International Journal of Security and Its Applications
Vol. 10, No. 4 (2016), pp.51-62
http://dx.doi.org/10.14257/ijsia.2016.10.4.06

Real-time Network Attack Intention Recognition Algorithm

Qiu Hui and Wang Kun

Zhengzhou Institute of Information Science and Technology, Zhengzhou, China
pioneerqh@126.com

Abstract

Attack intention recognition is to reason and judge the goal of attackers according to attack behavior and network environment. In order to deal with the dynamical character of offense-defense confrontation, a dynamical real-time network attack intention recognition algorithm was proposed. By correlating real-time security alerts and vulnerabilities, we recognized the spread route and stage of attacks based on graph theory and probability theory. Then we identified the attack intention and predicted the possible transition of attacks, combined with network connectivity relationship. A simulation experiments for the proposed network attack intention recognition algorithm is performed by network examples. The experimental results show that the proposed method can be more accurately identify attack intention and fully predict the post stage of attacks.

Keywords*: attack intention; pattern recognition; network security; multi-stage attack; state transition*

1. Introduction

Intrusion intention recognition is to interpret and judge the purpose, vision and intention of attackers through analyzing a large number of low-level alarm information, which is to give a reasonable explanation of a large number of attack data. Identifying attack intention can determine the real purpose of attackers and predict the subsequent attack behavior, which is the premise and foundation of threat analysis and the important part of network security situation awareness. It has become a hot topic in the field of network security.

At the earliest, the research of intention recognition was carried out in the field of artificial intelligence. Intention of agent is the chosen planning route to achieve a goal [1, 2], which role is to guide the rational decision-making and plan future behavior. Intention recognition is the process of apperceiving and reasoning intention of agent. In network security field, the research of attack intention recognition has just begun. At present, existing research is divided into two parts. A method builds attack scenario through correlating the alarms of intrusion detection systems, the other one enumerates all possible attack behavior by constructing attack graphs. The paper [3] correlated the alarms through modeling attack behavior. For a single attack behavior, the method refined the preconditions and consequences of the attack. Then the method correlated two attack behaviors according matching condition between prerequisite of subsequent behavior and consequences of previous acts. Thus, the method is no need to establish attack pattern base, and it can discover some unknown attack scenario with flexibility. The paper [4] automatically generated attack strategy described by attack strategy graph through alarm correlation. The method correlated the ultra-alarm generalized by the alarms with same essence. The flexibility and associate efficiency of method was increased, and the method simplified the analysis of attack strategy by measuring the similarity between attack strategies to discover the essence of the attack strategies. The paper [5] proposed a complex attack prediction method based on fuzzy hidden Markov model, identified attack scenarios membership of alarms by using stage transition matrix and fuzzy difference

ISSN: 1738-9976 IJSIA

method. These alarm correlation methods can find attackers' behavior characteristics and analyze attack strategy, but alarm correlation is a rebuilding process of attack behavior after attacks. Due to identifying intention after attacks, these methods can't provide support for advance guard.

The methods, analyze and correlate by exploiting vulnerabilities, can reasoning and predict intrusion by simulating attacks which security analysis in the form of attack graphs synthesizing the network topology, vulnerability information, firewall rules and other information. The paper [6] proposed application model to automatically generate attack graph. The paper [7] reduced the complexity of attack graph to easily understand using connection matrix clustering techniques. The paper [8] implements policy-based multi-host, multi-step analysis of the vulnerability. Model representation of network and simplifying the attack rules can greatly reduce the time to generate attack graph. The paper [9] analyzed network security protection using attack graph. However, most of these methods which are static analysis can't adaptively adjust the generation and display of attack graph based on real attacks and response measures. And the study of uncertainty that the probability of being exploited of attacks caused by different attack difficulty and hidden degree is not yet sufficient.

Therefore, we proposed a dynamic real-time network attack intention recognition method based on attack route graph. By correlating real-time network attacks and vulnerabilities, the method determines spread routes and stages of attack based on graph theory and probability theory, then dynamically reasoning possible intrusion intention and its probability according to attack behavior characteristics and network environment.

2. Real-Time Network Attack Intention Recognition Model

Attack intention recognition is a process of reasoning and judging intruders' intention. In practice, it is very difficult to identify the intention. Due to the complex of intruders' intention, it is difficult to establish a intention base to clear descript all intentions of attackers. And the uncertainty of complex attacks, which achieve an attack intention with different attacks combination and intrusion paths and exist randomness in attack process, increase difficulties to intention recognition.

Computer network is an open complex system. Achievement of intrusion intention not only depends on attackers themselves, but also relates specific network environment and protection measures. Therefore, the intrusion intention achieved by attackers is limited and predictable given a known network environment and protective measures. We proposed a dynamic attack intention recognition model based on network attack-defense confrontation (as shown Figure. 1).

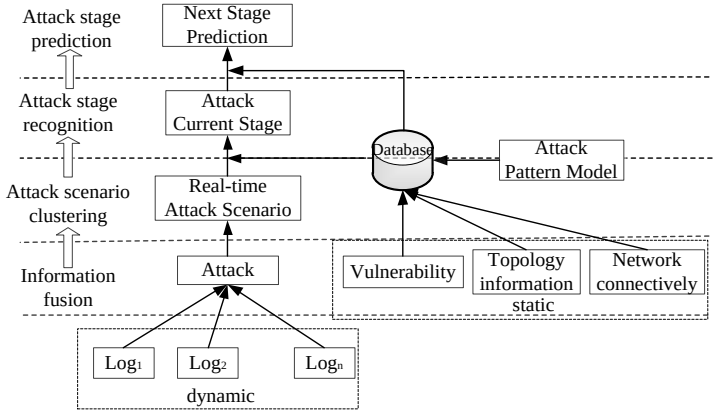


Figure 1. The Framework of Real-Time Network Attack Intention Recognition

According to the recognition model, we design the model of network security is as follows:

2.1 Alarm Information

Alarm information *Log* includes the alarm logs from intrusion detection systems, firewalls, system logs and other sensors which use a six-tuple $(id, time, type, content, id_s, id_d)$ to represent. Where *id* is a unique identifier of alarm information, *time* is the generation time of alarm, *type* is the type of alarm, *content* is content of the alarm, id_s is the node generating the alarm, id_d is the detection node of the alarm.

2.2 Security Event

Security event alert is the advanced alarms after information fusion, which uses a seven-tuple *(id, time, Sip, Dip, Sport, Dport, AttackType)* to represent. Where *id* is a unique identifier of the event, *time* is the occurrence time of the event, *Sip* is the attacker's source address, *Dip* is the attack target address, *Sport* is the attacker's source port, *Dport* is the attack destination port, *AttackType* is the attack type used by the attack.

2.3 Network Connectivity

Network connectivity represents communication relationship between hosts. To protect the important assets of network, managers will set up firewall access policy to prevent the external hosts access the internal network or only allow communication through specific ports. We use a triple to describe the network Connectivity $(host_i, host_j, protocol/port)$, where $host_i, host_j$ represent connected host, *protocol/port* represents the communication protocol and port of hosts.

2.4 Vulnerability Exploiting Relationship

Vulnerability exploiting relationship represents dependency attacks with vulnerabilities, which is successful invasion probability of an attack exploiting the specific vulnerability. We use a triple to describe the vulnerability exploiting relationship $(attack_i, vuls_j, p_{ij}(e))$, where $attack_i$ represents types of attacks, $vuls_j$ represents types of vulnerabilities, $p_{ij}(e)$ represents the successful invasion probability when $attack_i$ exploits $vuls_j$.

Calculating the dependency attacks with vulnerabilities is an important research direction in the field of information security, which requires analysis of a large number of attack data, and combined with prior knowledge of network security experts. We calculate $p_{ij}(e)$ by referencing CVSS [10] (Common Vulnerability Scoring System). VS (Vulnerability Scores) can be got based on CVSS, which ranges from 0 to 10. When VS = 0, it indicates that the attack could not exploit the vulnerability; when VS = 10, it indicates invasion will be successful. In this paper, a simple conversion defines as $p_{ij}(e) = VS/10$.

2.5 Attack Intention Stage Transition Model

We use (s_i, s_j) to describe attack intent stage transition, $s_i, s_j \in S$, where S represents the stage sets of attacks, s_i, s_j represent previous and post stage, $s_i = pre(s_j)$, $s_j = post(s_i)$. To complete a network attack, attackers need carry out multiple attacks which become attack pattern according with causality.

We use a triple $s = (A(s), E(s), s)$ to describe each attack stage. Where $A(s) = \{a_1, a_2 \cdots a_n\}$ represents the types of attacks needed to complete the stage; $E(s) = \{e_1, e_2 \cdots e_m\}$ represents the dependency between attacks, which uniquely determined by the ordered pair of attacks $(a_i, a_j) = e_{i \times n + j}$. If successful invasion of an attack intention need all of the two attacks must be successful, the dependency of two attacks is parallel relationship, *e*=1; if successful invasion of an attack intention just need any one of two attacks, the dependence of two attacks is select relationship, *e*=0; *s* represents the stage of attack intention.

The attack pattern base can be generated from known network attack mode based on the model. Figure. 2 is an instance of a state transition diagram of an attacks intention.

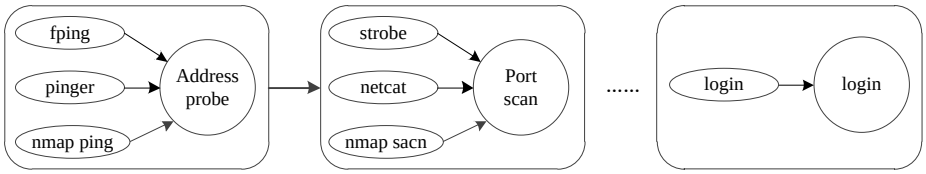


Figure 2. Attack Intention Stage Transition Diagram

3. Real-Time Attack Intention Identification and Prediction

3.1 Information Fusion

Currently a large number of security sensors are deployed in the network, alarm information from intrusion detection systems, firewalls, virus detection systems and other sensing devices reflect the different levels of security status. However, the data format is chaotic, and there is a lot of redundancy and false positives, which cannot be applied directly to the network security analysis. Currently, there are some algorithms [11-13] to fuse alarm information. Use the proposed algorithm in [11] for data to fuse alarm information and get more accurate attack probability *p(a)*, so that the alarm information from different sensors complementary and mutually confirmed, more accurate information about attack can be obtained.

Attack success depends on attack techniques and its environment configuration and vulnerability information of the invasion network, just when the environment configuration and vulnerability information of intrusion network can be exploited by this attack, it can be successful invasion. So successful invasion probability *p(ac)* will be calculating based on the exploit relationship of vulnerabilities $(attack_i, vuls_j, p_{ij}(e))$.

$$p(ac) = \begin{cases} p(a)\, p_{ij}(e), & vlus_j \in Vlus \\ 0, & otherwise \end{cases} \quad (1)$$

Where *p(a)* is the invasion probability, $vlus_j$ is the vulnerabilities exploiting by the attack, $p_{ij}(e)$ is the probability of successful invasion when the attack exploiting the depending vulnerability, $vlus_j \in Vlus$ means invasion host exists vulnerabilities depended by the attack.

3.2 Attack Scenario Clustering

Due to network may be attacked by many intruders in a same period of time, it needs cluster the security event into different attack scenarios to identify intrusion intention of each attackers. We divide each new received alert into attack scenario based on a quantitative alarm correlation method.

In a same multi-step attack, since the intrusion intention and target is very clear, the property of security event caused by attack steps exist some correlation. For example, if

an attacker wants to scan vulnerability of network, the attacker must first IPSweep scanning. The source IP addresses of alarm caused by two attacks are same, and their occurrence time exists context-related. Therefore, the attack correlation degree is defined mainly decided by association of property.

Definition. 1. Attack Correlation Degree. The Attack Correlation Degree, *cor(a,b)*, represents the possibility of the two attacks, *a* and *b*, belonging to the same attack scenario. In this paper, we regard several attributes of the attack, such as IP address, port, timestamp, attack style and *etc*., as the basis to calculate the attack correlation degree. The Attack Correlation Degree Function is defined as follow:

$$cor(a,b) = \sum_{k=1}^{n} \alpha_k Feature_k(a,b) \Big/ \sum_{k=1}^{n} \alpha_k \quad (2)$$

Where $Feature_k(a,b)$ represents the correlation degree among the attribute k^{th} and α_k represents the weight. The two parameters can be selected according to Reference [14].

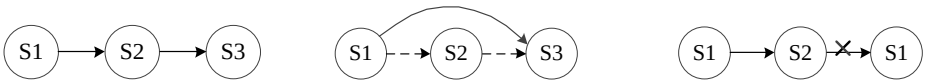
When the system receives a new security event, we calculate attack correlation degree between the alert with every saved attack scenarios. If there are many correlation degree exceed the pre-set threshold, then put the alert into the attack scenario with largest correlation degree. If all correlation degree doesn't exceed the pre-set threshold, consider the security incident as a new attack scenario.

3.3 Real-Time Attack Stage Recognition Algorithm

Definition. 2. State Function. This function, *bool(s)*, is used to identify whether the attack stage is occurred. If the attack stage has occurred, *bool(s)* equals to true; otherwise, *bool(s)* equals to false.

Definition. 3. Transference Waiting Window. The intrusion of a network attack always carries in a period. If the next attack stage doesn't occur after a long time, the intrusion will fail due to the attacker's ability can't exploit the network vulnerabilities. Thus a Transfer Waiting Window is necessary to judge whether the attack is success. Mostly, an attack period is 2h, so we set the Transference Waiting Window $\partial\tau = 2h$.

The method clustered real-time alerts into different attack scenes based on attack correlation degree, and associated the alerts with attack pattern base. We summarize 3 kinds of typical scene (shown as Figure. 3).



(a) Normal transferring (b) Skipped transferring (c) Repeated transferring

Figure 3. State Transferring Scenes

Figure. 3(a) shows the normal transferring, which the previous state and the post state of an attack intention can occur sequentially. In Figure. 3(b), the state transfers from S1 to S3 without discovering S2, which we call it as skipped transferring. Actually it is the most common scene in a real network due to the high false negative rate existing in the intrusion detection equipment caused by the difference between detection strategies and characters of intrusion. As shown in Figure. 3(c), the repeated transferring represents the scene that some state occurs repeatedly, like S1. This is always because the alert data is delayed while transmission or the clocks in different security sensors are asynchronous. Based on the above analysis, the Real-Time Attack Stage Recognition Algorithm can be described in the following.

Algorithm. 1. Real-Time Attack Stage Recognition Algorithm
Input: fusion security events
Output: the intention of attacker; the current attack phase

Begin

Step 1: Calculate the attack correlation degree, *cor(a,b)*, between the real-time alerts and the current state of the generated attack scene, s_{current}. Then cluster the alerts into different attack scenes according to the attack correlation degree.

Step 2: By analyzing the correlation between alerts in each attack scene and the generated attack pattern base $(A(s), E(s), s)$, search for the attack stage *s*, and record the current time *t*.

Step 3: If the previous state of *s* is the current state in the attack scene, which can be described by the equation $s_{current} = pre(s)$, and then we regard this type of scene as normal transferring. Add *s* into the attack scene, set the parameter *bool(s)* =true, update the current state $s_{current} = s$ and the state occurring time $t_{current} = t$. Then turn to Step7.

Step 4: If the function *bool(s)* =true, the scene belongs to repeated transferring. We discard this attack state and turn to Step7.

Step 5: If the previous state of *s* is not the current state in the attack scene, which state function *bool(pre(s))* =false. Search the attack pattern base, if *s* is several steps latter than the current state, then we regard this type of scene as repeated transferring. Under this situation, we add *s* as well as the states between *s* and s_{current} into the attack scene, set the function *bool(s)* =true and update the current state, $s_{current} = s$, as well as the state occurring time, $t_{current} = t$. Then turn to Step7. If *s* isn't latter than the current state, then turn to Step6.

Step 6: If s isn't latter than the current state, this situation dedicates this attack pattern haven't been occurred. We label this attack path as a new attack path and add it into the attack pattern base. Set *bool(s)* =true, update the current state, $s_{current} = s$, as well as the state occurring time, $t_{current} = t$. Then turn to Step1.

Step 7: Associated the attack scene and the attack pattern base, we can get the attack intention set {G1, G2,⋯, Gn}. The attack technology and stage of attack patterns can possibly be the same, so we can recognize several attack intentions according to the generated attack scene (as shown in Figure. 4.). After updating the current state, we judge whether s is on the path of a recognized attack intention. If yes, then $s \in path_i$ and keep this attack intention. Otherwise we have to delete the attack intention.

Step 8: Check the time of state transferring of all attack scenes, if $t - t_{current} > \partial\tau$, then delete the attack scene and turn to Step1.

End

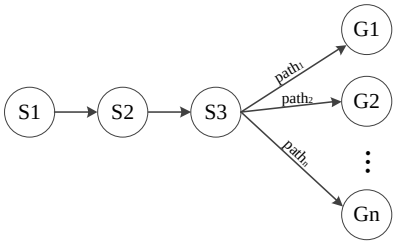


Figure 4. Attack Intention Reorganization

Calculating attack phase achievement probability *p(s)* utilizes real-time detecting attack behavior *Alert* and dependency relationship of attacks $(A(s), E(s), s)$, based on establishing attack pattern base.

$$p(s) = \begin{cases} p_i(ac) + p_j(ac) - p_i(ac)p_j(ac), & e_{i\times n+j} = 0 \\ p_i(ac)p_j(ac), & e_{i\times n+j} = 1 \end{cases} \quad (3)$$

Where $p_i(ac)$ and $p_j(ac)$ are the attack success probability of $Alter_i$ and $Alter_j$, $e_{i\times n+j} = 0$ means it is possible to achieve the intention of the attack when all of attack have succeed, $e_{i\times n+j} = 1$ means it is possible to achieve the intention of the attack when any one of the attacks has succeed.

3.4 Real-Time Attack Stage Prediction Algorithm

Definition. 4. Minimum Vulnerabilities Set Needed in Attack Stage. In this paper we describe this set as $\{MinVuls_i \mid MinVuls_i = vuls_j \,\&\, vuls_k \ldots \,\&\, vuls_l\}$ .This set represents the essential vulnerabilities which attackers exploit to achieve a certain attack phase. The attacker perhaps utilize various attack method to achieve the aim, thus this set contains more than one element.

Definition. 5. Accessible Host, which means the set of hosts which can be used to carry out the post intrusion. This set includes the host which exist the current attack state as well as the hosts that connect with it.

Definition. 6. Exploitable Vulnerabilities Set. This concept means the vulnerabilities of accessible hosts that can be exploited by the attacker. In this paper we describe this set as $\{ExploitVuls_i\}$, which isn't equal to all vulnerabilities existed in the accessible hosts because of the limitation of protocol and port. The host which existing the current attack state, can exploit all vulnerabilities in it, which means $\{ExploitVuls_i\} = Vuls$; while other hosts can generate the exploitable vulnerabilities set according to the connectively relationship $(host_i, host_j, protocol/port)$.

Definition. 7. Exploitable Ratio of Vulnerability. This parameter, *Av*, describes the situation whether the exploitable vulnerabilities can satisfy the requirement of the following intrusion. If yes, which means $\{MinVuls_i\} \subset \{ExploitVuls_i\}$, every vulnerabilities in this host can be used by the attacker. Detailed description is shown in the following:

$$Av = \begin{cases} 1, \exists MinVuls_i \in \{MinVuls\}, MinVuls_i \subset \{ExploitVuls\} \\ 0, \quad otherwise \end{cases} \quad (4)$$

Because the ability of attacker is unknown, we suppose that the attacker can utilize all kinds of attack technologies; in the mean time we suppose that the attacker choose the next target with equal probability. The detailed process of the Real-Time Attack Stage Prediction Algorithm is described as the following.

Algorithm. 2. Real-Time Attack Stage Prediction Algorithm

Input: the current attack path $path_i$; the recognized attack intention set {G1,G2,⋯,Gn}

Output: the post attack stage; the post intrusion hosts.

Begin

Step 1: Waiting for the attack path that is updating. If the attack path turns into the current attack path s_{current} , turn to Step 2. Otherwise repeat this step.

Step 2: Search for the accessible hosts according to the attack-existed host and the connected relationship.

Step 3: On the basis of the connected relationship $(host_i, host_j, protocol/port)$, generate the exploitable vulnerabilities sets of all accessible hosts $\{ExploitVuls_i\}$.

Step 4: In accordance with the attack pattern base, search for the minimum vulnerabilities set needed in the ost intrusion $\{MinVuls_i\}$.

Step 5: Calculate every utilization ratio of vulnerability in each accessible host. If $Av_{ij} = 1$, $host_i$ can become the following goal in the intrusion; otherwise $host_i$ is inaccessible in the following attack phase.

End

For each identified attack intention G_j of attack path $path_i$, the invasion hosts $host_k$ of next attack phase will be got with real-time attack phase prediction algorithm. Under ability of attackers is unknown, we assume that the attackers have a strong ability to implement all the existing methods to complete their intentions. Given the each group required minimum vulnerability of attack phase $MinVuls_h = vuls_x \& vuls_y \ldots \& vuls_z$ in available vulnerabilities set, we have

$$p_h(ns) = \prod^{H} p_x(e) \tag{5}$$

Where *H* is number of vulnerability in the minimum vulnerability group, $p_x(e)$ is invasion access probability of vulnerabilities to the corresponding attack.

Given all of the available vulnerabilities set, we have

$$p(ns) = 1 - \prod (1 - p_h(ns)) \tag{6}$$

4. Experimental Analysis

In order to verify the feasibility and effectiveness of the model and algorithm, we build an experimental network, the network topology as shown in Figure. 5. The network consists of a firewall, a Web server, a Mall server, a SQL server, two intrusion detection systems, and an attack host. The firewall policy parts network into two subnets. Web server, Mall server and IDS1 distributed in DMZ Zone, SQL server and IDS2 distributed in Trusted Zone.

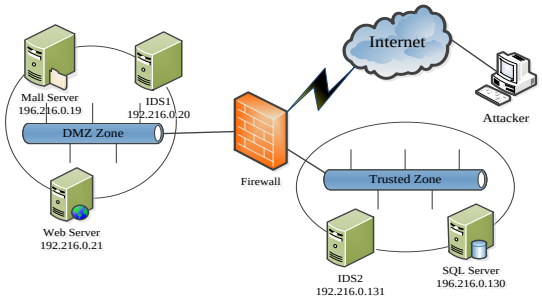


Figure 5. The Network Topology

The firewall has a strong set of policies (shown in table I) to prevent remote access to SQL server. In particular, all machines in DMZ Zone passively receive service requests and only respond to the sender as needed. In order to accommodate Web service's transactions, the web server is allowed to send SQL queries to the SQL server. The firewall policy is network connectively relationship too.

Table 1. List of Firewall Policy

From host	To host	Protocol/port
Remote machine	Mail server	IMAP(143)&SMTP(25)
	Web server	HTTP(80)
Web server	SQL Server	SQL(1433)

Through the network vulnerability scanning, the vulnerabilities of hosts are given in Table II.

Table 2. List of Vulnerabilities in Network

Host	Vulnerability	CVE#
SQL Server	SQL Injection (V1)	CVE 2008-5416
Mail server	Remote code execution in SMTP (V2) Error message information leakage (V3) Squid port scan vulnerability (V4)	CVE 2004-0840 CVE 2008-3060 CVE 2001-1030
Web server	IIS vulnerability in WebDAV server (V5)	CVE 2009-1535

To steal the data from SQL server, attacker must get the root privilege of SQL server. The attacker first searched for valid hosts of the network through IPsweep address scanning, and attacker found Web server and Mall server. Then attacker scanned the ports of valid hosts, and discovered Web server's communication port is 80. The attacker obtained user permissions of Web server exploiting CVE-2009-1535, then obtained the root privilege of SQL server exploiting CVE-2008-5416 through SQL protocol. We collected the alarm data of IDS and firewall, as well as the security audit logs of hosts. Then the attack stage transition diagram (shown in Figure. 6.) can be got with the attack intention recognition algorithm proposed in Section 3.

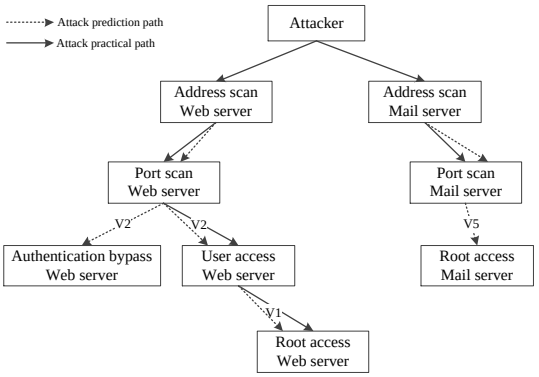


Figure 6. Attack Stage Transition Diagram

Then, calculating the stage occurrence and stage transition situation of network attack (shown in Table III) is according to the identified attack stage transition diagrams, combined with intrusion detection alarm data and audit logs. Where *host* and *p(s)* in stage occurrence situation (*host, p(s)*) mean the attacked host of stage and attack stage achievement probability; *host*, *post(s)* and *p(ns)* in stage transition situation (*host, post(s), p(ns)*) mean the attacked host of next stage, next attack stage and attack stage transition probability.

Table 3. List of Attack Stage Situation

Attack stage	Stage occurrence situation	Stage transition situation
Address scan	(Mail, 0.98) (Web, 0.98)	(Mail, port scan, 1) (Web, port scan, 1)
Port scan	(Mail, 0.96) (Web, 0.95)	(Mail, root access, 0.98) (Web, user access, 0.96)

		(Web, Authentication bypass, 0.96)
User access privilege	(Web, 0.93)	(SQL, root access, 0.94)
Root access privilege	(SQL, 0.93)	

From Figure. 6 and table. 4 can be seen that the proposed method can accurately predict as well as constantly update attack intention and targets in advance. Then we analyze the performance of proposed algorithm.

1) Time complexity. Algorithm. 1. scanned every security events once to match the pattern in the attack pattern base. The number of attack patterns, *m*, as well as the number of stages in every pattern is steady. The algorithm starts only when a new attack is occurred, so the time complexity of Algorithm. 1. is *O(mn)*. Algorithm. 2. searched for the possible attack intentions in all accessible hosts, while the number of accessible hosts and the number of possible attack intentions is steady. The time complexity of Algorithm. 2 is *O(kl)*.
2) Storage size. AG Algorithm, proposed in paper [6], requires listing all states in the network, which the time complexity is $O(2^n)$. In paper [15], EDG Algorithm improves the scale by simplifying twice to eliminate the loops. TSTG Algorithm, proposed in paper [16], avoids the above problems by lifting the authority and associated analyzing the attack. But it is hard to deal with the large network.

In this paper, the proposed stage recognition algorithm based on the attack intention divides attacks into different scenes rather than enumerate all states. We also avoid the redundancy of attack state and attack link by recognizing every attack intention. The scale of the proposed algorithm is polynomial. And it improves the comprehension of the attack scene by the recognition of attack intention. In the meantime, it avoids the appearance of loop through discussing the possible state transferring scenes to weaken the influence of false positive and repeated alert. TABLE IV compares the above algorithms in the network with 3 hosts and 5 vulnerabilities.

Table 4. Comparison of Algorithms

Algorithm	AG	EDG	TSTG	This paper
nodes	25	15	11	8
edges	43	26	16	7
granularity	exponential	polynomial	polynomial	polynomial
loop	no	yes	no	no

5. Conclusions

Based on the basis of artificial intelligence intention recognition, we proposed a dynamic real-time network attack intention recognition algorithm. By correlating real-time security alerts and vulnerabilities, we found the spread route and stage of attacks based on graph theory and probability theory. Then we identified the attack intention and predicted the possible transition of attacks, combined with network connectivity relationship. A simulation experiments for the proposed network attack intention recognition algorithm is performed by network examples. The experimental results show that the proposed method can be more accurately identify attack intention and fully predict the post stage of attacks. Due to attackers always use deception, concealment or other means to conceal their behavior and intention, there is different between attack-defense intention recognition with intention recognition in artificial intelligence field. So the intrusion intention recognition needs further research.

References

[1] K. A. Tahboub, "Intelligent human-machine interaction based on dynamic Bayesian networks probabilistic intention recognition", Journal of Intelligent and Robotic Systems. no.45, **(2006)**, pp. 31-52 .
[2] X. Chai, Q. Yang, "Multiple-goal recognition form low-level signals. Proceedings of the 20th National Conference on Artificial Intelligence", Pittsburgh, Pennsylvania, **(2005)**, pp. 9-13.
[3] F. Cuppens, F. Autrel, A. Miege, S. Benferhat, "Recognizing malicious intention in an intrusion detection process". Proceedings of the 2nd International Conference on Hybrid Intelligent Systems, **(2002)**; Santiago, Chile.
[4] P. Ning, D. Xu, "Learning attack strategies from intrusion alerts", Proceedings of the 10th ACM Conference on Computer and Communications Security, **(2003)**; Washington, DC, USA.
[5] Z. Yanxue, Z. Dongmei, L. Jinxing. "Approach to forecasting multi-stage attack based on fuzzy hidden markov model". Electronics Optics & Control, **(2015)**, no. 22, pp. 39-44.
[6] O. Sheyner, J. Haines, S. Jha, R Lippmann, "Automated Generation and Analysis of Attack Graphs". Proceedings of the 2002 IEEE Symp on Securtiy and Privacy, **(2002)** May 12-15; Berkeley, California, USA
[7] S .Noel, S. Jajodia, "Understanding complex network attack graphs through clustered adjacency matrices. Proceedings of the 21st Annual Computer Security Applications Conference", **(2005)** December 5-9; Tucson, AZ, USA.
[8] X. Ou, S. Govindavajhala A. W., Apple. "MulVAL: A logic-based network security analyzer". Proceedings of the 14th Usenix security Symp. **(2005)** August 1–5; Baltimore, MD.
[9] M. Alhomidi, M Reed." Attack graph-based risk assessment and optimization approach". International Journal of Network Security & Applications., no. 6., **(2014)**, pp.. 31-43.
[10] M. Schiffman, "Common Vulnerability Scoring System (CVSS)," http://www.first.org/cvss/cvss-guide. html **(2011)**.
[11] W. Yong, L. Yifeng, F. Dengguo. "A Network Security Situational Awareness Model Based on Information Fusion". Journal of Computer Research and Development. no. 46, **(2009)**, pp. 353-362.
[12] Q. Peili, Y. Yang, "Study on Application of Honeypot in Network Security". Journal of Harbin University of Science and Technology. 14, 37-41 **(2009)**
[13] Q. Peili, S. Ping, "Research and Implementation of Intrusion Detection System Merged Scanner Technique". Journal of Harbin University of Science and Technology, no. 14, **(2009)**, pp. 5-59.
[14] F. Kavousi, B. Akbari, "Automatic learning of attack behavior patterns using Bayesian networks". 6th International Symposium on Telecommunications, **(2012)** November 6-8; Tehran, IRAN.
[15] S. Noel, Jajodia, O'Berry B, S Jacobs M. "Efficient minimum-cost network hardening via exploit dependency graphs". Proc of the 19th Annual Computer Security Applications Conference, CA: IEEE Computer Society, **(2003)** December 8-12; Las Vegas, Nevada, USA.
[16] Lv Huiying, P. Wu, W. Ruimei, W. Jie. "A Real-time Network Threat Recognition and Assessment Method Based on Association Analysis of Time and Space. Journal of Computer Research and Development. 51, **(2014)**, pp. 1039-1049.

