IET Information Security

Research Article

Framework for risk assessment in cyber situational awareness

ISSN 1751-8709
Received on 8th May 2018
Revised 31st August 2018
Accepted on 27th September 2018
E-First on 16th November 2018
doi: 10.1049/iet-ifs.2018.5189
www.ietdl.org

Xi Rongrong[1] ✉, Yun Xiaochun[1], Hao Zhiyu[1]
[1]Second Research Laboratory, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100089, People's Republic of China
✉ *E-mail: xirongrong@iie.ac.cn*

Abstract: A large number of data is generated to help network analysts to evaluate the network security situation in traditional detection and prevention measures, but it is not used fully and effectively, there is not a holistic view of the network situation on it for now. To address this issue, a framework is proposed to evaluate the security situation of the network from three dimensions: threat, vulnerability and stability, and merge the results at decision level to measure the security situation of the overall network. In the case studies, the authors demonstrate how the framework is deployed in the network and how to use it to reflect the security situation of the network in real time. Results of the case study show that the framework can evaluate the security situation of the network accurately and reasonably.

1 Introduction

The cyberspace provides convenience and benefits for people with its fast development, but it also makes people face the challenge of network security. There is a high demand for network analysts to know about the security situation of the network easily and comprehensively. The capability of understanding the security situation of the network can help network analysts know whether or not their network is secure, and help them to make decisions. Although various security measures such as firewalls and intrusion detection systems have been deployed to detect and prevent attacks in the network, these measures often generate massive alerts as well as numerous false positives and false negatives. It is difficult for network analysts to understand and manage the extremely large amount of network alerts. Therefore, it is needed to develop effective mechanisms which can consolidate all available information and provide high-level view of the security situation of the network, so as to help network analysts make decisions.

In this paper, we propose a framework, which can fuse all available information to evaluate the security situation of the network. It has the following features:

i. The framework can provide comprehensive network security situational assessment by merging multi-source data. Traditional assessment methods evaluate network security based on single data sources such as vulnerabilities, alerts or network traffic. They cannot provide a comprehensive assessment. The framework merges all available information to evaluate the security situation of the overall network.
ii. The framework can improve the validity of alerts by alert verification. There are many false-positive or irrelevant alerts in raw collected alerts, which impact the accuracy of the assessment. The framework filters out these imperfect alerts by matching them with the configuration of the target network to improve the accuracy of the assessment.
iii. The framework can fill the missing value of the temporal metric in common vulnerability scoring system (CVSS) based on their statistical features. CVSS omits the value of temporal metrics because it is difficult to obtain. The framework fills the missing value of temporal metrics by researching their statistical features. The improved severity of vulnerabilities makes assessment more objectively and accurately.

The remainder of this paper is organised as follows. In Section 2, we review the related works. Section 3 presents the overall architecture of the framework and its detailed description. We discuss the case study in Section 4. Conclusions and future works are discussed in Section 5.

2 Related works

To our knowledge, some works have been done in cyber situational awareness. Numerous mature tools have been developed. NVisionIP [1] and VisFlowConnect-IP [2] present a visual representation of the flow on a single screen; SiLK tool suite [3] is a highly scalable flow-data capture and analysis system developed by the network situational awareness group. These tools focus on capturing and analysing network traffic. However, there is lack of consolidation with other network information.

On the other hand, several techniques of cyber situational awareness have been proposed in recent years. The techniques can fall into three categories: vulnerability-based assessment, alert-based assessment and honeynet-based assessment. Jajodia *et al.* [4] and Wang *et al.* [5, 6] focus on analysing network security based on vulnerabilities and their logical relations. They analyse vulnerabilities and their interdependencies by attack graph. It can convey the impact of individual vulnerability and combined vulnerabilities on overall security. On the basis of the attack graph, they can obtain high-level view of network security situation. Xu and Ning [7], Zhai *et al.* [8] and Allodi and Massacci [9] analyse network security based on attack scenarios, which are obtained by correlating alerts. Barford *et al.* [10] and Thonnard and Dacier [11] collect and analyse numerous honeynet information to extract attack model to assess network security. However, these methods, respectively, focus on the single data source and lack of data consolidation. It is needed to consolidate all available information to improve the accuracy of the assessment.

There are some techniques that consolidate all available information to analyse network security. Research of M2D2 [12], the mission-impact-based correlation method [13] and CVSS-based risk assessment [14] are closest to ours. M2D2 is a formal model, which correlates alert with multi-sources including characteristics of monitored systems, properties of security tools and observed events; it does not provide a specific mechanism for automatic reasoning. The mission-impact-based method ranks alert based on their relationships with critical resources, but it requires human experts to specify the correlation models. The CVSS-based risk assessment evaluates the risk of the network based on CVSS and attack modelling. However, the premise of their method can accurately assess the risk of the network is that ‘attack action is

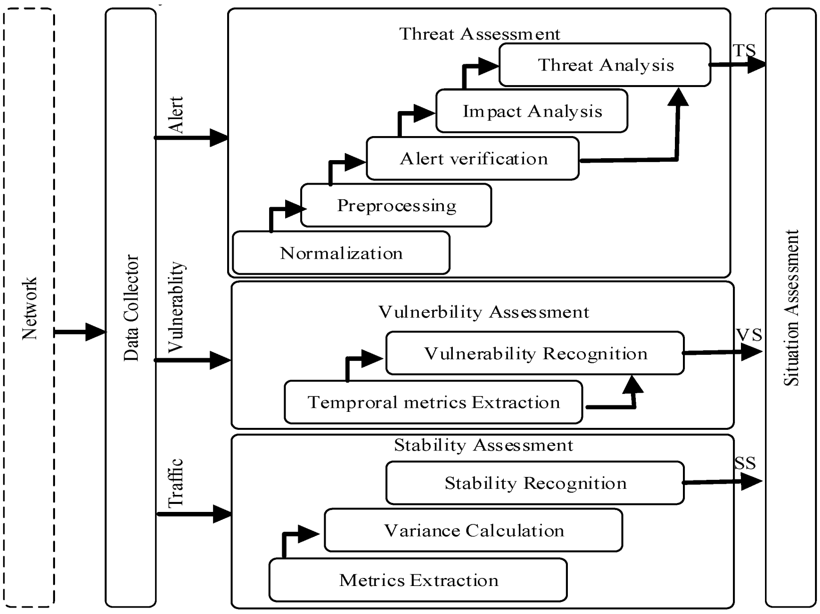


Fig. 1 *Overview of assessment architecture*

Table 1 Common format for alert attributes

Attribute	Description
analyser	information identifying the sensor
time	time the alert is detected
alert-name	information identifying the alert
sourceIP	source IP of the events leading up to the alert
sourceport	source port of the events leading up to the alert
destinationIP	target IP of the events leading up to the alert
destinationport	target port of the events leading up to the alert
classification	'type' of the alert, which determines how to distinguish the alert
completion	probability of the event success
severity	impact of the event on the target

exploitation of the vulnerability of software or hardware'. However, we know that not all attacks exploit the vulnerability. For example, port scans only probe information from the target network. It does not exploit any vulnerability. Therefore, the premise restricts the accuracy of their method. In this paper, we provide a framework to assess the security situation of the network, which can consolidate all available data to assess network security situation quantitatively and automatically.

3 Assessment architecture

The security of the network is concerned not only with threats, but also with vulnerability. Moreover, it changes over time. To measure time-dependent change, network stability is introduced to assess the security situation of the network. Therefore, the assessment architecture is constructed based on three dimensions: threat, vulnerability and stability. It is shown in Fig. 1.

There are four modules in this architecture, which are threat, vulnerability, stability and situation assessment modules. In details, the threat assessment module evaluates the threat of the network based on the alerts. Vulnerability assessment module utilises CVSS to evaluate the vulnerability of the network. Stability assessment module evaluates the stability of the network by analysing the abnormal traffic. Situation assessment module consolidates the results of sub-assessment to measure the security situation of the overall network.

3.1 Threat assessment

The amount of alerts collected from detection and prevention devices is massive, and many of them are false positive or irrelevant. That reduces the validity of alerts and impacts the accuracy of threat assessment. To improve the accuracy of assessment, threat assessment handles alerts as follows.

3.1.1 Normalisation: The framework normalised alerts into a unified common format, which ensures that it can be compatible with various intrusion detection sensors.

The most widely accepted format for alerts is intrusion detection message exchange format (IDMEF) [15] data model. So we adopt the assessment class in IDMEF as the common format. The attributes in common format are shown in Table 1.

Most fields of the common format can be copied from attributes of raw alert directly. However, classification, completion and severity fields are missing. Since most intrusion detection sensors do not provide information about them. In this framework, the classification field is assigned by looking up a database. The database contains a mapping from alert-name into its corresponding classification. It is manually created by analysing the rules of intrusion detection system (IDS). Completion and severity fields will be assigned in alert verification and impact analysis component.

3.1.2 Preprocessing: The framework merges multiple alerts with the same attributes into a hyper-alert, which can reduce redundant alerts.

The framework keeps a sliding time window to preprocess the normalised alerts. When a new alert arrives, it is compared with the alerts in the queue, starting with the alert with the earliest timestamp. A match is found if all overlapping attributes are equal and a new hyper-alert is produced. On finding a match, the two alerts are merged into a hyper-alert to replace the matched alert in the queue, and the search is terminated. If no match is found after searching through the whole queue, the alert is inserted into the queue, to be considered for matching with future alerts.

3.1.3 Alert verification: The framework filters out false positives/irrelevant alerts by verifying them, which can improve the validity of alerts.

Alert verification is implemented by matching the requirement of alerts with the configuration information of the target network. When a preprocessed alert is detected, its requirement is matched with the configuration information stored in the topology database. When the configuration information satisfies the requirement, the alert is flagged as successful; when the configuration information does not satisfy the requirement, the alert is flagged as unsuccessful; when the configuration information is missing, the alert is flagged as undetermined.

For undetermined alerts, we extract its appropriate Nessus Attack Scripting Language script based on its corresponding common vulnerabilities and exposures (CVEs) identifiers and run the script in the target network. If the target network is vulnerable to the script, the alert is tagged as successful. Otherwise, it is assumed to be unsuccessful.

3.1.4 Impact analysis: The framework quantifies the impact of attacks by analysing the severity of their corresponding alerts.

When a verified alert is detected, we extract its corresponding CVE_ID and take its CVSS score as the severity of the alert. If the alert has no corresponding CVE_ID, a default value is assigned to it. The default value is determined by averaging the severity of all alerts in its classification. The classification is obtained from a mapping database, which is constructed based on Snort rules.

3.1.5 Threat analysis: The framework quantifies the threat situation of the network based on the likelihood and the impact of attacks [16]. It is calculated using the following equation:

$$\text{TS} = \frac{1}{n}\sum_{i=1}^{n}(\text{hyper\_alert}[i]\cdot\text{completion})\times(\text{hyper\_alert}[i]\cdot\text{severity}) \tag{1}$$

where n denotes the number of hyper-alerts, the completion denotes the successful probability of attacks, and the severity denotes the impact of attacks. These attributes are assigned by alert verification and impact analysis. The value of TS ranges from 0 to 10. The higher its value is, the less secure the network will be.

3.2 Vulnerability assessment

Vulnerability assessment module evaluates the vulnerability situation of the network by measuring the severity of vulnerabilities. The most widely accepted quantitative standard for measuring the severity of vulnerabilities is CVSS. It provides a base, temporal and environmental metrics for evaluating the severity of vulnerabilities. However, the value of temporal metrics is missing because it is difficult to obtain. In practise, collecting temporal and environmental information represents a significant organisational or financial effort. While researchers have found that the severity of vulnerability varies greatly among different temporal metrics. For example, Allodi and Massacci [9] proposed an operative framework for security studies by using a case-control study methodology. They found that the CVSS base score alone is a poor risk factor from a statistical perspective. However, the existence of a proof-of-concept exploit is an interesting risk factor to consider. Nappa *et al.* [17] investigated the patch deployment process for 1593 vulnerabilities from ten client applications. They found that risk assessment should take into account the milestones in the vulnerability lifetime such as the patching delay and the median time-to-patch. Wang *et al.* [18] proposed a novel security metric, *k*-zero day safety, to measure the risk of the network. It is noted that they defined the level of effective patching as a sub-metric to measure the networks' current security. It shows that they think that the remediation level of vulnerability is an important parameter for the security of the network. These studies denote that the CVSS base score cannot measure the vulnerability of the network accurately, and its temporal metrics are an important factor to consider. Therefore, we study the temporal metrics based on their statistical features.

3.2.1 Temporal metric extract: Researchers have conducted a comprehensive study in the temporal metric of vulnerabilities [19, 20]. They found that the temporal metrics of vulnerabilities are related to various attributes such as type, severity, vendor, product and their disclosure-, exploit- patch-date *et al*. These attributes determine the temporal metrics of the vulnerability. To analyse the effect of these attributes on temporal metrics, we describe them in detail as follows:

(i) *Vulnerability type:* There are many types of vulnerabilities in the network. However, the number of many vulnerabilities is very few. To make statistically sound observations, only the prevalent types of vulnerabilities are extracted. They are clustered into PHP vulnerabilities (PHP), executable code extension of executable program (EXE), denial of service (DOS), buffer overflow (BO), structured query language (SQL) injection (SQL), cross-site scripting (XSS) and miscellaneous vulnerabilities (Misc).
(ii) *Severity:* We divide the severity of vulnerabilities based on CVSS scores. They are divided into three categories: Low: 0_CVSS Score<4; Medium: 4_CVSS Score <7; and High: 7_CVSS Score_ 10.
(iii) *Vendor and product:* There are more than 11,000 vendors and over 17,000 software products for vulnerabilities. However, over 95% of the vendors have <10 vulnerabilities [20]. Therefore, we focus on the top seven vendors each of which has at least 500 vulnerabilities. They include Microsoft, Apple, Sun, Oracle, VMware, Mozilla and Google. We also study popular software products of these vendors that include Internet Explorer, Safari, Firefox, Chrome, Windows, Linux, Red Hat, Solaris and several Linux-based operating systems.
(iv) *Exploit-disclosure (t_{ed}):* It is the duration (in days) between the dates of an exploit for a given vulnerability was provided by hackers and the date the vulnerability was disclosed. There are three cases for the ranges of t_{ed} values. $t_{\text{ed}}<0$ shows that an exploit for a given vulnerability was released before its public disclosure. $t_{\text{ed}}=0$ refers to the case when an exploit for a given vulnerability was released on the day it was disclosed. $t_{\text{ed}}>0$ means that the exploit for a vulnerability was released after its public disclosure.
(v) *Patch-disclosure (t_{pd}):* It is the duration (in days) between the dates of a patch for a given vulnerability was released by the vendor and the date the vulnerability was disclosed. Its range of value is the same with t_{ed}. $t_{\text{pd}}<0$ shows that the patch for a given vulnerability was released before its public disclosure. $t_{\text{pd}}=0$ means that the patch for a vulnerability was released on the disclosure day. $t_{\text{pd}}>0$ refers to the case where the patch for a given vulnerability was released after its public disclosure.

On the basis of these attributes, we use the same rule mining algorithm with Muhammad to extract rules about the exploitation behaviour and patching behaviour [20]. The rules can be formalised as:

$$< vnd >< prod >< typ >< sev > \rightarrow < \boldsymbol{t_{ed}} \parallel \boldsymbol{t_{ed}} >$$

where < *vnd* > denotes vendor name, < *prod* > denotes product name, < *typ* > denotes vulnerability type and < *sev* > denotes severity. Some rules are illustrated as follows:

Rule1: vnd = Microsoft; typ = XSS||DOS||BO; sev = H → t_{ed} = 0day

It means that in the case of Microsoft, the majority of vulnerabilities including DOS, XSS, and BO are exploited on the day they are disclosed.

Rule2: vnd = Apple; prod = MAC OS; typ = DOS→ 0 < t_{pd} < + 1week

It shows that Apple generally takes about a week to fix DOS vulnerabilities even if they are exploited on the day they are disclosed.

On the basis of these rules, for any given vulnerability, we can infer its exploitation/patching behaviour and determine its value of t_{ed} and t_{pd}. Then, we assign a value for exploit code maturity and remediation level according to the value of t_{ed} and t_{pd}. The discussion of report confidence is outside of the scope of this paper. We assign it as the default value. The temporal score is assigned using the following code:

```
switch($t_{ed}$) {
  $t_{ed}<0$: Exploit Code Maturity is assigned as HIGH. break;
  $t_{ed}=0$: Exploit Code Maturity is assigned as Proof-of-Concept. break;
  $t_{ed}>0$: Exploit Code Maturity is assigned as Unproven. break;}
```

```
switch($t_{pd}$) {
$t_{pd}$<0: Remediation Level is assigned as Official Fix. break;
$t_{pd}$ = 0: Remediation Level is assigned as Temporary Fix. break;
$t_{pd}$>0: Remediation Level is assigned as Unavailable. break;
Temporal Score = round-to-1-decimal (Base Score * Exploit Code Maturity*Remediation Level * Report Confidence)}.
```

3.2.2 Vulnerability recognition: Vulnerability recognition assesses the vulnerability situation of the network using the average severity of all vulnerabilities in the protected network. It is noted that their severity has been modified by adding the temporal metric to its CVSS base score, which makes the assessment more accurately. It can represent by the following equation:

$$\mathrm{VS} = \frac{1}{n}\sum_{i=1}^{n} v_i \cdot \text{temporal score} \tag{2}$$

where v denotes the vulnerability and n denotes the number of vulnerabilities in the protected network. The value of VS ranges from 0 to 10.

3.3 Stability assessment

Stability assessment module evaluates the stability situation by analysing the abnormal traffic. Owing to the nature of protocols, we can get the conclusion that sudden changes in traffic are indications of some intrusions. Therefore, we adopt the metric $x = (\text{incoming packet}/s)/(\text{outgoing packet}/s)$ to describe the stability situation of the protected network.

There is a clear symmetry in transfer control protocol (TCP) flow that makes (incoming TCP packet/s)/(outgoing TCP packet/s) fairly stable with a value close to one [21]. For UDP and ICMP flows, there is not a clear symmetry, but there is still a fairly stable site-dependent behaviour (ratio value). So we adopt the following metrics to measure the stability situation of the network:

$$\mathrm{TR} = \frac{\text{incoming TCP packet/s}}{\text{outgoing TCP packet/s}}$$

$$\mathrm{UR} = \frac{\text{incoming UDP packet/s}}{\text{outgoing UDP packet/s}}$$

$$\mathrm{IR} = \frac{\text{incoming ICMP packet/s}}{\text{outgoing ICMP packet/s}}$$

On the basis of the analysis of these metrics, we can conclude that if ratio of transfer control protocol (TR), ratio of user datagram protocol (UR) and ratio of internet control messages protocol (IR) remain stable, then the network is stable and secure; otherwise, if either TR or UR or IR fluctuates, then the network is instable and may be under attack. Therefore, the variances of these metrics are used to measure their stability. In each sampling period, TR, UR, IR and their variances S_{TR}, S_{UR}, S_{IR} are calculated using the following equation:

$$\begin{aligned} S_{\mathrm{TR}} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mathrm{TR}_i - E_{\mathrm{TR}})^2} \\ E_{\mathrm{TR}} &= \frac{1}{n}\sum_{i=1}^{n}\mathrm{TR}_i \end{aligned} \tag{3}$$

where n denotes the sampling times; i denotes the ith variance; and S_{UR}, S_{IR} can be obtained using the same method.

Finally, these variances are integrated into SS using the following equation:

$$\mathrm{SS} = \sum_{i = \mathrm{TR,\,UR,\,IR}} \alpha_i S_i \tag{4}$$

where S_i denotes variances of the different protocol and α_i denotes the weight of different protocol

Table 2 Judgement matrix

	Threat	Vulnerability	Stability
threat	1	2	3
vulnerability	1/2	1	2
stability	1/3	1/2	1

Table 3 Weight of each metrics

	Threat	Vulnerability	Stability
M	6	1	1/6
W	1.82	1	0.55
w	0.54	0.3	0.16

$$\alpha_i = \frac{i\text{ protocol packet}}{\text{all packets}} \tag{5}$$

The value SS ranges from 0 to 10. The higher its value is, the less secure the network will be.

3.4 Situation assessment

Situation assessment module provides a more specific, comprehensive, unified and enhanced security index (SI) to measure the security situation of the overall network by merging the results of sub-assessments at the decision level. It can improve the accuracy of assessment and reduce uncertainty.

There are many data fusion methods. Different methods have different applications. In which, weight analysis method is the simplest and most effective one. It is used to merge the results of sub-assessments and calculate SI in this paper, which can be formalised as follows:

$$\mathrm{SI} = \sum_{i = \mathrm{TS,\,VS,\,SS}} \varpi_i \,\mathrm{Factor}_i \tag{6}$$

where ϖ_i denotes the weight of each sub-assessment.

There are many ways to determine the value of ϖ_i such as Delphi, analytic hierarchy process (AHP), expert prediction etc. In which AHP quantifies the relative importance of each metrics by introducing the digital ratio scale, which makes the weight more reasonable. Therefore, we adopt AHP to determine the weight of each sub-assessment.

Pairwise comparisons are carried out to express the relative importance of one metric over another. For example, in our network, we think the threat is considered to be strongly important than stability and moderately important than vulnerability. These relations are converted into numerical values, which are used to generate a judgement matrix. It is shown in Table 2.

On the basis of the judgement matrix, we can calculate the weight of each sub-assessment using square root law. The process is shown as follows and their results are shown in Table 3:

(i) Calculate the product of each row in the matrix

$$M_i = \prod_{j=1}^{n} b_{ij} \tag{7}$$

(ii) Calculate the nth root of M_i

$$W_i = \sqrt[n]{M_i} \tag{8}$$

(iii) Weight normalisation

$$w_i = \frac{W_i}{\sum_{i=1}^{n} W_i} \tag{9}$$

This determined weight represents the impact of each sub-assessment on the comprehensive security situation. It is noted that the weight of each sub-assessment is determined based on different security requirement of the network.

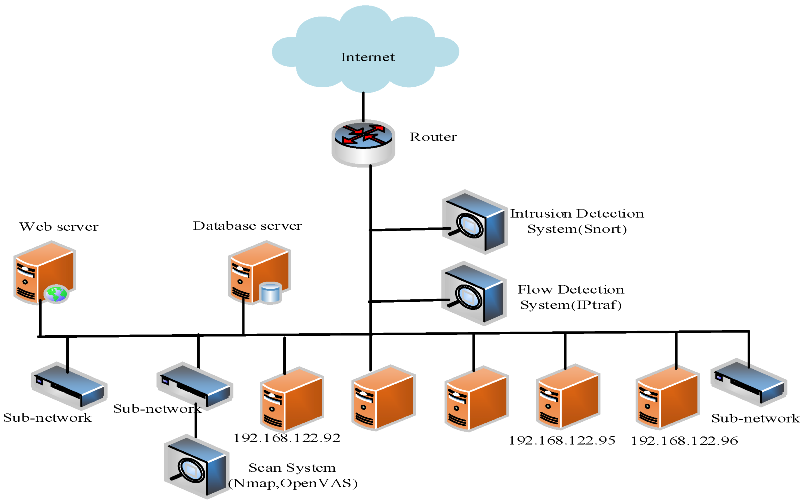


Fig. 2 *Deployment in the case study*

Table 4 Raw data collected in three scenarios

		Scenario 1	Scenario 2	Scenario 3
number of alerts		365	1656	1735
number of vulnerabilities		323	323	148
traffic	incoming	1503	8410	6347
	outgoing	1809	10,269	31,757

Table 5 Results of preprocessing

	Scenario 1	Scenario 2	Scenario 3
number of raw alerts	365	1656	1735
number of hyper-alerts	95	293	283
reduction ratio, %	73.9	82.3	83.7

According to the relative importance in this paper, the weight is set as $\varpi_{\text{TS}} = 0.54$, $\varpi_{\text{VS}} = 0.3$ and $\varpi_{\text{SS}} = 0.16$. On the basis of these weights, the results of sub-assessment are merged into SI, which represents the comprehensive security situation of the network.

4 Case study

This section demonstrates how the framework is used to evaluate the security situation of the network in real time. To our best knowledge, there is no any public test dataset for the evaluation of network situation. Therefore, we set up a test bed and simulate different attack scenarios to demonstrate the performance of this framework.

4.1 System deployment

The deployment in the case study is shown in Fig. 2. The information collector includes three components of the intrusion detector, traffic detector and vulnerability scanner. We deploy the intrusion detector (snort in [22]) and traffic detector (IPtraf in [23]) at the gateway. The vulnerability scanner (OpenVAS in [24]) is deployed in each subnet since normally the firewall rules are not settled to admit the scanner probing to penetrate different subnets. The database server and situation awareness module are deployed in the local network. In this test bed, snort is used as intrusion detection system to monitor attack and generate intrusion alerts. IPtraf is adopted as a flow detection system to record flow information. The vulnerability is collected by OpenVAS. The network topology information is supported by Nmap [25]. Both raw and preprocessed data are stored in MySQL database.

4.2 Dataset

To validate the performance of the framework, we run the system and simulate some intrusion scenarios in our subnet 192.168.122.0/24 during the dates from 2016-03-18 13:30:00 to 2016-03-21 15:40:00. The details of the simulated attacks are described as follows.

Probe attack: An attacker is resident on the host 192.168.122.95 used Nmap to perform probe attack by using of synchronization character (SYN), finish character (FIN) and user datagram protocol (UDP) scanning to subnet 192.168.122.0/24 from 9:30 AM to 10:05 AM on the date of 2016-03-19.

DOS attack: Portscan, smurf and teardrop attacks were launched simultaneously from 11:30 AM to 12:10 PM on the date of 2016-3-20. In details, one attacker resident on the host 192.168.122.95 launched a port scan to the subnet 192.168.122.0/24; one attacker resident on the host 192.168.122.92 carried out smurf attack to victims with Internet protocols (IPs) from 192.168.122.5 to 192.168.122.10 and one attacker launched teardrop attack to victims with IPs from 192.168.122.15 to 192.168.122.20.

4.3 Evaluating the solution

4.3.1 Local evaluation: To highlight our method, we present three representative scenarios to illustrate the executive procedures in details:

Scenario 1 represents a sampling period that the network is in a normal state. It is sampled from 7:30 to 7:35 on 2016-03-19.

Scenario 2 represents a sampling period that the network is suffering probe attack. It is sampled from 9:45 to 9:50 on 2016-03-19.

Scenario 3 represents a sampling period that the network is suffering DOS attack. It is sampled from 11: 50 to 11:55 on 2016-03-20.

The raw data collected in these three scenarios are shown in Table 4.

Table 4 shows that 365 alerts are collected in Scenario 1. The number increased significantly in Scenario 2 and Scenario 3. It shows that network attacks have increased in the latter two scenarios. About 323 vulnerabilities are scanned in Scenario 1 and Scenario 2. While only 148 vulnerabilities were scanned in Scenario 3. This is because hosts in local network updated themselves by installing some patches at 8:00 on 2016-3-20. Table 4 also shows that traffic increased significantly in Scenario 2 and Scenario 3. It is related to attacks. These raw data provide source data for security assessment.

Alerts collected in these three scenarios are used to assess the security situation of the network from the perspective of threat. The results of preprocessing are shown in Table 5.

Table 5 shows that preprocessing can effectively reduce the number of redundant alerts. Its reduction rate can reach above 70%. It suggests that preprocessing can improve the performance of assessment significantly.

Table 6 Packet distribution of each scenario

		Scenario 1	Scenario 2	Scenario 3
TCP	incoming	1492	8394	6334
	outgoing	1796	10,249	31,693
UDP	incoming	21	22	68
	outgoing	2	3	9
internet control messages protocol (ICMP)	incoming	1	9	0
	outgoing	0	2	0

Table 7 Security metrics for three scenarios

	Scenario 1	Scenario 2	Scenario 3
TS	2.61	5.23	7.14
VS	5.08	5.08	3.62
SS	2.22	6.10	9.82
SI	3.29	5.32	6.51

The traffic collected in the three scenarios is analysed using tcpdump. Its results are shown in Table 6.

Table 6 shows that TCP traffic accounts for 99% in all traffic collected in three scenarios. Moreover, the metric $x = (\text{incoming packet/s})/(\text{outgoing packet/s})$ is closer to 1 in the normal network state. It is consisted with our assumption.

On the basis of these collected data, security metrics score of threat (TS), score of vulnerability (VS), score of stability (SS) and security index (SI) can be calculated using our method. The results are shown in Table 7.

Table 7 shows that security metrics (TS, VS, SS, SI) in Scenario 2 are all greater than them in Scenario 1. It indicates that risk is increased in Scenario 2, which corresponds to the probe attack in this period. Table 7 also shows that TS and SS in Scenario 3 are greater than them in Scenario 2. It indicates that network is suffering a more serious attack. We know that the DOS attack launched in this period while VS in Scenario 3 is lower than it in Scenario 2. It is because hosts updated themselves by installing some patches at 8:00 on 2016-3-20. The analyses of three scenarios show that security metrics can reflect the changes in the security state reasonably.

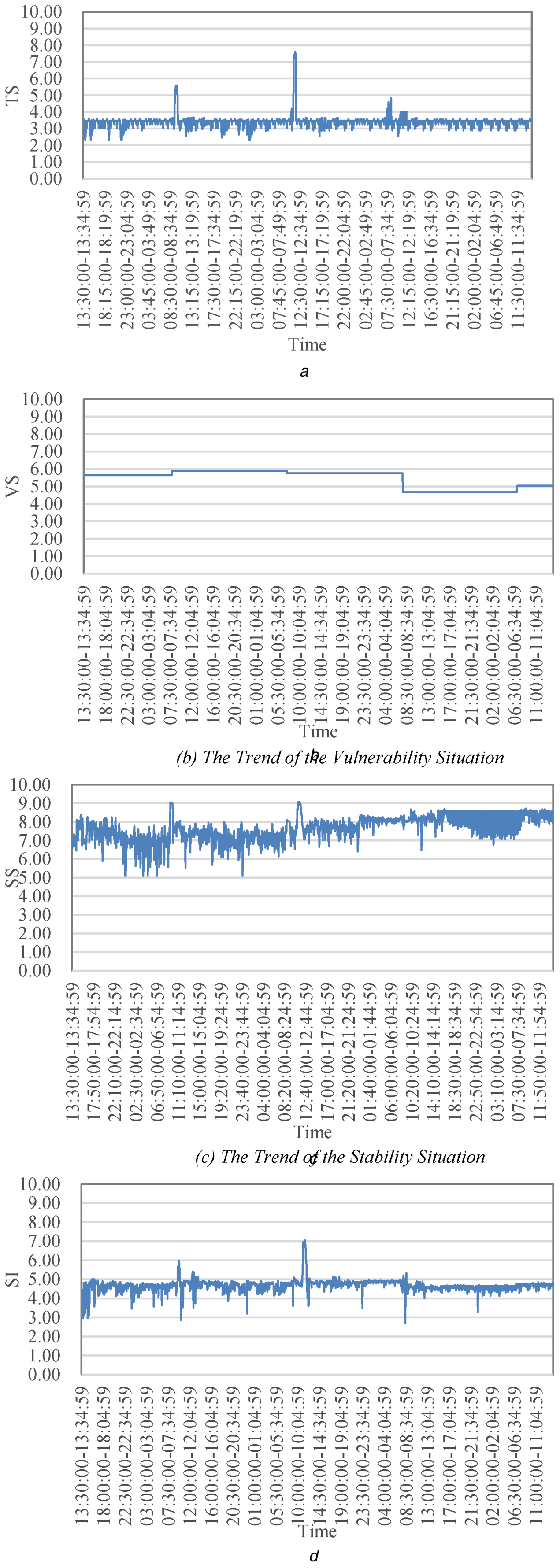
4.3.2 Global evaluation: The situation changes in terms of threat, vulnerability, stability and the whole network security are shown in Figs. 3*a–d*, respectively.

In details, Fig. 3*a* shows the value of TS increased from 3.5 to above 5.2 at 9:30 AM on the date of 2016-03-19, which means that the network becomes unsafe during this period because of some threats. It corresponds to the probe attack started at 9:35 AM on that day. It also shows that the value of TS suddenly increased to about 7.0 at 11:30 AM on the date of 2016-3-20, which indicates there are severity attacks happened during that period. We know that the smurf and teardrop attacks are carried out during this period.

Fig. 3*b* shows the value of VS has small changes at 8:00 AM everyday. That is because that the framework scans the vulnerabilities of the whole network at 8:00 AM everyday. We can also note that there is a significant change at 8:00 on 2016-3-20, which is because the hosts updated themselves by installing some patches. By default, many systems updated themselves on Monday. After the system update, the VS value declines to indicate that the whole network is less vulnerable than before.

Fig. 3*c* shows that the value of SS increases at about indicate the stability of the network was reduced during that time.

Fig. 3*d* shows that the value of SI suddenly increased from 5 to 7 at about 11:30 AM on 2016-3-20. It shows that the situation of network security decreased during that time, which mainly due to the serious threat to the network during that period. According to the simulate attack scenarios, we know that the smurf and teardrop attacks happened exactly during that period. In addition, the value of SI has a minor increase from 5 to 6 at about 9:30 AM on 2016-3-19. It shows that there are slight attacks in this period. The simulated probe attack that happened during that time can explain the increase of SI.



a

(b) The Trend of the Vulnerability Situation

(c) The Trend of the Stability Situation

d

Fig. 3 *Evaluations in case studies*
(a) Trend of the threat situation, *(b)* Trend of the vulnerability situation, *(c)* Trend of the stability situation, *(d)* Trend of the overall security situation

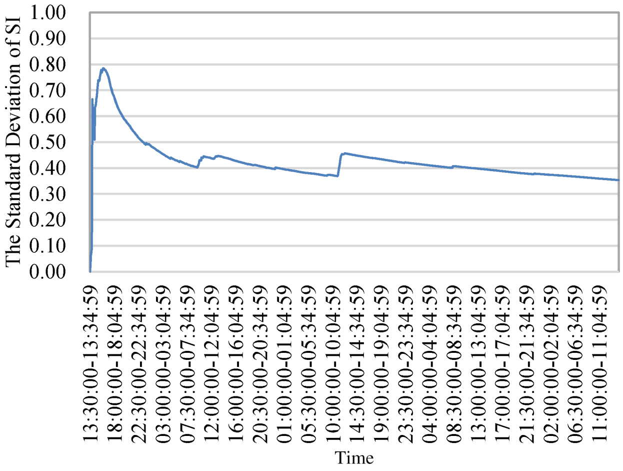


Fig. 4 *Variation trend of SD_{SI}*

4.4 Discussion

We note that all of the measurements on threat, vulnerability and stability reflect the situation changes at two time points in the attack scenarios. These two time points are 9:00 AM on 2016-3-19 and 11:30 AM on 2016-3-20 when the probe attack, the smurf and teardrop attacks were launched. We should also note that though each measurement reflects the normal situation of the network differently, the SI can give an objective assessment of the situation of network security. For example, the value of TS in Fig. 3*a* was around 3.7, which was below its average value of 5.0, indicating that the network is far more safe than that of the normal situation, while the value of SS in Fig. 3*c* was around 8, which was above its average value of 5.0, indicating that the network is more unstable than that of the normal situation. The system will combine all of the estimations from TS, SS and VS to achieve an objective assessment by using of the indicator SI. The value of SI in Fig. 3*d* was around 5.0 when no attacks happened; indicating the situation of the whole network was secure.

To further demonstrate the rationality of SI, we use SD_{SI} (the standard deviation of SI) to analyse its variation. It is defined as follows:

$$SD_{SI} = \sqrt{\frac{1}{(N-1)}\sum_{i=1}^{N}(SI_i - \overline{SI})^2} \qquad (10)$$

where $\{SI_1, SI_2, \ldots, SI_N\}$ are the values of SI in each sampling period, $\overline{SI}$ is the mean value of all SI_i and N is the sampling count.

SD_{SI} is used to quantify the variation range of SI values. A low SD_{SI} means the SI deviation is slight, indicates the security status of the network is stable. While a high SD_{SI} means that SI deviation is acute indicates the security status of the network varies greatly. The variation trend of SD_{SI} is shown as follows.

Fig. 4 shows that the SD_{SI} value fluctuates distinctly at the beginning. It is because SD_{SI} is sensitive to changes in individual sample values when the sample size is small. With the increase of the sample size, the SD_{SI} value tends to be stable. There are other two fluctuations in the later trend. The first occurs around 9:00 AM on 2016-3-19 and the second is at around 11:30 AM on 2016-3-20, which means the security status of the network varies greatly. According to the simulate attack scenarios, we know that the probe attack starts at 9:35 AM on 2016-3-19 and the smurf and teardrop attacks are carried out at 11:30 AM on 2016-3-20. Therefore, the variation trend SD_{SI} demonstrates that SI can be used to measure the network security status reasonably.

5 Conclusion

In this paper, we present a framework to evaluate the security situation of the overall network. The framework evaluates the security situation from three dimensions: threat, vulnerability and stability. Furthermore, the framework fuses the results of sub-assessment at decision level to measure the security situation of the overall network, which can help network analysts to make decisions. It is noted that the validity of the alert is effectively improved by alert verification and the value of temporal metric in CVSS is completed based on their statistical features in this framework, which can improve the accuracy of the assessment.

In the future research, we will further focus on improving the accuracy of the assessment and developing prediction functions to this framework. Improvement and development are already underway.

6 Acknowledgment

This work was supported by the National Key Research and Development Program of China under Grant no. 2016QY04W0804.

7 References

[1] Lakkaraju, K., Yurcik, W., Lee, A.J.: 'NVisionIP: netflow visualizations of system state for security situational awareness'. Proc. 2004 ACM Workshop on Visualization and Data Mining for Computer Security, Washington, D.C., 2004, pp. 65–72

[2] Yin, X., Yurcik, W., Slagell, A.: 'The design of VisFlowConnect-IP: a link analysis system for IP security situational awareness'. Int. Workshop on Information Assurance, College Park, MD, 2005, pp. 141–153

[3] Bandes, R., Shlmeall, T., Heckathorn, M.*, et al.*: 'Analysts handbook: using SiLK for network traffic analysis'. Software Engineering Institute, CERT Program, Pittsburgh PA, 2010

[4] Jajodia, S., Noel, S., OBerry, B.: 'Topological analysis of network attack vulnerability', in Kumar, V., Srivastava, J., Lazarevic, A. (Eds.): '*Managing cyber threats*' (Springer, USA, 2005), pp. 247–266

[5] Wang, L., Singhal, A., Jajodia, S.: 'Measuring network security using attack graphs'. Proc. Third ACM Workshop on Quality of Protection, Alexandria, VA, October 2007, pp. 49–54

[6] Wang, L., Singhal, A., Jajodia, S.: 'Measuring the overall security of network configurations using attack graphs', in Barker, S., Ahn, G.J. (Eds.): '*Data and applications security XXI*' (Springer, Berlin Heidelberg, 2007), pp. 98–112

[7] Xu, D., Ning, P.: 'Alert correlation through triggering events and common resources'. Proc. 20th Annual Computer Security Applications Conf., Tucson, AZ, December 2004, pp. 360–369

[8] Zhai, Y., Ning, P., Iyer, P.*, et al.*: 'Reasoning about complementary intrusion evidence'. Proc. 20th Annual Computer Security Applications Conf., Tucson, AZ, December 2004, pp. 39–48

[9] Allodi, L., Massacci, F.: 'Comparing vulnerability severity and exploits using case-control studies', *ACM Trans. Inf. Syst. Secur. (TISSEC)*, 2014, **17**, (1), p. 1

[10] Barford, P., Chen, Y., Goyal, A.*, et al.*: 'Employing honeynets for network situational awareness', in Jajodia, S., Liu, P., Swarup, V.*, et al.* (Eds.): '*Cyber situational awareness*' (Springer, USA, 2010), pp. 71–102

[11] Thonnard, O., Dacier, M.: 'A framework for attack patterns' discovery in honeynet data', *Digit. Invest.*, 2008, **5**, pp. S128–S139

[12] Morin, B., Mé, L, Debar, H.*, et al.*: 'M2d2: A formal data model for IDS alert correlation'. Recent Advances in Intrusion Detection, 2002, pp. 115–137

[13] Porras, P.A., Fong, M.W., Valdes, A.: 'A mission-impact-based approach to INFOSEC alarm correlation'. Recent Advances in Intrusion Detection, 2002, pp. 95–114

[14] Doynikova, E., Kotenko, I.V.: 'CVSS-based probabilistic risk assessment for cyber situational awareness and countermeasure selection'. Int. Conf. Parallel, Distributed and Network-Based Processing, 2017, pp. 346–353

[15] Debar, H., Curry, D.A., Feinstein, B.S.: 'The intrusion detection message exchange format (IDMEF)', 2007

[16] AS/NZS 4360: risk management. Standards Australia and Standards New Zealand, 2004

[17] Nappa, A., Johnson, R., Bilge, L.*, et al.*: 'The attack of the clones: a study of the impact of shared code on vulnerability patching'. IEEE Symp. Security and Privacy, 2015, pp. 692–708

[18] Wang, L., Jajodia, S., Singhal, A.*, et al.*: '*k*-zero day safety: a network security metric for measuring the risk of unknown vulnerabilities', *IEEE Trans. Dependable Secur. Comput.*, 2014, **11**, (1), pp. 30–44

[19] Frei, S., May, M., Fiedler, U.*, et al.*: 'Large-scale vulnerability analysis'. Proc. 2006 SIGCOMM Workshop on Large-Scale Attack Defense, 2006, pp. 131–138

[20] Shahzad, M., Shafiq, M.Z., Liu, A.X.: 'A large scale exploratory analysis of software vulnerability life cycles'. Proc. 34th Int. Conf. Software Engineering, 2012, pp. 771–781

[21] Siaterlis, C., Maglaris, V.: 'Detecting incoming and outgoing DDoS attacks at the edge using a single set of network characteristics'. IEEE Symp. Computers and Communications, 2005, pp. 469–475

[22] 'Snort-he open source network intrusion detection system'. Available at http://www.snort.org, accessed January 2016

[23] 'IPtraf-an IP network monitor'. Available at http://iptraf.seul.org/, accessed January 2016

[24] 'OpenVAS-open vulnerability assessment system'. Available at http://www.openvas.org/, accessed January 2016

[25] 'Nmap-free security scanner for network'. Available at http://nmap.org/, accessed January 2016