Postprint: Research on Attack Path Prediction Method Based on T_{NAG} Model

Authors: Zhai Haixia, Lu Yuemeng, Wang Hui, Ao Shan

Date: 2020-09-28T00:00:00+00:00

Abstract

To better characterize attacker attack trajectories, we design a path prediction method based on the T_{NAG} (time attribute network attack graph) model. First, we propose a new attack graph model T_{NAG}, which distinguishes attacker capabilities based on real-time behavior trajectories. Second, based on the characteristic that attackers possess different capabilities, we introduce the concept of attack intention, comprehensively consider operational risk and attack gain, incorporate a time decay parameter into the attack intention calculation, and design a quantification method for vulnerability exploitability based on attacker capability. Finally, we further integrate attack intention and vulnerability exploitability into the consideration of path reachability probability, and present the IntenAbi-PathPre algorithm for predicting attack paths. Experimental results demonstrate that the proposed method can effectively eliminate redundancy in the attack graph and significantly improve the accuracy of attack path prediction.

Full Text

Research on Attack Path Prediction Method Based on T_{NAG} Model

Vol. 38 No. 3
Application Research of Computers

Zhai Haixia, Lu Yuemeng†, Wang Hui, Ao Shan
(College of Computer Science & Technology, Henan Polytechnic University, Jiaozuo 454003, Henan, China)

Abstract

To better characterize an attacker' s trajectory, this paper proposes a path prediction method based on the T_{NAG} (time attribute network attack graph)

model. First, we introduce a novel attack graph model T_{NAG} that distinguishes attacker capabilities according to real-time behavioral traces. Second, based on the characteristic that attackers possess different capabilities, we propose the concept of attack intention, which comprehensively considers operational risk and attack profit while introducing a time decay parameter into the intention calculation. We also design a vulnerability utilization quantification method based on attacker capabilities. Finally, we integrate attack intention and vulnerability utilization into the consideration of path reachability probability and present the IntenAbi-PathPre algorithm for predicting attack paths. Experimental results demonstrate that this method can effectively eliminate redundancies in attack graphs and significantly improve the accuracy of attack path prediction.

Keywords: attack graph; path prediction; attacker' s intention; path accessibility

0 Introduction

The widespread application of the Internet has made social production and human life more efficient and convenient. However, numerous vulnerabilities in complex networks inevitably lead to various cybersecurity threats. According to the "2018 Summary of China' s Internet Network Security Situation" released by the National Internet Emergency Center (CNCERT), malicious programs such as computer viruses, worms, and Trojans were transmitted more than 5 million times per day on average in 2018 [**?**]. Since 2014, the number of security vulnerabilities collected by the CNVD vulnerability platform has increased annually, with an average growth rate of up to 15%. The latest cybersecurity report published by Positive Technologies in the United States shows that 765 million users worldwide suffered varying degrees of cyberattacks in 2018. According to a recent survey report released by Radware in Israel, 34% of respondents considered application vulnerabilities to be the primary factor seriously affecting network security in 2019 [**?**].

For enterprise networks, it is necessary to anticipate existing risks when configuring protection measures, thereby enabling targeted protection of critical areas and nodes. Attack graph technology [**?**] can visually display potential attack paths that attackers may adopt. By constructing attack graph models to analyze network security, network administrators can not only identify important nodes requiring enhanced protection [**?**, **?**] but also predict subsequent attack targets when attacks occur [**?**, **?**].

In recent years, numerous scholars have not only used attack graph models as tools for analyzing network security but have also proposed many methods for evaluating network security and predicting attack intentions from different perspectives. Ye Ziwei et al. [**?**] analyzed the basic elements of attack graph models, provided definitions and brief introductions of attack graphs, and summarized the current status and existing problems in the application of attack

graph models. Qin Hu et al. [?] argued that changes in attacker capabilities should be marked by privilege escalation and proposed an attack graph generation method based on an attacker capability escalation model. Mohammad et al. [?] pointed out the limitations of using static vulnerability probabilities alone for risk prediction, extracted and analyzed real-time intrusion alerts and dependency graph information during prediction, and proposed a network attack prediction method based on information integration. From these research results, we can conclude that, on the one hand, differences in attackers' levels of vulnerability exploitation and risk avoidance during attacks have certain impacts on network security, and the degree of impact can be understood through changes in objective factors during the attack process. On the other hand, the large amount of information generated by attack behaviors can assist in predicting attack behaviors, but only after analyzing the influence relationships and trends of various factors on path reachability and network security and representing them with appropriate mathematical models.

Literature [?] proposed an attack path generation algorithm based on attack graph models. Assuming the goal is to obtain K paths with the highest reachability probability, the algorithm dynamically adjusts the probability distribution according to the K value and achieves path prediction by calculating the cumulative reachability probability of attack paths. However, when quantifying node reachability probability, this algorithm only considers the factor of vulnerability utilization rate, lacking modeling and analysis of the attack events themselves.

Literature [?] used Bayesian attack graphs to assess dynamic network risks, fully considering the dynamic impact of attack events on node posterior probabilities and employing Bayesian inference methods to predict attack paths. However, the success probability of attack events relies on expert experience, ignoring the fact that the probability of resource nodes being occupied after an attack event is related to the specific context of the attack behavior, which affects prediction accuracy.

Literature [?] mapped attack graphs to Markov chains and proposed a state transition probability algorithm for nodes in the Markov chain. However, the calculation of state transition probabilities depends on the static factor of vulnerability availability metrics in CVSS, without considering the dynamic impact of attacker capabilities on vulnerability availability, resulting in low accuracy of attack path prediction.

Literature [?] assumed that attackers are rational, qualitatively analyzed factors affecting attack cost and profit, provided quantification algorithms for attack cost and profit respectively, and introduced a cost-benefit model into node reachability calculation to achieve attack path prediction. However, the impact of attack time consumption was not considered when quantifying node reachability.

Literature [?] proposed a network risk assessment method based on risk flow attack graphs, which not only describe network structure but also model attack

scenarios. The paper focused on analyzing the relationship between intrusion events and network risk, then proposed a risk assessment scheme based on fuzzy evaluation. However, the large amount of information extracted from intrusion events was only used to quantify network risk, without analyzing other factors affecting network security such as attack behavior profit and time consumption, or the skill differences demonstrated by attackers.

Literature [**?**] designed an attack path optimization scheme based on time gain compensation rate, integrating the time characteristics reflected by attack behaviors into the attack graph model, considering attack time consumption as a time cost factor in attack cost, focusing on analyzing the correlation between attack time consumption and node reachability probability, and eliminating path redundancy based on quantification of time cost growth. However, the quantification of time cost in the paper mainly depends on the accumulation of time spent by meta-operation sequences, leading to an uncontrollable value range of time cost and affecting the accuracy of risk analysis.

Addressing the limitations of existing research in comprehensively considering factors affecting network security and low prediction accuracy, this paper proposes the T_{NAG} model based on attacker intention, analyzes and provides quantification methods for factors influencing attacker intention one by one, improves the vulnerability utilization calculation method according to attacker capability levels, and finally optimizes the attack graph by combining attacker intention with vulnerability utilization and calculates path reachability probability. By comparing path reachability probability values, we can predict the attack paths that attackers may adopt.

1 T_{NAG} Attack Graph Model

To analyze attacker behavior and describe the attack graph model more accurately and clearly, we first introduce some variable definitions.

Definition 1 (Network Security Vulnerability). Represented by a binary tuple $(vID, vInfo)$, where vID is the unique identifier of the vulnerability and $vInfo$ describes the vulnerability.

Definition 2 (Network Device). Including hosts and firewalls in the network, represented by $(hID, Services, V, hValue)$. Here, hID represents the device number; $Services$ represents the set of services running on the device; V represents the vulnerability information set of the device, described by Definition 1; $hValue$ represents the asset value of the device. The set of all devices in the network is denoted by H.

Definition 3 (Attack Intention). Represents the quantification of the attacker' s subjective tendency to launch attacks, denoted as $\kappa \in (0, 1)$. A larger value indicates stronger attack intention.

Definition 4 (Vulnerability Exploitability). Represents the difficulty for an attacker to launch an atomic attack using a specific vulnerability, denoted as

$vul() \in (0,1)$. A larger $vul()$ value indicates stronger vulnerability exploitability and lower difficulty in launching atomic attacks.

Based on existing research, attack graph models are mainly divided into state attack graphs and attribute attack graphs. Among them, attribute attack graphs [**?**] use elements affecting network security as independent vertices, effectively controlling the scale of generated attack graphs. Compared with state attack graphs, they are more suitable for large-scale networks. To reflect the impact of time parameters on attack intention and the influence of attack intention on path prediction, this paper proposes the T_{NAG} (Time attribute Network Attack Graph) model based on attribute attack graphs to evaluate network security.

Definition 5 (Attack Graph).
a) R is the set of resource nodes, denoted as $R = R_0 \cup R_{mid} \cup R_g$. Here, R_0 is the set of initial resource nodes, i.e., the set of resource nodes occupied by the attacker at the initial moment; R_{mid} is the set of intermediate resource nodes, i.e., the set of intermediate resource nodes that the attacker needs to occupy from the initial resource nodes to the target resource nodes; R_g is the set of target resource nodes, i.e., the set of resource nodes that the attacker ultimately expects to occupy. An independent resource node is denoted as $r \in R$, $r = (rID, hID, j)$, where rID is the unique identifier of the resource node, hID is the identifier of the network device to which the resource belongs, and j represents the specific content of the resource, such as a certain level of privilege or trust relationship.

b) A is the set of attack nodes, denoted as $A = \{a_1, a_2, ..., a_n\}$. For $a_i \in A$, it is denoted as $a_i = (aID, vID, aSTime, aETime)$. Here, aID represents the attack name, vID represents the vulnerability exploited by the attack, $aSTime$ represents the start time of the attack, and $aETime$ represents the end time of the attack.

c) E is the set of directed edges in the attack graph, denoted as $E = E_1 \cup E_2$. $E_1 = \{e_{ij} | e_{ij} = \langle r_i, a_j \rangle, r_i \in R_0 \cup R_{mid}\}$ indicates that the attacker can only launch attack a_j after occupying the predecessor resource node r_i; $E_2 = \{e_{ji} | e_{ji} = \langle a_j, r_i \rangle, r_i \in R_{mid} \cup R_g\}$ indicates that the attacker launches attack a_j to occupy the successor resource node r_i.

d) $\theta_{ij} \in \{\text{true}, \text{false}\}$. If $\theta_{ij} = \text{true}$, the attacker may launch an attack on resource node r_j after obtaining resource r_i; if $\theta_{ij} = \text{false}$, the attacker abandons the attack on resource node r_j.

e) $\delta \in \{\text{and}, \text{or}\}$ represents the logical relationship between a node and its predecessor nodes. If $\delta(a_i) = \text{and}$, the prerequisite for attack a_i to occur is that the attacker occupies all predecessor resource nodes; if $\delta(a_i) = \text{or}$, the attacker only needs to occupy any predecessor resource node for attack a_i to occur. If $\delta(r_j) = \text{and}$, resource node r_j can only be occupied after all predecessor node attacks occur; if $\delta(r_j) = \text{or}$, resource node r_j may be occupied after any predecessor node attack occurs.

f) Attack weight set W. For $e \in E_2$, there is a corresponding weight $w \in W$, where $w = (vul, \kappa)$, vul represents vulnerability exploitability, and κ represents the attacker' s intention.

Definition 6 (Attacker Capability Level). Represents a unified quantification of different attackers'capabilities, denoted as $Attabi_{low}$, $Attabi_{mid}$, $Attabi_{high}$.

Definition 7 (Attack Path). Composed of a binary tuple, denoted as $PATH(Seq, Attabi)$. Here, Seq represents the path sequence, and $Attabi$ represents the attacker capability level. For any target node $r_g \in R_g$, starting from the initial node r_0, there exists an ordered node sequence $r_0, a_1, r_1, a_2, ..., r_m, a_n, r_g$. If the edges between any two adjacent nodes in the sequence satisfy $\langle r_{i-1}, a_i \rangle \in E_1$ and $\langle a_i, r_i \rangle \in E_2$, then the node sequence is called a path sequence, denoted as $Seq = \langle r_0, a_1, r_1, a_2, ..., r_m, a_n, r_g \rangle$.

Definition 8 (Attack Pattern). Represents the general manner in which attackers implement attacks. The prerequisite for an attack behavior is dependence on specific resources, such as accessing a service on the attacked host or obtaining the trust of the attacked host. The purpose of the attack is to further obtain more resources. As shown in [Figure 1: see original paper], in continuous attacks, the resources obtained from the previous attack are the prerequisite resources for the next attack. A series of attack behaviors by the attacker are organically connected through resource nodes, forming specific attack intentions.

Definition 9 (Attack Sub-path). Based on the analysis of attack patterns, since the completion of atomic attack a_i always depends on prerequisite resource r_i and the purpose of the attack is successor resource r_j, we consider $r_i \rightarrow a_i \rightarrow r_j$ as the smallest unit and use it as a whole consideration for the path.

2.1 Attacker Capability Level

Considering that in practical applications, attackers with different capabilities have varying degrees of exploitation for the same vulnerability, thereby affecting attack results, the probability of successful attack is related not only to complexity but also to attacker capability. This paper distinguishes attacker capabilities based on the difficulty of exploiting vulnerabilities required for attacks. This approach avoids the traditional practice of determining attacker capabilities through subjective experience and improves the credibility of results.

Literature [**?**] proposed and proved that the probability distribution of attacker capability levels inferred from the same attack behavior result is proportional to the probability distribution of the attacker successfully or unsuccessfully implementing the attack. CVSS [**?**] uses the attack complexity parameter Ac to divide the difficulty of vulnerability exploitation into Low, $Medium$, and $High$ levels. The higher the probability of successful attack, the higher the value of attack complexity Ac. Therefore, we define the attacker' s capability level as equal to the highest difficulty level of vulnerabilities in the set exploited during the attack process. Assuming the set of vulnerabilities exploited by a series of

continuous attacks launched by the attacker is V, the calculation formula for attacker capability level is as follows:

$$Attabi = \max_{v \in V}(Ac(v)) \quad (1)$$

2.2 Attacker Intention

Typically, an attacker' s choice of attack path is influenced not only by operational risk and attack profit but also by dynamic factors such as attacker capability and attack time consumption. Attackers implementing different attack behaviors may have different capability levels, which affect the value of attacker intention. Assuming attackers are rational, they inevitably consider time cost factors during attacks. If an attack behavior has consumed a long time without progress, the attacker' s intention will significantly decrease. Based on this analysis, we introduce capability factors and time factors to dynamically analyze and evaluate attacker intention.

2.2.1 Risk Factor

$Risk(a)$ represents the severity of risk caused by a single attack behavior. The value of the risk factor is related to the attributes of the vulnerability exploited by the attack. The greater the impact of a vulnerability, the higher the risk of attack behavior exploiting that vulnerability. The more severe the risk caused by an attack, the greater the possibility that the attack behavior will be discovered by network administrators, thus the higher the cost required for successful attack, and the weaker the attacker intention. Therefore, we can use vulnerability impact to measure the risk generated by attacks. Let the vulnerability exploited by attack a be v. Referring to the vulnerability impact calculation formula in CVSS, we obtain the risk factor calculation formula:

$$Risk(a) = 10 \times (1 - C) \times (1 - I) \times (1 - A) \quad (2)$$

Here, C, I, and A represent the confidentiality metric value, integrity metric value, and availability metric value of the vulnerability in CVSS, respectively. The value of attack risk $Risk(a)$ ranges from 0 to 10, with higher values indicating greater impact caused by the vulnerability.

2.2.2 Capability Factor

The capability factor Atr represents the influence degree of attacker capability on the risk factor. Equation (3) shows the calculation method of capability factor proposed by predecessors, where N is the number of attacks, α is the capability coefficient, and β is the risk coefficient whose value is given by expert experience. This method uses the number of attacks to quantify attack capability without considering capability gaps between different attackers. If the

attacker capability level is high, the attack complexity will be reduced. Since attacker capability level is inversely proportional to attack risk, the attacker' s intention to launch attacks strengthens. The improved capability factor calculation formula is as follows:

$$Atr = \frac{1}{1 + \ln(1 + \alpha \times N_{Time})} \quad (4)$$

Here, N_{Time} represents the number of time units consumed by the attack, and α is the capability coefficient used to correct the value of the capability factor. Therefore, the improved formula is:

$$Atr = \frac{1}{1 + \alpha \times (1 - Attabi)} \quad (5)$$

Here, *Attabi* represents the attacker capability level, and α is the adjustment factor. Unlike traditional methods, if the attacker capability is weak, according to Equation (1), the capability level value approaches 0, and the capability factor *Atr* value will not be too low but close to 1. Moreover, the value range of the capability factor is more moderate, avoiding large gaps in capability factor values under different capability levels.

2.2.3 Attack Profit

Attack profit measures the degree to which resources obtained by the attacker benefit their attack intention when an atomic attack succeeds. Attack profit has the following properties:
a) The maximum value of attack profit can only equal the value of the attack target asset.
b) The profit value is determined based on the impact of the attack behavior on the target host, i.e., the change in privileges obtained on the target host, combined with the value of the target resource.

Attack behaviors cause security events to occur. According to the form and impact scope of events, security events can generally be divided into five types: information disclosure, remote login, bypassing authentication, obtaining user privileges, and obtaining root privileges. Due to differences in form and impact scope, the severity levels of different security event categories vary, and the severity of security events is positively correlated with the profit obtained by the attacker. The more severe the event, the higher the operational privileges obtained through the attack, resulting in a higher proportion of profit.

Therefore, based on the severity of security events, attack profit is divided into five levels G1 to G5, where G1 represents the lowest profit level and G5 represents the highest. For calculation convenience, we quantify the profit levels and define the profit factor $\rho \in (0, 1]$, representing the ratio of profit obtained by

the attacker to the target resource value through attack behavior. Higher profit levels correspond to larger profit factor values.

The correspondence between profit levels, security event names, and profit factor values is given in . The value of target resources is determined based on their importance—more important resources have higher values. Then, the profit $Gain(a)$ obtained through attack a is:

$$Gain(a) = (\rho_{now} - \rho_{pre}) \times \omega_j \quad (6)$$

Here, ρ_{now} is the profit factor corresponding to the profit level the attacker will achieve if the attack succeeds, ρ_{pre} is the profit factor corresponding to the profit level the attacker has already obtained before this attack, and ω_j is the target resource.

2.2.4 Time Coefficient

In addition to risk, capability, and profit, attack time consumption also affects attack intention to a certain extent. This paper defines the time coefficient $Time(a)$ to represent the impact of attack duration on attack intention. Specifically, the time spent on an attack is a major factor affecting attack intention. Under certain vulnerability exploitation difficulty, shorter attack time leads to stronger attack intention. As time consumption increases, attacker intention gradually decreases. When the time consumption is very long, the time coefficient value approaches 0. Additionally, we define the time decay coefficient τ to adjust the time coefficient value to better fit general situations. The time coefficient $Time(a)$ is calculated as follows:

$$Time(a) = \frac{1}{1 + \tau \times (t - aSTime)} \quad (7)$$

Here, t represents the current time, $aSTime$ represents the attack start time, and τ is the time decay coefficient. The value of τ is determined based on the specific situation of attack time consumption. If the attack launched by the attacker takes a long time, τ should be assigned a smaller value; if the time consumption is short, τ should be assigned a larger value. The value range of $Time(a)$ is $(0, 1]$. Compared with previous calculation methods, the time coefficient $Time(a)$ has a clear value range and optimized calculation method, thus better reflecting the impact of time cost on attack intention.

2.2.5 Attacker Intention Analysis

When using traditional methods to analyze attack possibilities, only the cost paid for attacks is often quantified. If the attack cost is higher than a set threshold, the attack on that node is abandoned. Although some attacks have high risk and cost, they may yield higher-value resources, in which case the

attacker' s intention remains strong. Based on the ratio of vulnerability risk value to attack profit, combined with analysis of the impact of attacker capability and time coefficient on attack intention, we propose a new attacker intention calculation formula:

$$\kappa = \frac{Gain(a)}{Risk(a)} \times Atr \times Time(a) \quad (8)$$

Here, $Gain(a)$ is the profit, $Risk(a)$ is the vulnerability risk value, Atr is the capability factor, and κ represents the attacker' s intention. A larger value indicates stronger attacker intention. We stipulate that only when κ reaches a certain value will the attacker possibly launch an attack. During the attack process, time cost changes dynamically. As time cost increases, the $Time(a)$ value gradually decreases, attack intention gradually weakens until the value of attack intention κ falls below the threshold, indicating abandonment of the attack.

2.3 Path Reachability Probability and Prediction Algorithm

2.3.1 Path Reachability Probability

Based on the obtained attack path information and the determined attacker intention calculation method, we present the path reachability probability calculation method. Combining conditional probability with attacker intention, we comprehensively consider sub-path reachability and then determine the attack stage that the entire path can achieve and its reachability probability. Since the attack complexity Ac in CVSS represents vulnerability utilization, according to the analysis in Section 2.1 and combining the values of various attributes of attack complexity Ac in CVSS, we can derive the vulnerability utilization values for attackers with different capability levels, as shown in .

Based on the vulnerability utilization value method given in , we provide the formula for calculating vulnerability availability:

$$Vul(v) = Au \times P \times \frac{1}{Ac} \quad (9)$$

Here, Au represents the attack vector of vulnerability v in CVSS, and P represents the authentication metric of vulnerability v, i.e., the number of authentications required to exploit the vulnerability to launch an attack. $Vul(v, Ac, Attabi)$ represents the vulnerability utilization value obtained under the condition of known vulnerability complexity Ac value and attacker capability $Attabi$. For a vulnerability, a smaller Ac value indicates greater difficulty in launching attack behavior exploiting that vulnerability.

Atomic attack-related sub-paths use $Pr(r_i \rightarrow a \rightarrow r_j)$ to quantify the threat possibility of a single state node when an attack succeeds. Let attack a exploit vulnerability v to launch an attack, r_j is the resource node expected to be occupied through the attack, and r_i is the set of prerequisite resources needed to launch the attack. Combining attacker intention, for atomic attack a, the reachability probability of sub-path $r_i \rightarrow a \rightarrow r_j$ is:

$$Pr(r_i \rightarrow a \rightarrow r_j) = Vul(v) \times \kappa \quad (10)$$

Definition 10 (Path Reachability Probability). Assuming the path sequence Seq of path $PATH$ can be divided into multiple sub-paths, each corresponding to an atomic attack process, with sub-paths represented as $r_i \rightarrow a \rightarrow r_j$, the path reachability probability formula is:

$$P(PATH) = \prod_{i=1}^{m} Pr(r_i \rightarrow a \rightarrow r_j) \quad (11)$$

2.3.2 Path Prediction Algorithm

Based on the path reachability probability calculation method in Section 2.3.1, this paper designs the IntenAbi-PathPre algorithm. By calculating sub-path reachability probability, redundant attack paths are eliminated, and the path with the highest reachability probability is selected as the attacker' s priority path.

Algorithm 1 Path Prediction Algorithm
Input: Attack path set $Path$ in the attack graph, vulnerability utilization rate $Vul(e)$ of attack edge e, attack intention κ
Output: Predicted path U_Path

```
1.  for each Path_i in Path
2.    P_{kd}(Path_i) = 1
3.    for each e in Path_i // Traverse edges in the path
4.      if (e in E_2) // If current edge is an attack edge
5.        Calculate Vul(e) and dynamic attack intention
6.        P_{kd}(r_i $\rightarrow$ a $\rightarrow$ r_j) = Vul(e) $\times$
7.        if (P_{kd}(r_i $\rightarrow$ a $\rightarrow$ r_j) <  )
8.           _{ij} = false
9.          Delete corresponding path from Path
10.       else
11.          _{ij} = true
12.         P_{kd}(Path_i) = P_{kd}(Path_i) $\times$ P_{kd}(r_i $\rightarrow$ a $\rightarrow
13.       end if
14.   end for
15. end for
16. Order Path by P_{kd}(Path_i) descending
```

```
17. U_{Path} $\leftarrow$ Path
18. return U_{Path}
```

Lines 3-14 of the algorithm traverse each edge in the path. Lines 4-6 indicate that if the current edge represents an attack, vulnerability utilization and dynamic attack intention are calculated, and their combination yields the sub-path reachability probability. Lines 7-10 indicate that if the sub-path reachability probability is less than the threshold, the attacker abandons the attack on that path, and the corresponding path is removed from the path set. Lines 11-13 indicate that if the reachability probability is greater than the threshold, the sub-path reachability probability is multiplied to obtain the reachability probability of the path to which the sub-path belongs. Lines 17-19 indicate that after calculating the reachability probabilities of all paths, the paths are sorted by priority according to their reachability probability values. Higher priority indicates a greater likelihood of being chosen by the attacker, and the results are provided for network administrators' reference.

3 Experiments

3.1 Experimental Network Environment

As shown in [Figure 2: see original paper], the experimental network is divided into two zones: DMZ zone and Inside zone, isolated by firewalls. The DMZ zone contains one network server H1. The Inside zone contains four servers: FTP server H2, host H3, host H4, and database server H5. In the small LAN, intruders use host H0 to remotely access the target network. H1 in the DMZ zone can connect to the external network and also access FTP server H2, host H3, and host H4 in the Inside network. The Inside zone cannot be directly accessed by the external network. Within Inside, FTP server H2 and host H3 can access H4, and host H4 can access database server H5.

The services running on each host and their corresponding vulnerabilities are shown in . *Ac* is the complexity of the corresponding vulnerability in CVSS, and the attack number represents the attack launched using the corresponding vulnerability.

3.2 Experimental Process

Given the network configuration information, the attack graph generated using the MulVal tool is shown in [Figure 3: see original paper]. Based on the generated network attack graph, all paths to attack target host H5 are generated, with a total of 7 paths reaching host H5. The path information is shown in .

Since the complexity of attack a3 in PATH1 is medium, while the other three attacks a1, a7, and a10 are low, according to Equation (1), the capability required to launch attacks along this path is medium. If the attacker chooses this path, the attacker capability is medium. Similarly, if the attacker chooses PATH3 or PATH4, the most complex attack in the entire path is a4 with high complexity,

so the attacker capability is high. If the attacker chooses any path other than PATH1, PATH3, and PATH4, the attacker capability is low.

Based on host importance, the asset values of H1, H3, and H4 are set to 40, H2' s asset value is 30, and H5' s asset value is 60. According to expert experience, the capability coefficient is set to 0.5. The adjustment factor in Equation (4) and the time decay factor in Equation (7) are obtained through preliminary sample training. Attack time consumption is obtained from intrusion detection system alert information and then substituted into the attack intention analysis formula along with other parameters for calculation.

3.3 Experimental Results Analysis

First, we analyze the dynamic impact of the time decay coefficient on attack intention. According to Equation (6), the variation of time factor $Time(a)$ is shown in [Figure 4: see original paper]. When time consumption is reasonable (0-60 min), the time factor value changes relatively little, generally approaching 1, imposing less constraint on attacker intention. Therefore, under unchanged other parameters, attacker intention is relatively strong. When attack consumption time exceeds the reasonable range and gradually increases, there is a very obvious downward trend until consumption time reaches about 120 minutes, when it has dropped to a very low value, and as time continues to increase, attack intention continues to decline slowly.

After repeated experiments, the attack intention threshold is set to 0.2. For sub-attack a8 in PATH3, the attack time consumption is about 110 minutes, causing the time decay coefficient to drop to 0.17, thereby reducing attack intention to 0.16, which is below the threshold, so the attacker abandons this path. For sub-attack a8 on PATH5, the time consumption is also relatively long, about 100 minutes, causing the time decay coefficient to drop to 0.2 and the attacker' s intention to drop to 0.14, so PATH5 is also abandoned by the attacker. By introducing the time decay coefficient, effective optimization of existing paths is achieved.

Second, attack intention is affected not only by attack time consumption but also by attacker capability. The attacker capability for PATH5 is low, while for PATH3 it is high. According to Equation (4), the lower the capability level, the higher the capability coefficient value, and the larger the capability factor value. According to Equation (7), the larger the capability factor, the smaller the impact on attack risk, and the higher the operational risk. Therefore, the attacker choosing PATH5 pays a relatively high cost when launching attack a8. Although the time decay coefficient of a8 in PATH5 is slightly higher than that of a8 in PATH3, the attacker' s intention for PATH5 is slightly lower than for PATH3.

Taking PATH1 and PATH6 as examples to illustrate the calculation process of single path reachability probability. The first and last attack steps of these two paths are the same. However, since the attacker capability for PATH1 is medium

and for PATH6 is low, the success probabilities and attack intentions of these two single-step attacks in PATH1 are slightly higher than in PATH3. Although attacker capabilities are consistent when single-step attack time consumption is reasonable, the ratio of profit to risk obtained from attacks differs, so the attacker' s intentions for each single-step attack in the same path are not equal. For single-step attacks, those with relatively higher profit ratios have relatively stronger attacker intentions.

Let $P1$ be the path reachability probability results calculated without introducing attacker capability, attack time consumption, and attack intention, and $P2$ be the path reachability probability obtained using the improved algorithm proposed in this paper. Based on the information of all paths in the attack graph given in , we calculate the reachability probabilities of each path under $P1$ and $P2$ respectively, with results shown in [Figure 5: see original paper].

As can be seen from [Figure 5: see original paper], there are 7 reachable paths under $P1$, but this reduces to 5 paths under $P2$. This is due to the optimization effect of the IntenAbi-PathPre algorithm proposed in Section 2.3.2. Based on the previous specific analysis, since the attack intention values of PATH3 and PATH5 are below the threshold, these two paths are abandoned by the attacker after optimization, reducing the probability of confusion from overlapping paths and making the path reachability probability graph clearer and more concise.

Moreover, under $P1$, the reachability probability values of nodes in different paths differ relatively little, and various paths have already become confused at the H4 point and beyond, significantly increasing the difficulty for administrators to identify dangerous paths. However, in $P2$, after incorporating attacker capability and attacker intention into the reachability probability calculation method, although the path reachability probability values decrease to varying degrees for each path, the change trend of reachability probability becomes more moderate, and the differences between paths become more distinct. Although the reachability probability values at H4 and H5 positions are already relatively low, the gaps between paths are very obvious. It is clear that PATH2 and PATH4 have higher attack probabilities. Moreover, although the attackers choosing these two paths have strong capabilities, the differences in reachability probability between paths remain significant. Additionally, although the attacker capability using PATH7 is low, its success probability of intruding node H5 is second only to PATH2 and PATH4, making it also a target that network administrators need to guard against.

3.4 Experimental Results Comparison

To further verify the effectiveness of the attack path prediction method based on the T_{NAG} model proposed in this paper, we reviewed a large number of relevant literature on network attack prediction technology. Through comparison, we found that the method proposed in this paper is fundamentally different from previous research in terms of model concept and prediction method. After

careful screening and analysis, the prediction methods proposed in literature [?] and [?] have certain similarities with this paper. Therefore, we conduct comparisons from three aspects: time variable effect, path reachability probability quantification effectiveness, and prediction accuracy.

a) Time Variable Effect Comparison
For sub-attack a8 in PATH3, the attack time consumption is about 110 minutes. According to the method proposed in this paper, the calculated attack intention is 0.16, so the attacker abandons the attack node, making path PATH3 unreachable. However, according to the method in literature [?], the attack cost value does not change and cannot reflect the impact of the dynamic variable of attack time consumption on attack feasibility.

Literature [?] uses time gain compensation rate to quantify the change trend of time, but the time gain compensation rate is only used for attack graph optimization, i.e., for qualitative analysis of whether a node is reachable or not. For example, for node a7 on PATH1, both 40-minute and 70-minute attack durations result in reachable sub-paths according to literature [?]. However, for reachable nodes, the method in literature [?] cannot reflect the difference in the impact of attack duration length on path reachability probability.

According to the method proposed in this paper, we can calculate that the time coefficients corresponding to 40-minute and 70-minute attack durations are 0.95 and 0.51, respectively. The time coefficient affects the attack intention value, thus resulting in differences in the corresponding path reachability probabilities. Using the attacker intention quantification method with time coefficient proposed in this paper, we can not only qualitatively analyze path reachability to achieve the purpose of eliminating redundant paths and optimizing attack graphs, but also quantify the impact of attack time consumption on path reachability probability, making the results more accurately reflect the influence of attack time consumption on attack behavior.

b) Path Reachability Probability Quantification Effectiveness Comparison
The cost-benefit model proposed in literature [?] and the attack intention quantification model in this paper are both used to evaluate the feasibility of sub-path attacks. In the cost-benefit model, attack cost data comes from the NVD vulnerability database, and profit data comes from asset values—these data are static and cannot characterize the dynamic changes of attack scenarios over time and their impact on path reachability. The method proposed in this paper considers factors affecting path feasibility more comprehensively, including not only static factors such as vulnerability attributes and asset values, but also dynamic factors such as attack time consumption and attacker capability.

Moreover, when calculating path reachability probability, literature [?] stipulates that the reachability probability of a target node equals the recursive multiplication of the reachability probabilities of all direct and indirect parent nodes, equivalent to performing a logical "AND" operation on the reachability

probabilities of all paths connected to the target node. However, this method is suitable for evaluating the security of each node. For path prediction, the path reachability probability calculated by the probability multiplication method is too high, and the gap in reachability between paths is smaller, increasing the difficulty for administrators to identify risky paths. In reality, attackers generally choose only one path to attack. This paper calculates the reachability probabilities of each path separately according to the IntenAbi-PathPre algorithm and determines the most likely attack path by horizontally comparing the reachability probabilities of nodes on the paths, which better conforms to the general pattern of attack behavior.

c) Prediction Accuracy Comparison

To demonstrate the accuracy of attack path prediction using the method proposed in this paper, we conducted multiple experiments, simulated attack processes, collected relevant data, and input the obtained data into the prediction model. The prediction results are compared with those of literature [**?**] and [**?**], with experimental results as follows:

shows that as the number of experiments increases, the accuracy of the prediction method proposed in this paper gradually rises and still shows improvement compared with literature [**?**] and [**?**]. This is because the method fully considers differences between paths, comprehensively incorporates risk, attacker capability, attack time consumption, and attack profit as factors affecting attack intention and thus attack feasibility, and adjusts the vulnerability exploitability distribution according to attacker capability to calculate path prediction results. These improvements ensure the efficiency of the prediction method.

4 Conclusion

In today' s environment of increasingly complex and varied attack methods, how to effectively predict attack paths, accurately identify attacker intentions, and thus provide references for network security protection is a problem that needs to be solved. This paper models elements of the physical network and uses attack graphs to describe the relationship between attacks and resources. We distinguish and quantify attacker capabilities based on attacker behavior, propose the concept of attacker intention, add a time decay factor to the quantification of intention, and make reasonable judgments about path reachability from the perspective of attacker subjective intention, achieving attack graph optimization and path prediction to provide reference basis for network administrators.

References

[1] National Internet Emergency Center. Summary of Internet Security Situation in China in 2018 [EB/OL]. (2019-04-17) http://www.cac.gov.cn/2019-04/17/c_{1124379080}.htm

[2] Radware. Global Application & Network Security report 2018-2019 [EB/OL]. (2019-04-03) https://www.radware.com/ert-report-2018

[3] Phillips C A, Swiler L P. A Graph-based System for Network-vulnerability Analysis [J]. Proceedings of the Workshop on New Security Paradigms, 1998: 71-79.

[4] Munoz-Gonzalez L, Sgandurra D, Barrere M, et al. Exact Inference Techniques for the Analysis of Bayesian Attack Graphs [J]. IEEE Transactions on Dependable and Secure Computing, 2017: 1-1.

[5] Shameli-Sendi A, Dagenais M, Wang L. Realtime Intrusion Risk Assessment Model based on Attack and Service Dependency Graphs [J]. Computer Communications, 2018 (116), 253-272.

[6] Gao Ling, Wang Fan, Gao Ni, et al. Protection Strategy Selection Model Based on Improved Ant Colony Algorithm [J]. Computer Engineering and Applications, 2019, 55 (07): 100-107.

[7] Hu Hao, Ye Runguo, Zhang Hongqi, et al. Quantitative Method of Network Security Situation Based on Attack Prediction [J]. Journal of Communications, 2017, 38 (10): 122-134.

[8] Ye Ziwei, Guo Yuanbo, Wang Chendong, et al. Review of Attack Graph Technology [J]. Transactions of Beijing Institute of Technology, 2017, 38 (11): 121-132.

[9] Qin Hu, Wang Jianli, Peng Xiaoyao, et al. Attack Graph Generation Method Based on Privilege Lifting Matrix [J]. Transactions of Beijing Institute of Technology, 2019, 39 (01): 101-105.

[10] Ghasemigol M, Ghaemi-Bafghi A, Takabi H. A Comprehensive Approach for Network Attack Forecasting [J]. Computers & Security, 2015, 58: 83-105.

[11] Bi K, Han D, Wang J. K maximum probability attack paths dynamic generation algorithm [J]. Computer Science and Information Systems, 2016, 13 (2): 677-689.

[12] Gao Ni, Gao Ling, He Yiyue, et al. Dynamic Security Risk Assessment Model Based on Bayesian Attack Graph [J]. Journal of Sichuan University (Engineering Science Edition), 2016, 48 (01): 111-118.

[13] Hu Hao, Liu Yuling, Zhang Hongqi, et al. Network Intrusion Path Prediction Method Based on Absorbing Markov Chain [J]. Journal of Computer Research and Development, 2018, 55 (04): 831-845.

[14] Dewri R, Ray I, Poolsappasit N, et al. Optimal security hardening on attack tree models of networks: a cost-benefit analysis [J]. International Journal of Information Security, 2012, 11 (3): 167-188.

[15] Dai F, Hu Y, Zheng K, et al. Exploring risk flow attack graph for security risk assessment [J]. Iet Information Security, 2015, 9 (6): 344-353.

[16] Wang Hui, Wang Yincheng, Lu Shikai. Attack Path Optimization Algorithm Based on Time Gain Compensation rate [J]. Computer Engineering, 2018, 44 (08): 184-191+198.

[17] HOMER J, ZHANG S, OU X, et al. Aggregating vulnerability metrics in enterprise networks using attack graphs [J]. Journal of Computer Security, 2013, 21 (4): 561-597.

[18] Wang Shuo, Tang Guangming, Kou Guang, et al. Attack Path Prediction Method Based on Causal Knowledge Network [J]. Journal of Communications, 2016, 37 (10): 188-198.

[19] Mell P, Scarfone K, Romanosky S. Common vulnerability scoring system [J]. IEEE Security & Privacy Magazine, 2006, 4 (6): 85 –89.

[20] Gao Ni, Gao Ling, He Yiyue, Wang Fan. The Model of Selecting The Best Security Policy Based on Bayesian Attack Graph [J]. Computer Engineering and Applications, 2016, 52 (11): 125-130.

Note: Figure translations are in progress. See original paper for figures.