_y\ \_\ \text{time}| > \partial\tau .$$

Suppose that the intrusion detection devices do not leave out any alarm, when correlating the real-time threat activities with ASTG, several possible situations may happen, which is shown in Fig. 1, where

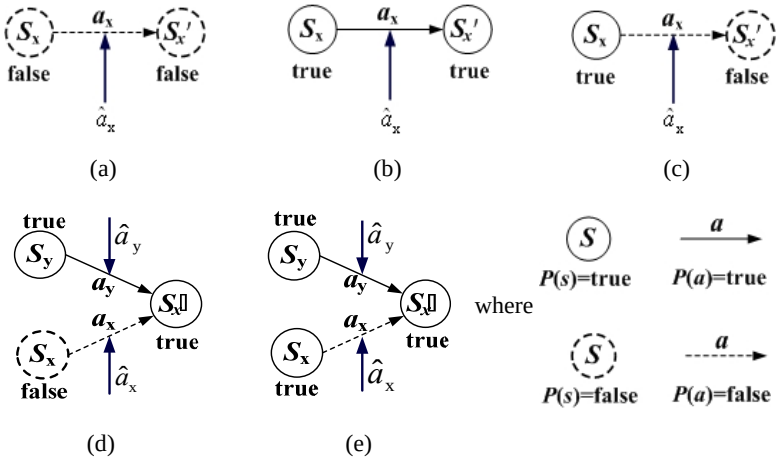

Fig. 1. Several possible situations by correlating analysis

$\hat{a}_x$ is a real attack in current time;

in (a) and (d), the precondition state is not met, so $\hat{a}_x$ is unsuccessful and can not be inserted into $RASG_M$;

in (b), the state transition is caused by a_x and its subsequence state has occurred;

in (d) and (e), a_x does not succeed, but its subsequence state has occurred, which shows some other attack, such as a_y, has occurred and brought about the state but is not detected.

3.3. Threat identification process and algorithm

The whole flow of real-time threat identification and analysis is illustrated in Fig. 2.

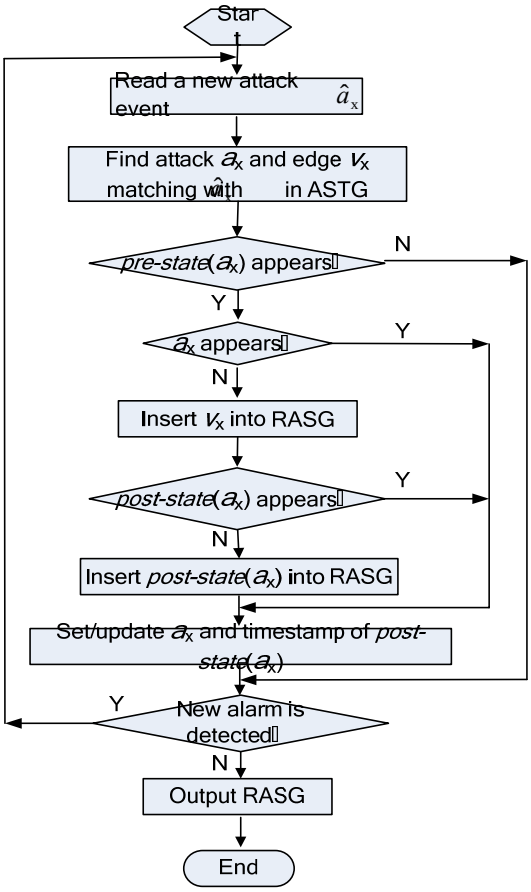

Fig. 2. Flowchart of real-time threat identification and analysis

The major steps for real-time threat identification algorithm *StateIdentification* (.) are specified as follows:

Step 1. Read a new attack event $\hat{a}_x$, and in the generated ASTG$_M$ = (S_A, V_A) find a_x that matches $\hat{a}_x$ and the state transition $v_x = \langle s_x, a_x, s_x' \rangle$, s_x =pre-state(a_x), s_x'= post-state(a_x).

Step 2. If $P(s_x)$ = false, that is to say the precondition state of a_x is not met, and s_x cannot successfully transfer the next state, and then go to Step 6.

Step 3. If $P(s_x)$ = true, and $P(a_x)$ = true, the state transition $\delta(s_x, a_x) = s_x'$ has been achieved, which shows a_x is a repeat alarm, and then go to Step 5.

Step 4. If $P(s_x)$ = true, and $P(a_x)$ = false, the state transition

$\delta(s_x, a_x) = s_x'$ occurs for the first time, so v_x can be inserted into RASG$_M$, and set $P(a_x)$ = true; If for the subsequence state of a_x, $P(s_x')$ = false, insert s_x' into RASG$_M$, and set $P(s_x')$ = true.

Step 5. Renew a_x and the timestamp of its subsequence state s_x'.

Step 6. If detecting a new alarm after Attack Matching Window, return to Step 1, otherwise put out RASG$_M$.

Step 7. Search RASG$_M$ for all final states to find the state with the latest timestamp $T(.)$ and regard it as the current state.

4. Experiment and analysis

4.1. Experimental network environment

Next we use an experimental network shown in Fig. 3 to verify our method. A firewall separates the external from internal network. An Intrusion Detection System (IDS) monitors the network traffic between internal hosts and outside. There are four protected hosts in the internal network: Web server H_1, clients H_2 and H_3, Database server H_4, whose configurations are enumerated in Table 1.

A simulating intruder launches his attacks from host H_0 outside the firewall into the internal, whose eventual goal is to disrupt the function of H_4 and to filch data on H_4, for which, the intruder needs root access to the database on H_4. His attack goal can be denoted as

(7) $g_{attack} = E_F(\text{privil}(\text{intruder}, H_4) = \text{root})$.

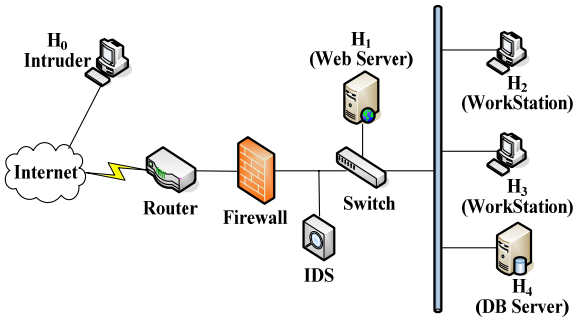

Fig. 3. Topological graph of experimental network

Table 1. Initial configuration of hosts

Host	Host_Type	Host_Os	Host_Servs	Host_Vuls
H_1	Web Server	Windows 2003 server	IIS, http	CVE-2002-0364(v1)
H_2	Workstation	Windows XP	RPC, Rlogin	CVE-1999-0651(v5) CVE-2005-0688(v6)
H_3	Workstation	Windows XP	http	CVE-2002-0193(v2)
H_4	DB Server	Red Hat Linux 8.0	Rlogin, RPC, DataBase, LICQ	CVE-2001-0439(v3) CVE-2002-0004(v4) CVE-2001-0851(v7) CVE-2001-0352(v8)

Then the initial connectivity relations among all the hosts are shown in Table 2.

Table 2. Connectivity relations matrix in a network

Host	H_0	H_1	H_2	H_3	H_4
H_0	{Localhost}	{http,IIS}	Φ	Φ	Φ
H_1	Φ	{Localhost}	{http,Rlogin}	{http}	{http}
H_2	Φ	{http,IIS}	{Localhost}	{http}	{http,Rlogin, LICQ,RPC}
H_3	Φ	{http,IIS}	{http,Rlogin}	{Localhost}	{http,LICQ}
H_4	Φ	{http,IIS}	{http,Rlogin}	{http}	{Localhost}

By associating the network configuration with attack rules, we get all available threats, whose rules corresponding to different vulnerabilities are listed in Table 3.

Table 3. Rules of available attack actions

Id	Action_Name	Pre_Vuls	Attack_Priority	Complexity
a_1	IIS buffer overflow	V_1	High	E_6
a_2	Scripting exploit	V_2	Medium	E_4
a_3	LICQ gain user	V_3	Medium	E_5
a_4	Local buffer overflow	V_4	High	E_6
a_5	Probe(Finger, showmount, rpcinfo)	V_5	Low	E_3
a_6	Land	V_6	Medium	E_5
a_7	FakeConnection	V_7	Medium	E_2
a_8	RemoteCommand RPC Dcom buffer overflow	V_8	Medium	E_6

4.2. Constructing attack state transition graph

The probable pervading processes of an intruder are simulated by ASTG in Fig. 4. By looking for the ASTG, eighteen different intruding paths can be obtained from the initial state S_0 to the goal state S_{10}.

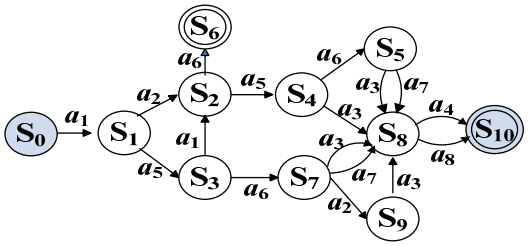

Fig. 4. ASTG of the experimental network

4.3. Constructing real-time attack state graph

By correlating the real-time alarms with ASTG, RASG for the experimental network is generated, shown in Fig. 5, in which the attacker has currently reached state S_7. From this it is easy to deduce the attack source and obtain the achieved

intrusion path (S_0, a_1, S_1, a_5, S_3, a_6, S_7). Apparently, the path includes all compromised hosts and all successful attacks.

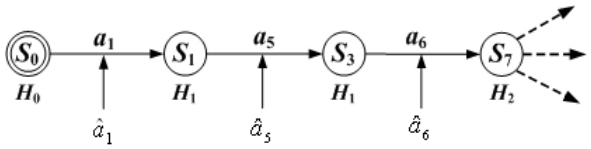

Fig. 5. RASG based on the real-time attack recognition

4.4. Real-time threat identification and analysis

Further on, based on the generated ASTG in Fig. 4, expand forward from state S_7, as shown in Fig. 6, it is obvious that there are three threat activities (a_2, a_3, a_7) that possibly occurred from the current state. And from the current state, six possible subsequence paths are deduced, as shown in bold type in Table 4.

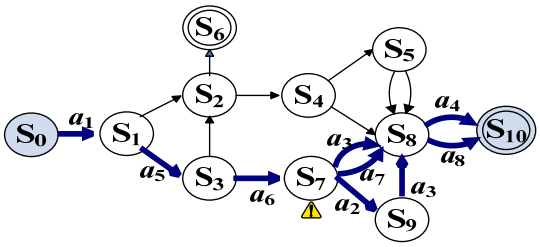

Fig. 6. Threat identification and analysis based on the current state

Table 4. Possible attack paths predicted based on the current state

Path13	(S_0, a_1, S_1, a_5, S_3, a_6, S_7, a_3, S_8, a_4, S_{10})
Path14	(S_0, a_1, S_1, a_5, S_3, a_6, S_7, a_3, S_8, a_8, S_{10})
Path15	(S_0, a_1, S_1, a_5, S_3, a_6, S_7, a_7, S_8, a_4, S_{10})
Path16	(S_0, a_1, S_1, a_5, S_3, a_6, S_7, a_7, S_8, a_8, S_{10})
Path17	(S_0, a_1, S_1, a_5, S_3, a_6, S_7, a_2, S_9, a_3, S_8, a_4, S_{10})
Path18	(S_0, a_1, S_1, a_5, S_3, a_6, S_7, a_2, S_9, a_3, S_8, a_8, S_{10})

5. Conclusion

In this paper a new method is presented to overcome some defects of the existing static security defense technologies, which is unable to dynamically identify the real-time security states and threats, nor predict the impendent threats and future security situation. This method correlates the real-time threat events with a static and complete attack scenario based on ASTG constructed in a previous work, so it can recognize that network entities have been intruded and is being intruded, and also can predict the next threat and subsequent intrusion path. The identification and analysis results not only illustrate the intruding process, present the security situation and dynamic trend at macroscopic level, but also accurately identify the real-time threat location, target, and threat path at microscopic level.

Meanwhile this method integrates different dynamic information obtained

from different security scanning and detection devices, such as system vulnerabilities, intrusion alarms, network configuration, and so on. Thus the identified results accurately and objectively provide useful evidence and guidance for intrusion responding and security strategies decision, and embody the idea of active and dynamic defence.

Our next study is to quantitatively assess real-time security threats based on recognizing real-time threats and security state.

Acknowledgments: The authors are grateful to all anonymous reviewers for their valuable comments. We would like to specifically thank to all people that provided support and made suggestions throughout the duration of this research. This work is funded by the Scientific Research Program of Beijing Municipal Commission of Education (Grant No KM201110028019 and No KM201310028020) and China Information Technology Security Evaluation Center CNITSEC-KY-2012-006/1.

References

1. Clarke, E. M., E. A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching Time Temporal Logic. – In: Proceedings of Logic of Programs, Vol. 131 of Lecture Notes in Computer Science, Workshop, Yorktown Heights, New York, 1981, 52-71.
2. Cuppens, F., F. Autrel, A. Miege, S. Benferhat. Recognizing Malicious Intention in an Intrusion Detection Process. – In: Second International Conference on Hybrid Intelligent Systems. Santiago, 2002.
3. Huiying, Lv, R. Wang. Identifying Attacking Process Based on Attack State Transition Graph. – In: 3rd International Conference on Computer and Network Technology (ICCNT'2011), TaiYuan, China, Vol. 4, 26-28 February 2011, 395-400.
4. Williams, L., R. Lippmann, K. Ingols. GARNET: A Graphical Attack Graph and Reachability Network Evaluation Tool. – In: VizSec'2008, LNCS 5210, 2008, 44-59.
5. Ning, P., D. Xu. Learning Attack Strategies from Intrusion Alerts. – In: Proceedings of the 10th ACM Conference on Computer and Communications Security, Washington, DC, USA, 2003, 200-209.
6. Sheyner, O., J. Haines, S. Jha, R. Lippmann, J. M. Wing. Automated Generation and Analysis of Attack Graphs. – In: Proceedings of the 2002 IEEE Symposium on Security and Privacy, Oakland, CA, 2002, 273-284.
7. Lippmann, R., K. Ingols, C. Scott et al. Validating and Restoring Defence in Depth Using Attack Graphs. – In: Military Communications Conference, Washington, DC, USA, 2006.
8. Noel, S., S. Jajodia. Understanding Complex Network Attack Graphs through Clustered Adjacency Matrices. – In: Proceedings of the 21st Annual Computer Security Applications Conference, Tucson, AZ, 2005, 160-169.
9. Martel, S. A New Model for Computer Network Security Risk Analysis. PhD Paper, Department of System Computer Engineering, Carleton University, Ottawa, Ontario, May 2002.
10. Wang, C., G. Huang. Network Threat Analysis Based on Vulnerability Relation Model. – Journal of Computer Applications, Vol. **30**, 2010, No 11, 3046-3050.
11. Chen, Y., B. Boehm, L. Sheppard. Value Driven Security Threat Modelling Based on Attack Path Analysis. – In: 40th Hawaii International Conference on System Sciences, 2007, 280-288.
12. Yureik, W., C. Abad. UCLog+: A Security Situational Identification System for Incident Storage, Querying, and Correlation. – In: Proceedings of the 14th International Conference on Telecommunication Systems Modelling and Analysis (ICTSM), 2006, 316-322.