Computer Networks 144 (2018) 216–229

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A systematic evaluation of cybersecurity metrics for dynamic networks

Simon Yusuf Enoch[a,*], Mengmeng Ge[a], Jin B. Hong[b], Hani Alzaid[c], Dong Seong Kim[a]

[a] Department of Computer Science and Software Engineering, University of Canterbury, New Zealand
[b] Department of Computer Science and Software Engineering, University of Western Australia, Australia
[c] Center for CyberSecurity, King Abdulaziz City for Science and Technology, Saudi Arabia

ARTICLE INFO

Article history:
Received 3 December 2017
Revised 22 May 2018
Accepted 30 July 2018
Available online 31 July 2018

Keywords:
Attack graphs
Attack trees
Dynamic networks
Security models
Metrics
Security assessment

ABSTRACT

It is difficult to assess the security of modern networks because they are usually dynamic with configuration changes (such as changes in topology, firewall rules, *etc*). Graphical security models (e.g., Attack Graphs and Attack Trees) are widely used to systematically analyse the security posture of network systems using security metrics. However, there are problems using them to assess the security of dynamic networks. First, most models are unable to capture dynamic changes occurring in the networks over time. Second, the existing security metrics are not designed for the analysis of dynamic networks and hence their effectiveness to the dynamic changes in the network still remains unclear.

In this paper, we systematically categorise network changes into two categories (i.e., changes in hosts and changes in edges). We conduct a comprehensive analysis to evaluate the effectiveness of security metrics using a Temporal Hierarchical Attack Representation Model, which can capture and analyse the changes in the security of network systems. Further, we investigate the varying effects of security metrics when changes are observed in the dynamic networks.

Our simulation results show that different security metrics (except the shortest attack path) have varying security posture changes with respect to changes in the network (when we introduce time to them). However, none of the security metrics consistently changes for all the network changes that we observe in our scenarios. Hence, the results provide some insights into what security metrics can change (accordingly) when a particular network change is observed. It also provides a foundation for further research in this area.

1. Introduction

Modern network technologies are becoming more dynamic with configuration changes. For example, the addition or removal of hosts in a network changes the information and the attack surface of the network [1]. Consequently, the security posture of the network may change as well (an attack surface represents the set of vulnerabilities an attacker can exploit to compromise a target network). So, it is important to assess the changes of dynamic networks in order to understand how the security posture changes.

Graphical security models (GSMs) (e.g., Attack Graphs (AGs) [2,3] and Attack Trees (ATs) [4,5]) are widely applied to systematically assess the security of networks, using various security metrics. However, there are problems in the GSMs being applied to dynamic networks. First, research on GSMs did not consider how the security assessment can be affected when the network changes. Thus, it becomes difficult to analyse the security of networks that are dynamic.

Second, previous research assumed that GSMs use static information, e.g., the hosts, edges (i.e., links connecting hosts), services running on the hosts, vulnerabilities, etc. But those components change over time and as such, the security analysis using those models is only applicable to one particular state of the network. In addition, the existing security metrics are not capable of representing the security posture of different states of the dynamic networks, where state refers to the configurations and settings of the network at a time *t*. For example, the probability of an attack success will change as the network topology is reconfigured because the previously identified attack paths have changed. Hence, it is important to investigate how different security metrics are affected by changes in the network, in order to identify which ones are (not) suitable for dynamic analysis.

The focus of this work is: (1) to use a GSM to capture and model all possible attack scenarios for dynamic networks, (2) to investigate how different security metrics are affected by changes

[*] Corresponding author.
E-mail addresses: sey19@uclive.ac.nz, simon.yusuf@pg.canterbury.ac.nz (S.Y. Enoch), mge43@uclive.ac.nz (M. Ge), jin.hong@uwa.edu.au (J.B. Hong), hmalzaid@kacst.edu.sa (H. Alzaid), dongseong.kim@canterbury.ac.nz (D.S. Kim).

https://doi.org/10.1016/j.comnet.2018.07.028
1389-1286/© 2018 Elsevier B.V. All rights reserved.

in the network, to identify which ones are (not) suitable for the analysis of dynamic networks.

In this paper, we use a dynamic GSM, more specifically a Temporal-Hierarchical Attack Representation Model (T-HARM) [6], in order to capture and analyse the changes in the security posture of dynamic networks. By using T-HARM, we can assess the security of the dynamically changing network with security metrics at different time points. Besides, we can observe how these security metrics change over the time with respect to changes in the network.

An earlier version of this paper appeared in [7], and we extend it with (1) a systematic categorisation of network changes, (2) a formal definition of the network changes, (3) a comprehensive security analysis with all potential network changes and (4) a categorisation of the effectiveness of security metrics with respect to the various changes in the dynamic networks.

The main contributions of this paper are summarised as follows:

- to systematically categorise all potential network changes (Section 4.1);
- to formally define the network changes w.r.t. a T-HARM (Section 4.2);
- to conduct a comprehensive security analysis with all potential changes in T-HARM (Section 5.4);
- to investigate the effect of the various network changes on the existing security metrics over time (Section 5.4);
- to categorise the effectiveness of the security metrics w.r.t. the security changes (Table 8).

The rest of the paper is organised as follows. Section 2 summarises the related work on the existing GSM approaches for enterprise networks and existing security metrics. Section 3 summarises the technical background for the GSM and security metrics used in this study. Section 4 presents the categorisation of network changes and the formalism of the network changes. Our system model, attacker model, simulation analysis and results are described in Section 5. Section 6 presents the limitations and extensions of our work. Finally, Section 7 concludes this paper.

2. Related work

We discuss related work on GSMs and security metrics in this section.

2.1. Graphical security models

A GSM is a tool for security assessment of real-life systems [8] (most popular applications domain is in: Internet attack [9,10], voting systems [11,12], supervisory control and data acquisition systems [13,14], online banking systems [15] *etc.*). Our interest is using GSMs for the analysis of cyber-attack and defence scenarios. We discuss it in two aspects: GSMs for static networks and GSMs for dynamic networks.

GSMs for static networks: Several papers addressed the problem of assessing the security of network systems. One of the earlier work is [16] where an AG is proposed to model computer attacks. In particular, the AG is used to show all possible sequences of attack steps to gain access to a target using network reachability information and a set of vulnerability. Some other graph-based approaches for assessing the security of network systems include [2,3,17–21] *etc*. However, analysing all possible sequence of attack paths using the AG has a scalability problem. As a result, various work proposed different approaches to improving the scalability of the AG [3,22–25]. For instance, Homer et al. [24] proposed two approaches. First, they proposed an approach that automatically identifies the portions of an AG that is not important in understanding the core security problems and subsequently, removed it. Secondly, they proposed an approach that grouped similar attack steps which they said it represents the number and type of security problems. Ingols et al. [26] proposed a Multiple Prerequisite Attack Graph (MPAG) which grouped multiple subsets of nodes in order to reduce the size of the AG. Even at that, the generation of the MPAG still suffers from the scalability problem since modern networks have become very large and highly dynamic [27].

Another type of the GSMs is the tree-based models such as AT in [5,28–30] *etc*. The AT is a tree-like structure which systematically presents attack scenario in a network with the target as the root node and the different ways of reaching the target as leaf nodes. They are used to analyse the security of systems but they cannot be generated from network system specifications [31] (e.g., using hosts reachability information to know how an attacker can move from one host to another host (i.e., cannot capture the attack paths information explicitly)).

To address these problems, a multi-layer security model named HARM was proposed by Hong and Kim [27] which simplify the evaluation of all possible attack scenarios. In particular, they presented a three-layer HARM where the reachability of network subnets is captured at the upper layer, the host reachability information for each subnet is captured in the middle layer (using AG) and the vulnerability information is captured in the lower layer (using AT). Further, they showed that the scalability and the computational complexity is improved when more layers (hierarchy) are used in the multi-layer HARM.

With respect to our research, the major challenges with the aforementioned approaches are that, they do not take into account the various changes that happen in the network for their security analysis. But modern networks are now dynamic. For instance, a network attack surface changes when a new host is connected to the network (e.g., bring your own device [32]), update of software vulnerabilities [33], the discovery of new vulnerabilities [33,34], firewall configuration and settings changed, *etc* and hence, those GSMs may not efficiently model and analyse the security of dynamic network since the attack surface from scenarios may be changing.

GSMs for dynamic networks: There are a few studies focusing on developing GSMs for assessing the security of dynamic networks. Frigault et al. [35] proposed a dynamic Bayesian network-based model to capture the evolving vulnerabilities in a network. It is a theoretical framework and they expect it to be used for analysing the changing security aspects of a network. Besides, it is limited to only changes in vulnerabilities as network changes (e.g., changes in topology) are not taken into account. Another similar approach is Bayesian AG (BAG) proposed in [25], however, in BAG, they adopt the idea of Bayesian belief networks and AG to encode different security conditions and the relationship between the various network states and the possibility of exploiting those relationships as well. Almohri et al. [36] presented a success measurement graph model where they analyse the chances of attacks in the presence of uncertainties and further showed how to optimally deploy security and service across the dynamic network. In our previous work [6], we adopted the idea of temporal graphs for the multi-layer HARM in order to model and analyse the security of dynamic networks. The model (T-HARM) captures the temporal security change into layers which thus have scalable properties over other GSMs (as shown in [27]). Here, we used the T-HARM to carry out a comprehensive assessment of security metrics in dynamic networks, to be presented in Section 5.4.

2.2. Security metrics

Many quantitative security metrics for assessing cybersecurity have been developed and formalised (a detailed survey of security metrics can be found in [37,38]). While these metrics remain use-

Table 1
Contribution highlights.

	This work	Previous work			
		[6]	[25]	[35]	[36]
GSM for dynamic networks	**O**	**O**	**O**	**O**	**O**
Countermeasures	**O**	**O**	**O**	X	**O**
Extensive security metrics	**O**	**Δ**	X	X	X
Categorisation of network changes	**O**	X	X	X	X
Formalism of security changes	**O**	X	X	X	X

ful in assessing the security of networks, their effectiveness with respect to dynamic changes in the network is still unclear. Moreover, they did not take into account dynamic changes to the network. Hence, it is difficult to estimate how the existing metrics will change as the network changes over time.

Bopche and Mehtre [1] proposed graph distance metrics for dynamic security analysis, in particular, they used maximum common subgraph and graph edit distance metric to quantify the distance between a pair of successive AGs, and they showed that these metrics could capture the changes in the attack surface of a network. However, these metrics are only used for AGs and thus, it is still unknown how these metrics will change when used for dynamic GSMs and moreover, these metrics cannot leverage the severity value of a vulnerability which is provided Common Vulnerability Scoring System (CVSS) [39] to quantify the impact and risk associated with an attack. In Awan et al. [40], a temporal risk assessment framework is proposed and validated using a dynamic network data. The authors provided a new definition of risk, taking into account the number of threats associated with each attack. In their computation, they introduced time unit t in order to continuously monitor the risk of software applications over time. However, it is still not clear how the existing metrics (with time unit introduced to them) will change when network configuration changes and moreover, the focus of their paper is to find the cyber risk hotspots while our research focused on assessing the security of dynamic networks.

To investigate the varying effects of existing security metrics when changes are observed in the network, we conduct various analysis with different types of changes via the T-HARM. The aim is to assist network/security administrators with the experimental results to determine the security metric to use when a particular type of change occurs in a dynamic network.

3. Preliminaries

This section highlights the contribution of the paper and presents the technical background for the T-HARM and the formulas of the security metrics that were proposed in our previous work [6].

3.1. Contribution highlight

Table 1 compares the work presented in this paper to our previous work and other related work. **O** means it is considered, **Δ** means it is partially considered, and **X** means not considered in the work, respectively. In addition to the contribution highlights, this work extends the experimental analysis from the previous work by considering classes of network changes, a longer period of network change observation time in simulations and addition of a new security metric which is the percentage of severe systems as shown in Section 5.4.

3.2. Dynamic GSM

Dynamic GSMs adjust to changes when changes are observed in the networks in order to effectively represent the new security posture. For instance, the attack surface of a network is frequently changing for a network that supports "the bring your own device policy" [32,41] and as a result, dynamic GSM will need to be adjusted in order to reflect the new security posture of the network.

Dynamic networks consist of a varying set of hosts and edges (i.e., the connections between the hosts) over time. In addition, each of the hosts can have a varying set of vulnerabilities which an attacker can use to compromise the host over time. We assume the attacker can use the reachability between hosts to move from one host to another. T-HARM can be used to capture all the security changes onto two layers: the upper and the lower layer. In the upper layer, the reachability information of the dynamic hosts is captured while in the lower layer, the set of vulnerabilities for each host is captured. Here, we use the definitions of T-HARM proposed in [6] as the dynamic GSM and add the definition of the HARM at a specific time point (which is one snapshot in the T-HARM). We further extend the T-HARM definition by formalizing the dynamic security changes later in Section 4.2.

3.2.1. Definitions of T-HARM

Given t_i (where $i = 1, 2, \dots, n$), a discrete time in a time window T, we define the T-HARM as follows.

Definition 1. T-HARM is a 3-tuple $T - HARM = (S, H, V)$, where $S = \{s_{t_0}, s_{t_1}, \dots, s_{t_n}\}$ is a set of HARM for each network snapshot at time t_i, $H = \{h_0, h_1, \dots\}$ is a set of all hosts in the network, and $V = \{v_0, v_1, \dots\}$ is a set of all vulnerabilities in the network. $H_{t_i} \subseteq H$ is a set of hosts only in the network snapshot at time t_i, and $V_{t_i} \subseteq V$ is a set of vulnerabilities only in the network snapshot at time t_i.

In addition, the HARM for each discrete time t_i is defined as follows.

Definition 2. A HARM at time t_i is a 3-tuple $s_{t_i} = (U_{t_i}, L_{t_i}, C_{t_i})$, where U_{t_i} is the upper layer (dynamic AG) that models the set of hosts H_{t_i}, and L_{t_i} is the lower layer (dynamic ATs) that models the set of vulnerabilities V_{t_i} for each host $h \in H_{t_i}$, respectively. The mapping between the upper and the lower layers is defined as $C_{t_i} \subseteq \{(h_j \leftrightarrow at_{t_i,k})\} \forall h_j \in U_{t_i}, at_{t_i,k} \in L_{t_i}$ (here, $k = 1, 2, \dots$ denotes a distinct attack tree $at_{t_i,k}$ in the lower layer).

We describe the attributes; the set of HARMs S, the set of hosts H, and the set of vulnerabilities V as follows:

- Each HARM $s_{t_i} \in S$ has a set of hosts $s_{t_i}^{hosts} \subseteq H_{t_i}$, a set of vulnerabilities $s_{t_i}^{vuls} \subseteq V_{t_i}$, a set of metrics $s_{t_i}^{metrics} \in$ {attack cost, attack risk, ROI, …} (see Section 3.3 for more details), a topology information $s_{t_i}^{topo} \in \{bus, tree, \dots\}$.
- Each host $h_j \in H_{t_i}$ has a set of adjacent hosts $h_j^{adj} \subseteq H_{t_i}$, a set of vulnerabilities $h_j^{v} \subseteq V_{t_i}$, and a set of security metrics $h_j^{metrics} \in$ {attack cost, attack risk, *ROI*, …}.
- Each vulnerability $v_k \in V_{t_i}$ has a privilege level that is acquired by the attacker after the vulnerability is successfully exploited $v_k^{privilege} \in \{root, user, \dots\}$ and a set of security metrics $v_k^{metrics} \in$ {attack cost, attack risk, …}.

Definition 3. A dynamic AG is a directed graph $U_{t_i} = ag_{t_i} = (H_{t_i}, E_{t_i})$ at time t_i, where H_{t_i} is a finite set of hosts and $E_{t_i} \subseteq H_{t_i} \times H_{t_i}$ is a set of edges. Let Z be a sequence of attacker's snapshots which include one or multiple attackers and $z \in Z$ is a sequence of snapshots $z_{t_0}, z_{t_1}, \dots, z_{t_n}$, where n is the total number of snapshots, $Z \notin S$ and $Z_{hosts} \cap H_{t_i} = \emptyset$. The representation of the upper layer is given by $H_{t_i} \subseteq S \cup Z$ and $E_{t_i} \subseteq (S \cup Z) \times S$.

Definition 4. A dynamic AT is a 5-tuple $at_{t_i,k} = (A_{t_i,k}, B_{t_i,k}, c_{t_i,k}, g_{t_i,k}, root_{t_i,k})$, where $L_{t_i} = \{at_{t_i,1}, at_{t_i,2}, \dots at_{t_i,k}, \dots\}$ at time t_i. Here, $A_{t_i,k} \subseteq V_{t_i}$ is a set of vulnerabilities, $B_{t_i,k} = \{b_{t_i,k}^1, b_{t_i,k}^2, \dots\}$ is a set

Table 2
Notations used in the security metrics computation.

Notation	Metric
Host level	
$ac^h_{t_i}$	is a host attack cost at t_i
$r^h_{t_i}$	is a host attack risk at t_i
$pr^h_{t_i}$	is the probability of attack success of a host at t_i
$aim^h_{t_i}$	is a host attack impact at t_i
$roa^h_{t_i}$	is the return on attack of a host at t_i
Attack path level	
ap	is an attack path which includes a sequence of hosts
$ac^{ap}_{t_i}$	is the attack cost on a path at t_i
$r^{ap}_{t_i}$	is the attack risk on a path at t_i
$pr^{ap}_{t_i}$	is the probability of attack success on a path at t_i
$roa^{ap}_{t_i}$	is the return on attack on a path at t_i
Network level	
AP_{t_i}	is all the possible paths from an attacker to a target for each snapshots at t_i. Each $ap \in AP_{t_i}$
f	is a function that identifies the length of the ap that occurs most frequently
AC_{t_i}	is the cost on attack paths at t_i
R_{t_i}	is the risk on attack paths at t_i
ROA_{t_i}	is return on attack paths at t_i
Pr_{t_i}	is the probability of attack success on paths at t_i
SAP_{t_i}	is shortest attack path at t_i
NAP_{t_i}	is the number of attack paths at t_i
$MAPL_{t_i}$	is the mean of attack path lengths at t_i
$SDPL_{t_i}$	is the standard deviation of attack path lengths at t_i
$MoPL_{t_i}$	is mode of attack path lengths at t_i
$NMPL_{t_i}$	is the normalised mean of attack path lengths at t_i
PSS_{t_i}	is the percentage of severe systems at t_i
NSS_{t_i}	is the number of severe systems at t_i
TNH_{t_i}	is the total number of network hosts at t_i

Table 3
Formulae for the security metrics.

S/N	Metrics	Formulae
1.	R_{t_i}	$r^{ap}_{t_i} = \sum_{h \in ap} pr^h_{t_i} \times aim^h_{t_i}, ap \in AP_{t_i}$ $R_{t_i} = \max_{ap \in AP_{t_i}} r^{ap}_{t_i}$
2.	AC_{t_i}	$ac^{ap}_{t_i} = \sum_{h \in ap} ac^h_{t_i}, ap \in AP_{t_i}$ $AC_{t_i} = \min_{ap \in AP_{t_i}} ac^{ap}_{t_i}$
3.	Pr_{t_i}	$pr^{ap}_{t_i} = \prod_{h \in ap} pr^h_{t_i}, ap \in AP_{t_i}$ $Pr_{t_i} = \max_{ap \in AP_{t_i}} pr^{ap}_{t_i}$
4.	ROA_{t_i}	$roa^{ap}_{t_i} = \sum_{h \in ap} \frac{pr^h_{t_i} \times aim^h_{t_i}}{ac^h_{t_i}}, ap \in AP_{t_i}$ $ROA_{t_i} = \max_{ap \in AP_{t_i}} roa^{ap}_{t_i}$
5.	SAP_{t_i}	$SAP_{t_i} = \min_{ap \in AP_{t_i}} \lvert ap \rvert$
6.	NAP_{t_i}	$NAP_{t_i} = \lvert AP_{t_i} \rvert$
7.	$MAPL_{t_i}$	$MAPL_{t_i} = \frac{\sum_{ap \in AP_{t_i}} \lvert ap \rvert}{NAP_{t_i}}$
8.	$SDPL_{t_i}$	$SDPL_{t_i} = \sqrt{\frac{\sum_{ap \in AP_{t_i}} (\lvert ap \rvert - MAPL)^2}{NAP_{t_i}}}$
9.	$MoPL_{t_i}$	$MoPL_{t_i} = \underset{ap \in AP_{t_i}}{f} (\lvert ap \rvert)$
10.	$NMPL_{t_i}$	$NMPL_{t_i} = \frac{MAPL_{t_i}}{NAP_{t_i}}$
11.	PSS_{t_i}	$PSS_{t_i} = \frac{NSS_{t_i}}{TNH_{t_i}} \times 100$

of gates, $c_{t_i,k} \subseteq \{b^j_{t_i,k} \rightarrow e_l\} \forall b^j_{t_i,k} \in B_{t_i,k}, e_l \in A_{t_i,k} \cup B_{t_i,k}$ is a mapping of gates to vulnerabilities and other gates, $g_{t_i,k} \subseteq \{b^j_{t_i,k} \rightarrow \{AND, OR\}\}$ specifies the type of each gate, and $root_{t_i,k} \in A_{t_i,k} \cup B_{t_i,k}$ is the root node of the $at_{t_i,k}$.

The gates {*AND, OR*} have relationships to vulnerabilities and other gates that establish the connection $c_{t_i,k}$ (the vulnerability of a host are combined using *AND* and *OR* gates). However, structuring the relationships between the above components of the AT (i.e., gates and vulnerabilities) are outside the scope of this paper. The construct of the AT structures can be found in [42–44], which can be used to generate ATs used in this paper.

3.3. Definitions and computations of security metrics

Security metrics and GSMs are used to quantitatively analyse the security of network systems. In the following, we describe the security metrics used in this study along with the formulas to calculate the metrics.

We collect vulnerabilities from the National Vulnerability Database (NVD) [45]), a repository of known vulnerabilities and their standardised severity score. These scores are ranging from 0.0 to 10.0, with 10.0 being the most severe level. We use the vulnerability impact sub score as the attack impact metric for each vulnerability (aim_v) and based on the vulnerability CVSS based score (BS) [45] we assign values to the probability of attack success (pr_v) and attack cost ac_v to the vulnerabilities (e.g., in [6]). We list and describe the notations use in this section in Table 2.

To calculate the metrics, we use T-HARM to find all potential attack paths over time. The potential attack paths *ap* are calculated in the upper layer of T-HARM (the potential attack paths and entry point can change over time). The host level metrics $ac^h_{t_i}$, $r^h_{t_i}$, $pr^h_{t_i}$, $aim^h_{t_i}$ and $roa^h_{t_i}$ are calculated from the lower layer of T-HARM. The metrics NAP_{t_i}, MoPL_{t_i}, SDPL_{t_i}, MAPL_{t_i}, NMPL_{t_i} and SAP_{t_i} [20] are calculated in the upper layer of the T-HARM while we calculate the metrics R_{t_i}, AC_{t_i}, Pr_{t_i}, ROA_{t_i} and PSS_{t_i} using the host level metrics in the upper layer. In this paper, we evaluate eleven main security metrics from [20,38,46–48] and show the metrics and their formulas in Table 3. For each formula, we introduce the attribute time t so that we can use it with the T-HARM.

4. Proposed approach

In this section, we propose a categorisation of network changes based on the causes of configuration changes. Then, we formalise these changes with regards to changes in security models. Further, we perform a comprehensive evaluation of the existing security metrics for use in the security analysis of dynamic networks. We demonstrate a proactive mechanism using the prioritised set of vulnerabilities (PSV) [7,34] and a reactive solution using the isolation of compromised application as a defence mechanism.

4.1. Categorisation of network changes

In networks, there are hosts and edges (i.e., links connecting hosts). For each host, there are applications and operating system (OS) running on it and these components are also with vulnerabilities (i.e., security weaknesses) [49]. The security status of the hosts, edges, applications, OSes and vulnerabilities can be affected by the activities of the users, network administrators and even other events not under the control of the administrators (e.g., software aging, the discovery of a new vulnerability, *etc*). Hence, the network configuration is changing continuously.

Here, since changes in network configuration can change the security posture of networks [1], we identify the causes of network changes and the possible security changes w.r.t. to a GSM, then categorise the network changes into two main categories which are; “change in host” and “change in edges (reachability)” (in Table 4). The possible types of security changes are:

- (1) Addition of host (AN)
- (2) Removal of host (RN)
- (3) Addition of edge (AE)
- (4) Removal of edge (RE)
- (5) Addition of vulnerability (AV) and
- (6) Removal of vulnerability (RV)

Table 4
Categorisation of network changes.

Enterprise network changes		Possible change(s) in GSM
Categories	Subcategories	
Host	Update software	RV, RE, RN
	Uninstall software	RV, RN, RE
	Install software	AV, AN, AE
	Software turn on \enable	AV, AN, AE
	Software turn off \disable	RV, RN, RE
Edges (reachability)	Firewall rules change (e.g., addition new rule)	AE, RE
	Forwarding table change (by SDN)	AE, RE
	Connection of new host	AN, AE, AV
	Disconnection of existing host	RN, RE, RV
	Host turn on	AN, AE, AV
	Host turn off \failure	RN, RE, RV

Further, we correlate their relationships (i.e., the network changes to the security changes) and show them in Table 4. For example, the update of a host software can possibly remove existing vulnerability, remove a connection (i.e., a reachable attack path), or remove a node from the security model (if the node has no vulnerability).

4.2. Formalism of security changes in T-HARM

We formalise the changes in security with respect to network changes for T-HARM as follows:

The addition of a host: The addition of hosts increases the number of host(s) in the network system, which changes both the upper and the lower layers of s_{t_i} in the T-HARM. We define those changes as follows.

Definition 5. The addition of a host h_j changes the HARM $s_{t_i}= (U_{t_i}, L_{t_i}, C_{t_i})$ to $s_{t_{i+1}} = (U_{t_{i+1}}, L_{t_{i+1}}, C_{t_{i+1}})$. The HARM upper layer $U_{t_{i+1}} = (H_{t_{i+1}}, E_{t_{i+1}})$ has a new set of hosts $H_{t_{i+1}} = H_{t_i} \cup \{h_j\}$, and a new set of edges $E_{t_{i+1}} = E_{t_i} \cup \{(h_j, h_x)\} \forall h_x \in h_j^{adj}$. The HARM lower layer changes with a new set of attack trees $L_{t_{i+1}} = L_{t_i} \cup \{at_{t_{i+1},k}\} \mid h_j^v = A_{t_{i+1},k}$, where $at_{t_{i+1},k}$ is a new attack tree that captures the vulnerabilities of h_j. Lastly, the new mapping between the upper and the lower layers of the HARM changes to $C_{t_{i+1}} = C_{t_i} \cup \{(h_j \leftrightarrow at_{t_{i+1},k})\}$.

The removal of a host: The removal of hosts reduces the number of hosts in the network systems. When this happens, the reachability and vulnerabilities associated with the hosts are removed as well. Thus, the upper and lower layers of the T-HARM are changed. We define this as follows.

Definition 6. The removal of a host h_j changes the HARM $s_{t_i}= (U_{t_i}, L_{t_i}, C_{t_i})$ to $s_{t_{i+1}} = (U_{t_{i+1}}, L_{t_{i+1}}, C_{t_{i+1}})$. The HARM upper layer $U_{t_{i+1}} = (H_{t_{i+1}}, E_{t_{i+1}})$ has a new set of hosts $H_{t_{i+1}} = H_{t_i} - \{h_j\}$, and a new set of edges $E_{t_{i+1}} = E_{t_i} - \{(h_j, h_x)\} \forall h_x \in h_j^{adj}$. The HARM lower layer changes with a new set of attack trees $L_{t_{i+1}} = L_{t_i} - \{at_{t_i,k}\}$, where $h_j \leftrightarrow at_{t_i,k}$. Lastly, the new mapping between the upper and the lower layers of the HARM changes to $C_{t_{i+1}} = C_{t_i} - \{(h_j \leftrightarrow at_{t_i,k})\}$.

The addition of a vulnerability: Given a new vulnerability v_x, the AT $at_{t_i,k}$ of a host h_j where $h_j \leftrightarrow at_{t_i,k}$ in C_{t_i} in the T-HARM at time t_i is changed when a new vulnerability is found for that host. The lower layer is changed in the HARM, the list of vulnerabilities for the host is updated, and the set of vulnerabilities at time t_{i+1} is also updated. We define the addition of a vulnerability as follows.

Definition 7. The addition of a vulnerability v_x changes the AT $at_{t_i,k}$ of a host h_j with $h_j \leftrightarrow at_{t_i,k}$ in C_{t_i} and $v_x \in h_j^v \subseteq V_{t_{i+1}}$. This changes $at_{t_i,k}$ to $at_{t_{i+1},k} = (A_{t_{i+1},k}, B_{t_{i+1},k}, c_{t_{i+1},k}, g_{t_{i+1},k}, root_{t_{i+1},k})$, where $A_{t_{i+1},k} = A_{t_i,k} \cup \{v_x\}$ (i.e., v_x is added to the set of vulnerabilities), $B_{t_{i+1},k} = B_{t_i,k} \cup \{b^y_{t_{i+1},k}\} \; \forall b^y_{t_{i+1},k} \rightarrow e_l \mid e_l \in A_{t_{i+1},k} \cup B_{t_{i+1},k}$ (i.e., new gates are added to connect other vulnerabilities and gates with relation to v_x), $c_{t_{i+1},k} = c_{t_i,k} \cup \{b^y_{t_{i+1},k} \rightarrow e_l\}$ where $e_l \in A_{t_{i+1},k} \cup B_{t_{i+1},k}$ and e_l has a relation to v_x. $g_{t_{i+1},k} = g_{t_i,k}$ (i.e., unchanged set of gate types). Finally, $root_{t_{i+1},k} = \in A_{t_{i+1},k} \cup B_{t_{i+1},k}$ (i.e., may have a new root node based on the relationship of v_x). A list of vulnerabilities is updated to $V_{t_{i+1}} = V_{t_i} \cup \{v_x\}$.

The removal of a vulnerability: Various event (e.g., applying security countermeasures, uninstalling a vulnerable application, *etc*) change the lower layer of T-HARM. And as a result, it will need to be updated and we define this change as:

Definition 8. The removal of a vulnerability v_x from a host h_j (where $h_j \leftrightarrow at_{t_i,k}$) will change the AT $at_{t_i,k}$ to $at_{t_{i+1},k} = (A_{t_{i+1},k}, B_{t_{i+1},k}, c_{t_{i+1},k}, g_{t_{i+1},k}, root_{t_{i+1},k})$, where $A_{t_{i+1},k} = A_{t_i,k} - \{v_x\}$, $B_{t_{i+1},k} = B_{t_i,k} - \{b^y_{t_{i+1},k}\}$ (i.e., if v_x is the only vulnerability associated with the gate), $c_{t_{i+1},k} = c_{t_i,k} - \{b^y_{t_{i+1},k} \rightarrow e_l\}$. $g_{t_{i+1},k} = g_{t_i,k}$ (i.e., unchanged set of gate types). Finally, $root_{t_{i+1},k} = root_{t_i,k}$. A list of vulnerabilities is updated to $V_{t_{i+1}} = V_{t_i} - \{v_x\}$.

The addition of an edge: An edge is created when there is a connection between hosts. Hence, the set of edges is updated when a new connection is created, and here, only the upper layer of T-HARM has changed. We define this as:

Definition 9. Given a new connection between hosts h_j and h_m at time t_{i+1}. The HARM s_{t_i} is change to $s_{t_{i+1}} = (U_{t_{i+1}}, L_{t_{i+1}}, C_{t_{i+1}})$, where the upper layer $U_{t_{i+1}} = (H_{t_{i+1}}, E_{t_{i+1}})$ has a new edge such that $E_{t_{i+1}} = E_{t_i} \cup \{(h_j, h_m)\}$, $H_{t_i}{=}H_{t_{i+1}}$. Here, the lower layer $L_{t_{i+1}} = L_{t_i}$ and the mapping between the upper layer to the lower layer $C_{t_{i+1}} = C_{t_i}$ is unchanged.

The removal of an edge: Similarly, an edge is removed from hosts when the reachability between them is disconnected. As a result, only the upper layer of T-HARM is changed accordingly. We define the removal of edges as follows.

Definition 10. Given a connection between hosts h_j and h_m. The HARM s_{t_i} is change to $s_{t_{i+1}} = (U_{t_{i+1}}, L_{t_{i+1}}, C_{t_{i+1}})$, where the upper layer $U_{t_{i+1}} = (H_{t_{i+1}}, E_{t_{i+1}})$ is changed such that $E_{t_{i+1}} = E_{t_i} - \{(h_j, h_m)\}$ and $H_{t_{i+1}} = H_{t_i}$. Here, the lower layer and the mapping between the upper layer to the lower layer is unchanged (i.e., $L_{t_{i+1}} = L_{t_i}$ and $C_{t_{i+1}} = C_{t_i}$), respectively.

5. Evaluations and results

We present the system model and the attacker model in Sections 5.1 and 5.2, respectively. In Section 5.3, we show the construction of the T-HARM for this simulation. In Section 5.4, we present the simulations and results. The focus of the paper is to investigate the effect of security changes on the existing security metrics. Our approach is applicable to any type of dynamic networks.

5.1. System model

We use the network in Fig. 1 as the system model. The network consists of a Demilitarised Zone (DMZ) and an internal network. The DMZ connects to the Internet via the external firewall which is configured to allow direct connections from the Internet only to the DMZ. The internal network is further divided into two subnets by the internal firewall 1 and firewall 2. The internal firewall 1 only allows traffic from the DMZ to have access to ports necessary for the services in the internal network. In the network, the web servers in the DMZ are allowed to access the Internet while

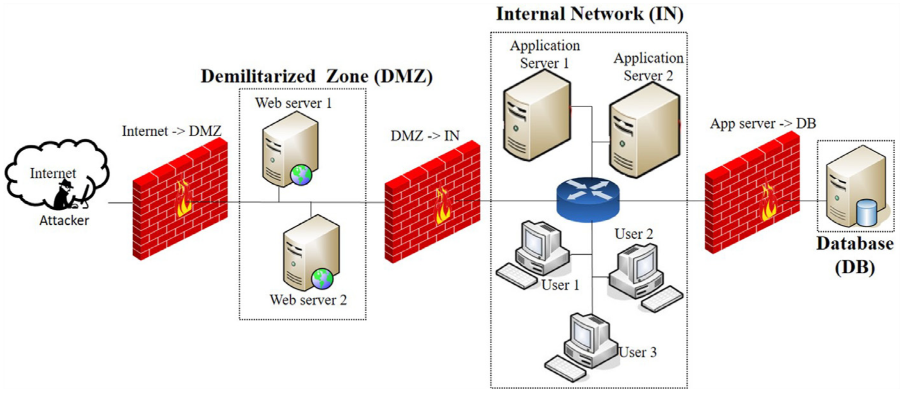


Fig. 1. Configuration of the network.

Table 5
OSes and applications running on hosts.

Host	OS	Service
WS_1	Redhat Enterprise Linux 6	Apache http server 2.4
WS_2	Redhat Enterprise Linux 6	Apache http server 2.4
AS_1	Windows 10	WebLogic server 12.1
AS_2	Redhat Enterprise Linux 6	Apache tomcat 7.0
DB	Windows 10	Oracle database 11g
$User_i$	Redhat Enterprise Linux 6	Mozilla firefox 31.1.0

the hosts in the DMZ are allowed access to the application servers in the internal network then to the database server. In specific, a remote user cannot have direct access to the database server. Only the application servers in the internal network can request data from the database server. Further, there are user workstations in the internal network (the number of hosts can vary for our simulation scenarios). In Fig. 1, we use the symbol '$- >$' to indicate the connections from a host or subnet to another (e.g., A can connect $(- >)$ to B). Table 5 shows the OSes and services running on the servers and workstations. In the Table, we denote web server as *WS*, application server as *AS*, Database server as *DB* and a user workstation as $User_i$ (where $i = 1, 2, \ldots, n$).

5.2. Attacker models

Users from the Internet can have access to services in the network. An attacker can easily access network hosts and compromise them remotely. We consider a targeted attack scenario in which an attacker is interested in the DB containing sensitive information. The entry points of the attacker are the web servers. We assume there is only one attacker and the attacker is located outside of the network (for example, an outside attacker can come from the Internet or a partner network linked to the enterprise network such as customers or vendors). In this model, the attack goal is to escalate privileges within a low privilege account and gain access to administrator rights and further steal sensitive information.

The DB is running the Oracle database that has a forever day vulnerability in its current version. The Common Vulnerabilities and Exposures (CVE) [45] ID for the vulnerability is CVE-2012-1675. It is often unable to patch the "forever day" vulnerability as the vendor is unlikely to provide the patch (for example, the service is no longer available or the vendors no longer support that product) and this vulnerability can be remotely exploited without authentication (i.e., without the need for a username and password) thus allowing the attacker to gain the root privilege with full control of the DB. We assume the attacker cannot compromise the target host directly. However, once the attacker successfully compromises a host, the attacker can gain the root privilege of the host and subsequently attack the next host in the network until the target host is reached. Besides, the attacker must combine the OS and the Oracle database vulnerability in order to have full control of the target host (i.e., the DB).

5.3. Construction of T-HARM

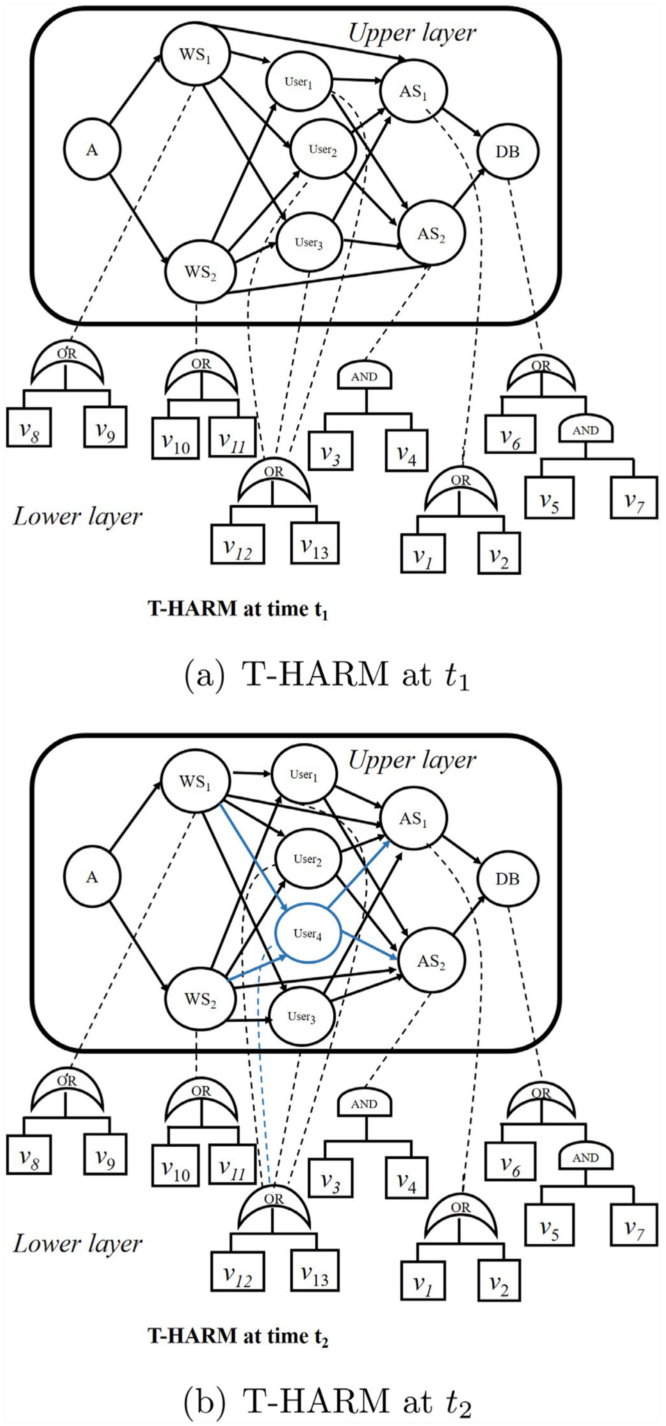
We use the T-HARM to captures network changes at different times. We assume a flexible time window, so this is adjusted to different time and interval as desired (e.g., 1 week or 1 month) thus providing different views to evaluate the security metrics over time. We show an example T-HARM in Fig. 2 for the network shown in Fig. 1 when a new workstation is connected to the network (i.e., when a change is detected). In the model, the attacker *A* is able to reach the DB via DMZ and IN subnet. He is able to compromise the vulnerabilities for the hosts in the DMZ subnet and also for the hosts in the IN subnet and finally gain the privilege of the DB.

5.4. Simulations and results

We conduct experimental analysis via simulations using a T-HARM for the network changes described in Table 4. To begin with, we map the network changes w.r.t. to security changes in Table 6 for our simulations.

Simulation Settings: We use the enterprise network in Fig. 1 as the initial network state to conduct different experiments via simulations. For the subsequent states, we introduce various network changes (the changes are listed in Table 6) and we will describe each of them in the various scenarios that they are used. In the simulations, we assume the dynamic host configuration protocol automatically assign IP address settings to hosts that are joining the network. Also, users can install software on this hosts and the software have one or more vulnerability. Among other activities, we assume the network administrators' tasks include: administering network security (e.g., disabling vulnerable hosts, software, *etc.*) and changing of firewall rules. We must note that, in the simulation networks, we ensure that there is always at-least a type of server that is up and running at any time *t* in order to guarantee the access of client to the DB (for example, we ensure that only one of the two web servers (WS_1 and WS_2) is disconnected or removed from the network per time). For the vulnerabilities use in the simulations, we collect them from a repository NVD). In real networks, such information is collected using scanners (such as Nmap [50], OpenVAS [51], *etc.*).

Next, we construct the T-HARM for the dynamic network using the aforementioned inputs and the attacker model specified in Section 5.2. Then, we use it to calculate several security metrics



(a) T-HARM at t_1

(b) T-HARM at t_2

Fig. 2. The T-HARM for the simulation network (i.e., when a change is detected) with T = 2.

in the various scenarios, respectively. Our simulations focus on investigating the varying effects of security metrics when changes are observed in the network for the following scenarios (such that all the network changes specified in Table 4 are captured and analysed): (1) Addition of software vulnerabilities, (2) Addition of new hosts (hosts having vulnerabilities), (3) Software Update (e.g., patching of vulnerabilities), (4) Disabling application software, (5) Installation of new application software, (6) Removal of existing hosts, and (7) Change of firewall rules.

Table 6
Simulations: security changes w.r.t. network changes.

Changes in security	Causes of configurations change	Related subsection(s)
Addition of vulnerabilities	(1) installation of software	5.4.5
	(2) software turned on or enabled	5.4.5
	(3) addition of new hosts	5.4.1 (1)
	(4) discovery of vulnerabilities	5.4.1 (2)
Removal of vulnerabilities	(1) update of software	5.4.2
	(2) uninstallation of software	5.4.4
	(3) software turned off or disabled	5.4.4
	(4) removal of hosts	5.4.1 (3)
Addition of nodes	(1) addition of hosts	5.4.1 (1)
	(2) installation of software	5.4.5
	(3) software turned on or enabled	5.4.5
	(4) host turned on	5.4.1 (1)
Removal of nodes	(1) removal of hosts	5.4.1 (3)
	(2) uninstallation of software	5.4.4, 5.4.2
	(3) update of software	5.4.2
	(4) software turned off or disabled	5.4.4
Addition of edges	(1) addition of host	5.4.1 (1)
	(2) host turn on	5.4.1 (1)
	(3) connection of host	5.4.1 (4)
	(4) firewall rules changed	5.4.1 (4)
	(5) installation of software	5.4.5
Removal of edges	(1) removal of host	5.4.1 (1)
	(2) host turned off or failed	5.4.1 (3)
	(3) disconnection of host	5.4.1 (3 and 4)
	(4) firewall rule changed	5.4.1 (4)
	(5) software turned off or disabled	5.4.4

Simulation results: We use line graphs to present the simulation results. In the graphs, we present the change of time t on the horizontal axis and the normalised metric value on the vertical axis. In particular, we use the metrics summarised in Table 3. We use time with each metric in order to capture a metric value per time (for instance, the metric R at time t_i is represented as R_{t_i}). In our computation, we normalise all the metrics values (i.e., ranging from 0.0 to 1.0) and we also consider that the attacker can find multiple attack paths to reach the target at any time.

5.4.1. Summary of previous results

In this section, we summarise the simulation results for the following scenarios (in relation to Table 4): addition of new hosts, the discovery of vulnerabilities, removal of hosts and changes of firewall rules as the detailed discussion of these results were presented in [7]. Moreover, in [7], a network model and attacker model similar to Sections 5.1 and 5.2 were used, respectively. Besides, we also perform simulations for these changes w.r.t. the new security metric *PSS* which was not considered in the previous work. We summarise the results as follows.

1. *Addition of new hosts*: an incremental addition of hosts to a network is considered, where the added hosts have similar vulnerabilities to the existing hosts in the internal network. The results showed that MePL_{t_i}, MoPL_{t_i}, SDPL_{t_i}, NAP_{t_i} and NMPL_{t_i} change continuously. On the other hand, R_{t_i}, AC_{t_i} and ROA_{t_i} only show limited change and the SAP_{t_i} and Pr_{t_i} do not change for all the time.
2. *Discovery of vulnerabilities*: the discovery of vulnerabilities over 12-months period was considered. The results showed that R_{t_i}, AC_{t_i}, Pr_{t_i}, ROA_{t_i} change in their values when new vulnerabilities are discovered for some of the time (which is indicating the changes in security accordingly). However, as the number of vulnerabilities found on each host becomes large, the values of the metrics become static for the remaining time. On the other hand, the MePL_{t_i}, MoPL_{t_i}, SDPL_{t_i}, SAP_{t_i}, NAP_{t_i} and NMPL_{t_i} remain static for all the time.
3. *Removal of hosts*: we considered the removal of hosts from a network (e.g., when an existing host disconnects from the net-

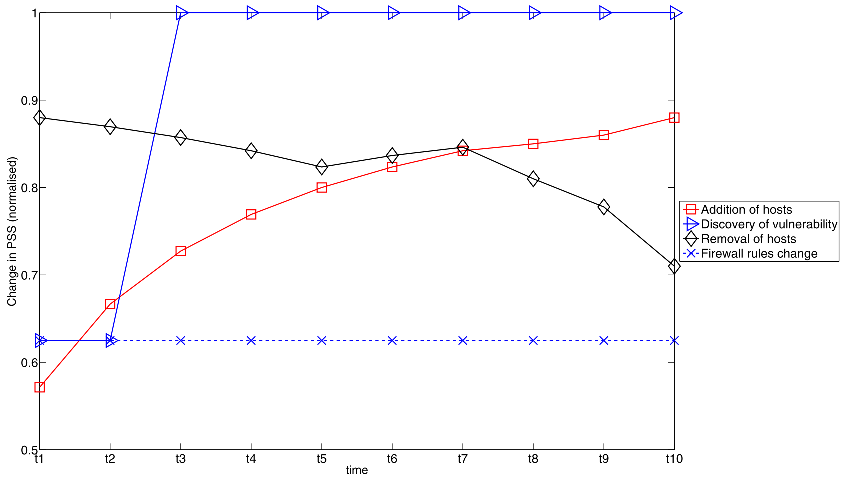


Fig. 3. Changes in PSS_{t_i} w.r.t. addition of hosts, discovery of vulnerability, removal of hosts and changes in firewall rules.

work). The results showed that MePL_t, MoPL_t, SDPL_{t_i}, NAP_{t_i} and NMPL_{t_i} change in their values when a host is removed. The R_{t_i}, AC_{t_i}, Pr_{t_i} and ROA_{t_i} on the other hand only begin to change when the number of hosts removed becomes large and when well-connected hosts (e.g., servers) are disconnected from the network.

4. *Change of firewall rules*: we reasonably modify existing firewall rules to observe how the security metrics are affected. The MePL_{t_i}, MoPL_{t_i}, SDPL_{t_i}, NAP_{t_i} and NMPL_{t_i} change accordingly for each change. While the R_{t_i}, AC_{t_i} and ROA_{t_i} show small changes and SAP_{t_i} and Pr_{t_i} do not change for the entire time window.

We present the results for the additions of hosts, the discovery of vulnerabilities, removal of hosts and changes in firewall rules over time w.r.t. the *PSS* metric in Fig. 3. The results show that for the addition and removal of hosts, the PSS_{t_i} metric changes continuously; for the discovery of the vulnerabilities, the metric only changes after t_2 and however becomes static for the remaining time afterward. This is because, at the point t_2, all the network hosts are having severe vulnerabilities which makes the total rate of severe systems to 100% (so the metric is not changing again). On the other hand, the *PSS* metric does not change when the firewall rules change. This is for the reason that changing the firewall rules only affects the hosts' connections but does not affect the number of critical vulnerabilities that are found on the network hosts.

5.4.2. Scenario I: software update

Software vendors tend to actively release updates for their software products and then leave it to the customers to decide whether to apply those software corrections or not [52]. The implementation of these updates removes one or more vulnerabilities from the software and hence, for a networked system (where a cyber-attacker uses this weakness to bypass security measures), the removal of these set of vulnerabilities changes the security posture and this requires that the network should be re-assessed in order to determine the new security. At this point, it is our interest to investigate what security metrics are changing when software is updated over time.

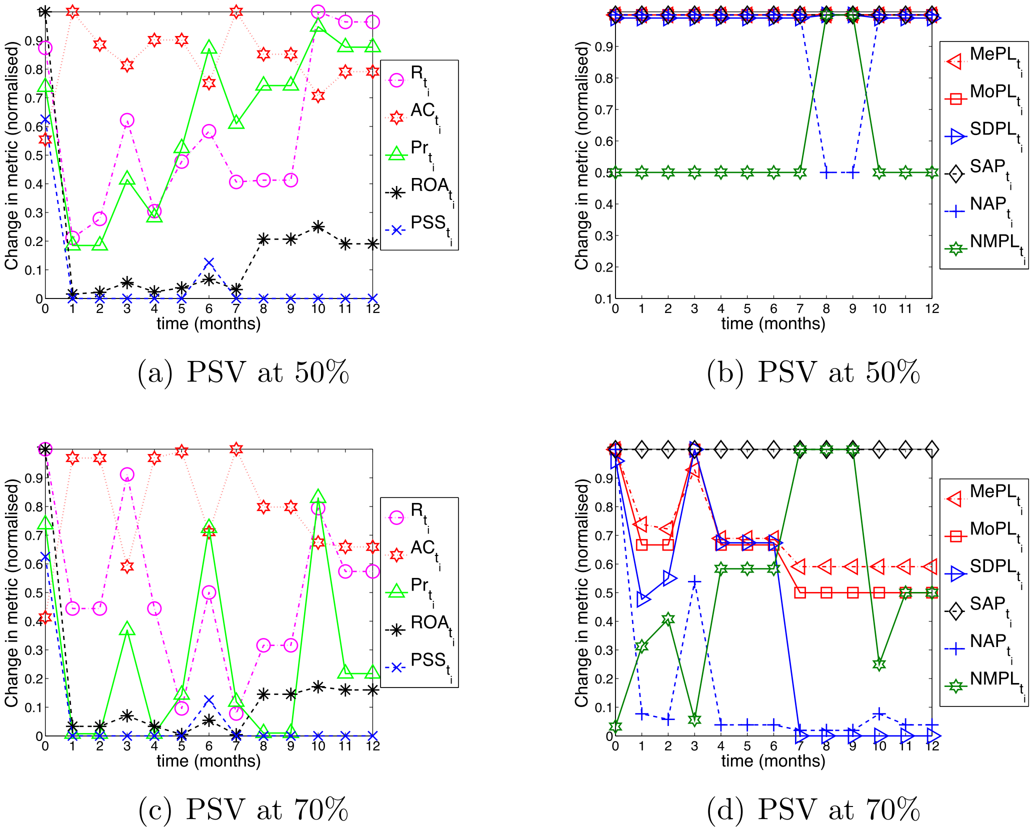
Here, we made the following assumptions: (i) that updates are regularly performed once per month and (ii) that all updates are tested first in a pre-production environment before applying to the network devices (hence, there is no failure during update process). Further, we assume that we know all the enterprise network vulnerability information for twelve months (this is because we collected vulnerability data for 12 months (i.e., from April, 2015 to March, 2016) from the NVD). Additionally, since the organisation cannot afford to remove all the vulnerabilities due to cost, time, unavailability of updates and other security policies, we use the risk based PSV using a hybrid method to determine what vulnerability to remove first. We perform simulation using different PSV values (i.e., 50% and 70%), and use only a PSV value for an entire time window. We plot the results in Fig. 4. In particular, we plot 50% and 70% in Fig. 4(a) and (b), (c) and (d), respectively.

In Fig. 4, we use the PSV value of 50% and 70%. The results show that, increasing the PSV value to 50% do not only changes the set of security metrics in Fig. 4(a) but NAP_{t_i} and NMPL_{t_i} as well (in Fig. 4(b)). However, the NAP_{t_i} and NMPL_{t_i} only show small change (i.e., between month 8 to month 9) while the other once in Fig. 4(b) remain static. When we increase the PSV value to 70% (in Fig. 4(c) and (d)), we observe that all metrics show a significant change in their values for all the months except the SAP_{t_i}. The SAP_{t_i} did not change because the minimum number of hosts along the attack path for the attacker to reach the target remain the same as the initial network. This implies that when the values of the PSV are high, all the security metrics will show a significant change in their values except for the SAP_{t_i}.

Further, we develop an algorithm based on *NAP* which selects PSV values that improve the security of networks over time (i.e., by selecting PSV that reduces the total number of attack paths). The algorithm selects different PSV value for each network state based on the number of vulnerabilities that are found (such that the security is always improved based on the calculation of *NAP* metric). Any other security metric can be used with the algorithm. Our aim is to observe how the security metrics are changing for this case. As a result, we perform simulations and use PSV = 60%, 60%, 80%, 60%, 80%, 90%, 30%, 30%, 70%, 90%, 80% and 80% for month 1 through month 12, respectively (which is determine by Algorithm 1). In the algorithm, we use the following notations:

- *PSV*: prioritise set of vulnerabilities
- $s_{t_i}^{psv}$: a PSV value for s_{t_i}
- *SW*: set of PSV value for a time window ($s_{t_i}^{psv} \in PV$)
- $s_{t_i}^{metric}$: a calculated security metric for s_{t_i}

The Algorithm 1 calculates the PSV value to use for each network state. First, the initial network state is collected (line 2). Then for each other network state (line 3), we calculate a metric of interest (here, we use NAP) for each s_{t_i} (line 4 and 5). Further, we calculate the PSV (line 7 and 8) and remove a vulnerability at a time (line 8 and 9). For each of the vulnerability (v) removed, the metric ($s_{t_i}^{NAP}$) is calculated (line 10) until the calculated $s_{t_{i+1}}^{NAP}$ is less



(a) PSV at 50%

(b) PSV at 50%

(c) PSV at 70%

(d) PSV at 70%

Fig. 4. Change w.r.t. software update.

Algorithm 1: Selecting PSV based on the NAP.

```
   Data: $S$
   Result: $SW$
1  begin
2      $s_{t_0} \leftarrow$ initial network
3      for $s_{t_i} \in \{s_{t_1}, s_{t_2}, \ldots, s_{t_n}\}$ do
4          NAP $= |AP|$
5          $s^{NAP}_{t_i} \leftarrow$ NAP
6          for $V \in s_{t_i}$ do
7              $s^{psv}_{t_i} \leftarrow PSV$
8              for each patchable $v \in s^{psv}_{t_i}$ do
9                  patch $v$
10                 $s^{NAP}_{t_{i+1}} \leftarrow NAP$
11                 if $s^{NAP}_{t_{i+1}} < s^{NAP}_{t_i}$ and $s^{NAP}_{t_{i+1}} <= s^{NAP}_{t_{i-1}}$ then
12                     append $|s^{psv}_{t_i}|$ to $SW$
13                 end
14             end
15         end
16     end
17 end
```

than the previous network state ($s^{NAP}_{t_{i-1}}$) in S (line 11 and 12) in the set of solutions in SW.

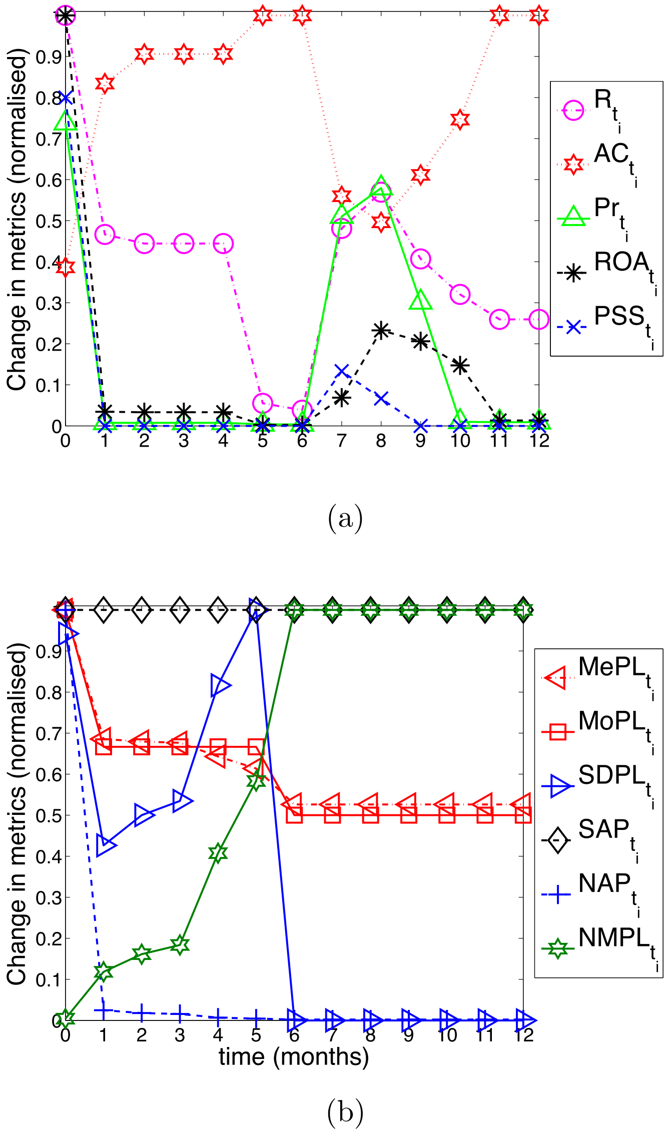
We plot the results in Fig. 5. From the results, the NAP$_{t_i}$, MAPL$_{t_i}$, MoPL$_{t_i}$, SDPL$_{t_i}$ and NMPL$_{t_i}$ change progressively and until 'month 6' when there is only an attack path to reach the target host. This is because we use a 'forever day' vulnerability which is not removed by software update hence, there is always an attack path to reach the target host. Additionally, we observe that the SAP$_{t_i}$ remains static for all the time (this is because the minimum number of hosts along the attack path for the attacker to reach the target remain the same as the initial network.). On the other hand, the R$_{t_i}$, AC$_{t_i}$, Pr$_{t_i}$, ROA$_{t_i}$, SDPL$_{t_i}$ and PSS$_{t_i}$ change progressively as the number of paths are progressively reduced over time. However, at R$_{t_7}$, AC$_{t_7}$, Pr$_{t_7}$, ROA$_{t_7}$ and PSS$_{t_7}$, the metric start deteriorating but later improve again. Our manual analysis revealed that even when there is only an attack path to reach the target host, the R$_{t_7}$, AC$_{t_7}$, Pr$_{t_7}$, ROA$_{t_7}$ and PSS$_{t_7}$ increases (i.e., from month 5) because the risk associated to the hosts (e.g., the CVSS BS) along the attack paths are high during that period, hence the deterioration in the security from month 7 to month 10 in Fig. 5(b).

This implies that improving the security of networks based on PSV can improve network security (as seen in Fig. 5(b)). However, improving the security based on a single metric (e.g., the NAP) may not improve the values of other security metrics (as seen in Fig. 5(a) compared to Fig. 5(b)).

In summary, the results showed that all the metrics change significantly in their value (except SAP$_{t_i}$) when the number of vulnerabilities removed is high.

5.4.3. Scenario II: addition and removal of connections

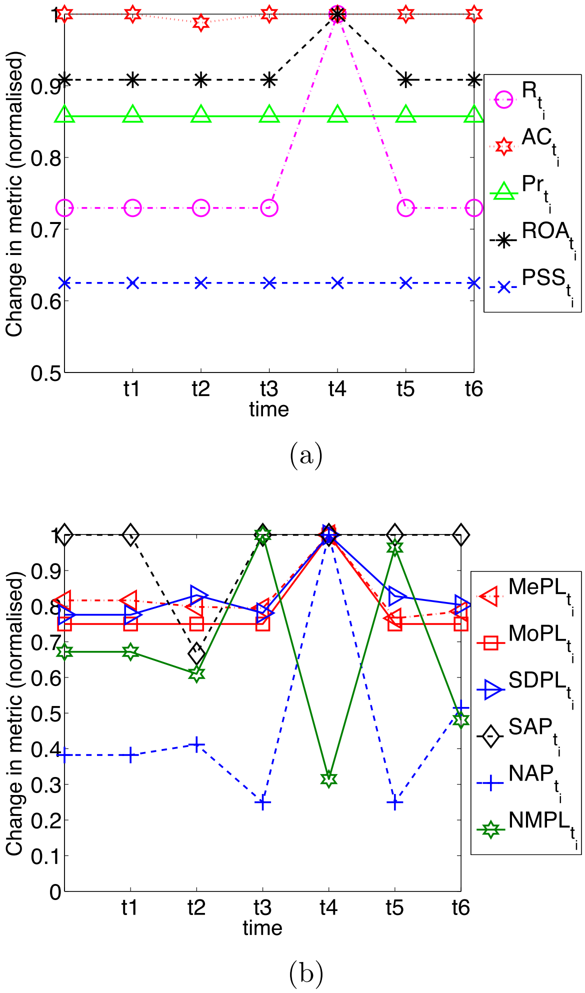
Modern networks allow it topology to change by allowing hosts to randomly connect/disconnect to or from each other. For example, users are allowed to access different resources (database, domain name server, *etc*) which in consequence create connections to the new resource or remove from the previous resource. We simulate this type of networking changes by modifying the existing network connections (this is also similar to firewall rules change). We observe the changes in the security metrics and plot the results in Fig. 6.



(a)

(b)

Fig. 5. Change w.r.t. emergence and patching of vulnerabilities with PSV = 60%, 60%, 80%, 60%, 80%, 90%, 30%, 30%, 70%, 90%, 80% and 80% for month 1 through month 12, respectively.



(a)

(b)

Fig. 6. Addition and removal of edges.

We observed that Pr_{t_i} and PSS_{t_i} remain static for all the different changes that we introduced. PSS_{t_i} and Pr_{t_i} do not change because the network vulnerability information does not change from the initial network configurations. R_{t_i} and ROA_{t_i} only show a change at t_4 when a new connection is added, which as a result create another attack path that happens to be having the most critical hosts along the path. Thus causing the significant change in R_{t_i} and ROA_{t_i} for that time. Conversely, in Fig. 6(b), all the metrics show significant changes w.r.t. to the different security changes (accordingly) except the *SAP*.

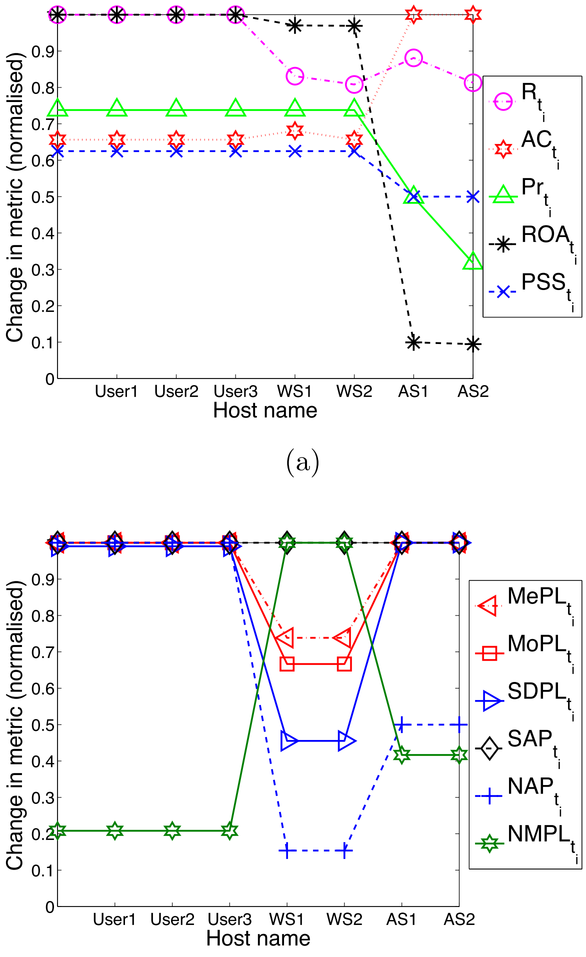
In summary, the MePL_{t_i}, MoPL_{t_i}, SDPL_{t_i}, SAP_{t_i}, NAP_{t_i} and NMPL_{t_i} change accordingly for this network change. R_{t_i}, AC_{t_i} and ROA_{t_i} only show small change and PSS_{t_i} and Pr_{t_i} do not change for the entire time window.

5.4.4. Scenario III: disabling application software

An unattained vulnerability (e.g., vulnerability in a software that the vendor are no longer supporting it) can lead to unpredictable outcome and as such, security administrators need to mitigate the impact of the unattained vulnerabilities by disabling or turning off the service that is associated with the vulnerability [49] in order to avoid exposure to different type of cyber-attack. One way to defend against this type of vulnerability is to disable the application software on the host machine. In this section, we investigate how the action of turning off a vulnerable application on a host will change the various security metrics in a network system. We use the system settings and configurations from Section 5 for this simulation and further, we assume that there is always one host that is having a non-patchable vulnerability per time. Hence, we disable only the software for that host per time in the simulations. To ensure users access to the database, we did not disable the application on the DB. We plot the results in Fig. 7.

In Fig. 7(a) and (b), the results show that disabling a vulnerable application on the user workstations (i.e., $User_1$, $User_2$ and $User_3$) does not change any of the security metrics whereas, when we disable the applications on WS_1, WS_2, AS_1 and AS_2, the NAP_{t_i}, MAPL_{t_i}, MoPL_{t_i}, SDPL_{t_i} and NMPL_{t_i} change accordingly. Similarly, the R_{t_i}, AC_{t_i}, Pr_{t_i}, ROA_{t_i}, and PSS_{t_i} change as a result of that as well. However, Pr_{t_i} and PSS_{t_i} do not change when the vulnerable application on WS_1 and WS_2 is disabled. On the other hand, SAP_{t_i} did not change for all the time. In our observations, we found that disabling the vulnerable applications on the servers affected their availability, hence the significant (sharp) change in the metrics values for most of the time that a server application is disabled. The results of the simulations presented in Fig. 7.

In summary, we observe that disabling a vulnerable application on a server compared to a user workstation can change the



(a)

(b)

Fig. 7. Disabling a vulnerable application on a host.

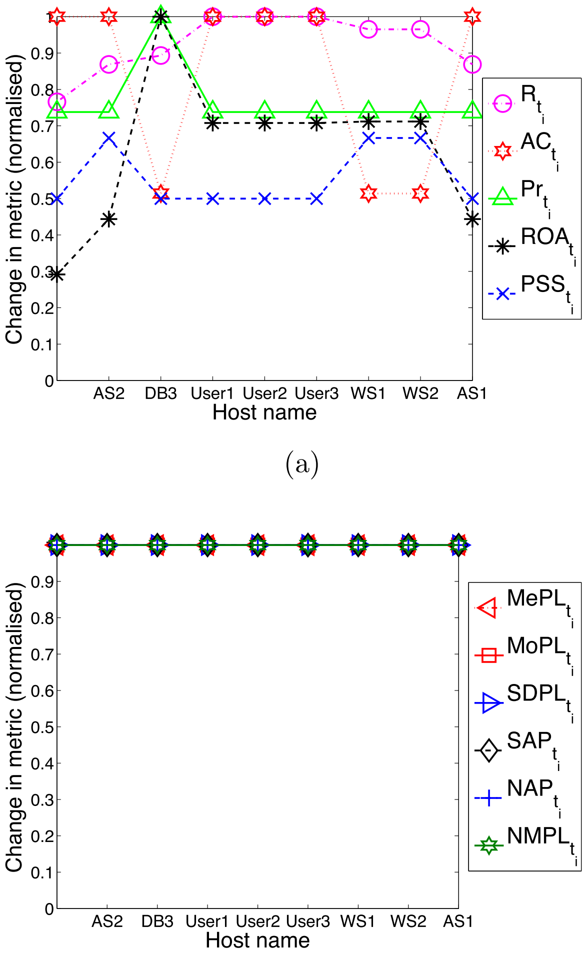
Table 7
List of vulnerabilities and metrics use for Google Chrome.

CVE-ID	CVSS BS	*pr*	*aim*	*ac*
CVE-2015-6790	10.00	1.00	10.00	0.10
CVE-2015-8664	7.50	0.75	6.4	2.50

security metrics more significantly thus the security is improved greatly. The reason is that all attack paths to reach the target hosts connect to most of the network servers and therefore, when a security measure is applied on any server, most attack paths will be affected as well.

5.4.5. Scenario IV : installation of new application

An application is usually installed to improve user experiences but these applications are not without vulnerabilities [49] which may increase the attack surface of a networked system and consequently, change the security posture. We investigate how security metrics are changing when a new application is installed for the network described in Section 5. Here, we choose one of the commonly used application (e.g., Google Chrome) then we add this application to the hosts, one per time. Next, we collect vulnerabilities associated with the application from the NVD in order to use for our simulation network. We list the vulnerabilities in Table 7 and assign values to the probability of attack success (*pr*), attack impact (*aim*) and attack cost (*ac*) to each of the Chrome vulnerability and then perform dynamic security analysis.



(a)

(b)

Fig. 8. Installing an application on the hosts.

We plot the results in Fig. 8. We observe that the metrics - R_{t_i}, AC_{t_i}, ROA_{t_i} and PSS_{t_i} deteriorate in their values for each time that a new application is installed on a host (this is because the installation of the application increases the risk value of hosts). Contrarily, $MePL_{t_i}$, $MoPL_{t_i}$, $SDPL_{t_i}$, SAP_{t_i}, NAP_{t_i} and $NMPL_{t_i}$ do not change for all the time for the reason that all the hosts in the network are regularly having other vulnerabilities. Consequently, the attack paths to the target host do not change for the entire time that a new application is installed. From the results of the simulations, we notice that installing another vulnerability application on the DB affects the R_{t_i}, AC_{t_i}, ROA_{t_i}, Pr_{t_i} and PSS_{t_i} significantly (the security decrease a lot) compared to any other hosts found in the network (with a high chance for the attacker to attack the target hosts). Similarly, the AC_{t_i} reduces when an application is installed on WS_1 and WS_2 demonstrating that a less effort is required by the attacker to reach the target hosts.

The results in this section showed that the installation of a new application on a host does not change $MePL_{t_i}$, $MoPL_{t_i}$, $SDPL_{t_i}$, SAP_{t_i}, NAP_{t_i} and $NMPL_{t_i}$. The reason is that there is already an exploitable vulnerability on each of the network hosts (in the upper layer) and thus, all the possible attack paths to reach the target host remain the same compared to the earlier network state. This is consistent with all the scenarios in which this sets of metrics do not change.

Table 8
Effects of security metrics w.r.t. changes in the network.

Security metrics	Possible network changes						
	Discovery of vulnerabilities	Addition of hosts	Update of software	Disabling application	Installing application	Removal of hosts	Add/remove edges or firewall rules
R_{t_i}	√	†	√	†	√	†	†
AC_{t_i}	√	†	√	†	√	†	†
Pr_{t_i}	√	✗	√	†	†	†	†
ROA_{t_i}	√	†	√	†	√	†	†
PSS_{t_i}	†	√	†	†	√	√	✗
SAP_{t_i}	✗	✗	✗	✗	✗	✗	✗
NAP_{t_i}	✗	√	√	†	✗	√	√
MAPL_{t_i}	✗	√	√	†	✗	√	√
NMPL_{t_i}	✗	√	√	†	✗	√	√
SDPL_{t_i}	✗	√	√	†	✗	√	√
MoPL_{t_i}	✗	✗	√	†	✗	√	√

In summary, the results show that the MePL_{t_i}, MoPL_{t_i}, SDPL_{t_i}, SAP_{t_i}, NAP_{t_i} and NMPL_{t} do not change while the others change appropriately w.r.t. to the installation of a new application.

5.4.6. Summary

The existing security metrics respond to changes in different ways when we investigate the metrics during a period of time. We found that, depending on the types of security changes the different security metrics can show changes in their values when there is a change in the network system. However, none of the security metrics change for all the network changes that we have observed. In Table 8, we summarise the results. We use the symbol √, † and ✗ to indicates a metric that shows significant change, small change and no change, respectively over time. Here, we refer to “significant change” as the metric that changes in it’s value when the configuration of the network change. While “small change” is the metric that change for only a few time and finally, we refer to a metric that did not change for all the time as “no change”.

6. Limitations and extensions

In this paper, we investigate the varying effects of security metrics when changes are observed in the network. We conduct comprehensive analysis with the different categories of network changes via T-HARM. However, the more dynamic analysis should be made in the following areas:

Different network characteristics: Although we considered typical enterprise network configurations for our simulation but generally, a network can be complex with different types of network topologies or a combination of different topologies (e.g., a combination of star and ring). In this paper, we did not consider different types of network topologies and network density and how the various characteristics affect security metrics in a dynamic network. Thus, it is important to consider how the different topologies and network density can affect the security of dynamic networks.

Furthermore, we did not consider other network technologies such as Cloud and IoT in the categorisation of the network changes. Hence, we will perform a more detailed categorisation of modern network changes (for Cloud, IoT, *etc.*) with their respective correlation to the changes in GSMs in order to perform more analysis.

Dynamic models: We used a time-sensitive scalable GSM (named T-HARM) that takes snapshots of the dynamic network at every period *t* (we captured the different state of the network) and then dynamically analysed the security. In spite of this, modern networks (e.g., Cloud and SDN) usually allow their components to change even more frequently (than enterprise networks) and in consequence, an important network state (for effective security analysis) can be skipped. Therefore, in our future work, we will consider an approach that can take into account all the network states.

Attacker models: In our attacker model, we considered a single target host and a single attacker. We can model multiple attackers trying to compromise different targets. We also did not consider an internal attacker who wants to compromise hosts in the internal network. Therefore, more research is needed to consider attacker models with multiple attackers, internal attackers and multiple targets, as well as considering various attack scenarios (e.g., Distributed Denial of Service attack [53]).

Optimal defence models: We considered software update and the disabling of a vulnerable application software as defence mechanisms. However, an optimal solution for different network snapshots varies but we did not compute their optimal solutions. Moreover, we did not assess the effectiveness of the defence model given multiple metrics. In our future work, it is our interest to improve the security of dynamic networks hence, we will find an approach that computes optimal security for the dynamic network.

Security metrics: We investigated the varying effects of eleven (11) security metrics when various changes are observed in the network. However, there are many other quantitative security metrics for assessing the security of networks (e.g., weakest adversary metrics [54], network compromise percentage [55], attack resistant metric [22], K-zero day safety [56], attack surface metric [57], *etc.*). So, more comprehensive evaluation of those security metrics for assessing the security of dynamic networks is required. In addition, Bopche and Mehtre [1] proposed graph distance metrics for dynamic security analysis. In particular, they used maximum common subgraph and graph edit distance metric to quantify the distance between a pair of successive AGs generated for a dynamic network. Their results showed that these metrics can capture the temporal changes in a network attack surface. But, these metrics are used with AGs and it is still unknown how they will change when they are used for dynamic GSMs.

Validation: Although we have modelled changes in vulnerabilities from a real-world network setting, one of the limitations of our work is that we did not extend it to a real test-bed network for validation. We can collect experimental data from a real enterprise network settings and configuration (e.g., how frequent hosts are added, removed, *etc.*) for a period of time, then use the data in our simulations in order to validate the findings. Therefore, we will validate the results of the simulations in a real test-bed network in our future work.

7. Conclusion

Assessing the security of dynamic networks is difficult as the network and security components change over time. Moreover, it is difficult to estimate how existing security metrics will change as the network changes over time, since current metrics have only been used to assess static networks.

In this paper, we have systematically categorised and formalised various network changes based on the causes of the change. Further, we have assessed the security of dynamic enterprise networks via T-HARM with the following changes: (1) Addition of software vulnerabilities (2) Update of vulnerabilities (3) Installation of new application software (4) Disabling of application software (5) Addition of new hosts (hosts having vulnerabilities) (6) Removal of existing hosts (7) Addition and removal of connections.

In addition, we have introduced time to eleven existing security metrics and investigated the effect of these changes on the metrics. Finally, we have summarised the effectiveness of each of the metrics according to their security changes. From the results, a security/network administrator is able to determine the security metric that will effectively present the security posture of a network when a certain type of network configuration change occurs. This work provides a foundation for further research in this area.

Acknowledgment

This work was made possible by the support of a grant (NPRP 8-531-1-111) from the Qatar National Research Fund (QNRF). The statements made herein are solely the responsibility of the authors.

References

[1] G.S. Bopche, B.M. Mehtre, Graph similarity metrics for assessing temporal changes in attack surface of dynamic networks, Comput. Secur. 64 (C) (2017) 16–43, doi:10.1016/j.cose.2016.09.010.
[2] M. Albanese, S. Jajodia, S. Noel, Time-efficient and cost-effective network hardening using attack graphs, in: Proceedings of the 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012), IEEE Computer Society, Washington, DC, USA, 2012.
[3] X. Ou, W.F. Boyer, M.A. McQueen, A scalable approach to attack graph generation, in: Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006), ACM, 2006, pp. 336–345.
[4] A. Roy, D.S. Kim, K.S. Trivedi, Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees, in: 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012), 2012, doi:10.1109/DSN.2012.6263940.
[5] V. Saini, Q. Duan, V. Paruchuri, Threat modeling using attack trees, J. Comput. Sci. Coll. 23 (4) (2008) 124–131.
[6] S.Y. Enoch, M. Ge, J.B. Hong, H.K. Kim, P. Kim, D.S. Kim, Security modelling and analysis of dynamic enterprise networks, in: Proceedings of the 16th IEEE International Conference on Computer and Information Technology, (CIT 2016), IEEE, 2016, pp. 249–256, doi:10.1109/CIT.2016.88.
[7] S.Y. Enoch, M. Ge, J.B. Hong, H. Alzaid, D.S. Kim, Evaluating the effectiveness of security metrics for dynamic networks, in: Proceedings of the 16th IEEE International Conference On Trust, Security and Privacy in Computing and Communications (TrustCom 2017), IEEE, 2017, pp. 277–284, doi:10.1109/Trustcom/BigDataSE/ICESS.2017.248.
[8] B. Kordy, L. Piètre-Cambacédès, P. Schweitzer, DAG-based attack and defense modeling: Don't miss the forest for the attack trees, Comput. Sci. Rev. 13 (2014) 1–38.
[9] T. Tidwell, R. Larson, K. Fitch, J. Hale, Modeling internet attacks, in: Proceedings of the 2nd IEEE Systems, Man and Cybernetics Information Assurance Workshop (IAW 2001), 2001.
[10] X. Lin, P. Zavarsky, R. Ruhl, D. Lindskog, Threat modeling for CSRF attacks, in: Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 03, IEEE Computer Society, Washington, DC, USA, 2009, pp. 486–491, doi:10.1109/CSE.2009.372.
[11] A. Buldas, T. Mägi, Practical Security Analysis of E-Voting Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
[12] E.L. Lazarus, D.L. Dill, J. Epstein, J.L. Hall, Applying a reusable election threat model at the county level, in: Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections, 2011.
[13] E.J. Byres, M. Franz, D. Miller, The use of attack trees in assessing vulnerabilities in scada systems, in: IEEE Conf. International Infrastructure Survivability Workshop, 2004.
[14] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, K. Stoddart, A review of cyber security risk assessment methods for SCADA systems, Comput. Secur. 56 (2) (2016) 1–27.
[15] K. Edge, R. Raines, M. Grimaila, R. Baldwin, R. Bennington, C. Reuter, The use of attack and protection trees to analyze security for an online banking system, in: 40th Annual Hawaii International Conference on System Sciences, IEEE, 2007.
[16] C. Phillips, L.P. Swiler, A graph-based system for network vulnerability analysis, in: Proceedings of the 1998 Workshop on New Security Paradigms, ACM, 1998, pp. 71–79.
[17] S. Abraham, S. Nair, A predictive framework for cyber security analytics using attack graphs, Int. J. Comput. Netw. Commun. (IJCNC 2015) 7 (1) (2015).
[18] S. Gupta, J. Winstead, Using attack graphs to design systems, IEEE Secur. Privacy 5 (4) (2007) 80–83.
[19] S. Jha, O. Sheyner, J. Wing, Two formal analyses of attack graphs, in: Proceedings of the 15th IEEE Computer Security Foundations Workshop, IEEE Computer Security Foundations, 2002.
[20] N. Idika, B. Bhargava, Extending attack graph-based security metrics and aggregating their application, IEEE Trans. Dependable Secure Comput. (TDSC) 9 (1) (2012) 75–85.
[21] K. Ingols, M. Chu, R. Lippmann, S. Webster, S. Boyer, Modeling modern network attacks and countermeasures using attack graphs, in: Annual Computer Security Applications Conference (ACSAC 2009), 2009, pp. 117–126.
[22] L. Wang, A. Singhal, S. Jajodia, Measuring the overall network security of network configurations using attack graph, in: Proceedings of 21st Annual IFIP WG.3 Working Conference on Data and Applications Security, 2007.
[23] F. Chen, D. liu, Y. ZhanG, J. Su, A scalable approach to analyzing network security using compact attack graph, in: Journal of Networks, Citeseer.
[24] J. Homer, A. Varikuti, X. Ou, M.A. McQueen, Improving attack graph visualization through data reduction and attack grouping, in: Visualization for Computer Security, Springer, 2008, pp. 68–79.
[25] N. Poolsappasit, R. Dewri, I. Ray, Dynamic security risk management using bayesian attack graphs, IEEE Trans. Dependable Secure. Comput. 9 (1) (2012) 61–74.
[26] K. Ingols, R. Lippmann, K. Piwowarski, Practical attack graph generation for network defense, in: 22nd Annual Computer Security Applications Conference (ACSAC 2006), 2006, pp. 121–130, doi:10.1109/ACSAC.2006.39.
[27] J.B. Hong, D.S. Kim, Towards scalable security analysis using multi-layered security models, J. Netw. Comput. Appl. 75 (2016) 156–168. https://doi.org/10.1016/j.jnca.2016.08.024.
[28] R. Dewri, N. Poolsappasit, I. Ray, D. Whitley, Optimal security hardening using multi-objective optimization on attack tree models of networks, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, in: CCS 2007, 2007, pp. 204–213.
[29] S. Mauw, M. Oostdijk, Foundations of attack trees, in: Proceeding of International Conference on Information Security and Cryptology (ICISC 2005) LNCS 3935, Springer, 2005, pp. 186–198.
[30] D. Suguo, Z. Haojin, Security Assessment in Vehicular Networks, Springer New York, New York, NY, 2013.
[31] J. Hong, D.S. Kim, HARMs: hierarchical attack representation models for network security analysis, in: Proceedings of the 10th Australian Information Security Management Conference on SECAU Security Congress (SECAU 2012), 2012.
[32] B. Louise, T. Henry, Bring your own device, ITNOW 54 (1) (2012) 24–25.
[33] A. Arora, C. Forman, A. Nandkumar, R. Telang, Competitive and Strategic Effects in the Timing of Patch Release, Fifth Workshop Economic Information Security 2006, Cambridge, UK.
[34] J.B. Hong, D.S. Kim, A. Haqiq, What vulnerability do we need to patch first? in: Proceeding of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2014, pp. 684–689, doi:10.1109/DSN.2014.68.
[35] M. Frigault, L. Wang, A. Singhal, S. Jajodia, Measuring network security using dynamic bayesian network, in: Proceedings of the 4th ACM Workshop on Quality of Protection (QoP), ACM, 2008, pp. 23–30.
[36] H.M.J. Almohri, L.T. Watson, D. Yao, X. Ou, Security optimization of dynamic networks with probabilistic graph modeling and linear programming, IEEE Trans. Dependable Secure Comput. (TDSC) 13 (4) (2016) 474–487, doi:10.1109/TDSC.2015.2411264.
[37] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, S. Xu, A survey on systems security metrics, ACM Comput. Surv. (CSUR 2016) 49 (4) (2016) 62.
[38] S.Y. Enoch, J.B. Hong, M. Ge, D.S. Kim, Composite metrics for network security analysis, Software Netw. 2017 (1) (2017) 137–160, doi:10.13052/jsn2445-9739.2017.007.
[39] CVSS, CVSS Version 3 - Forum for Response and Security Team, accessed March 20, 2017. https://www.first.org/cvss.
[40] M.S.K. Awan, P. Burnap, O. Rana, Identifying cyber risk hotspots: A Framework for measuring temporal variance in computer network risk, Comput. Secur. 57 (2016) 31–46. https://doi.org/10.1016/j.cose.2015.11.003.
[41] M. Ge, J.B. Hong, W. Guttmann, D.S. Kim, A framework for automating security analysis of the internet of things, J. Netw. Comput. Appl. 83 (2017) 12–27.
[42] F. Arnold, D. Guck, R. Kumar, M. Stoelinga, Sequential and parallel attack tree modelling, in: Proceedings of the International Conference on Computer Safety, Reliability and Security, Springer International Publishing, Cham, 2015, pp. 291–299, doi:10.1007/978-3-319-24249-1_25.
[43] J.B. Hong, D.S. Kim, T. Takaoka, Scalable attack representation model using logic reduction techniques, in: Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom, 2013), IEEE, 2013, pp. 404–411, doi:10.1109/TrustCom.2013.51.
[44] B. Schneier, Attack trees, Dr. Dobbs J. Software Tools 24 (12) (1999) 21–29.
[45] MITRE-Corporation, Common Vulnerabilities and Exposures, accessed August 12, 2017. https://cve.mitre.org/cve/cna.html.
[46] R. Ortalo, Y. Deswarte, M. Kaaniche, Experimenting with quantitative evaluation tools for monitoring operational security, IEEE Trans. Software Eng. 25 (5) (1999) 633–650.

[47] W. Li, R. Vaughn, Security research involving the modeling of network exploitations graphs, in: Proceedings of 6th IEEE International Symposium Cluster Computing and Grid Workshops, 2006.

[48] CIS, The Center for Internet Security: Security Metrics, accessed April 21, 2017. https://benchmarks.cisecurity.org/tools2/metrics/CIS_Security_Metrics_v1.1.0.pdf.

[49] S. Liu, B. Cheng, Cyberattacks: why, what, who, and how, IT Professional 11 (3) (2009) 14–21, doi:10.1109/MITP.2009.46.

[50] Nmap, Nmap-network mapper, accessed August 12, 2017. http://nmap.org/index.html.

[51] OpenVAS, Open Source Vulnerability Scanner and Manager, accessed March 20, 2017. http://www.openvas.org/.

[52] S. Donohue, Distribution of Software Updates via a Computer Network, 2001, US Patent 6,199,204 (Mar. 6, 2001). https://www.google.com/patents/US6199204.

[53] P. Mittal, D. Kim, Y.-C. Hu, M. Caesar, Mirage: towards deployable DDos defense for web applications, arXiv:1110.1060 (2011).

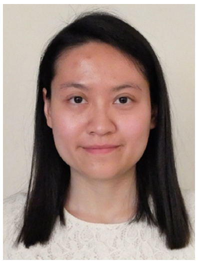
[54] J. Pamula, S. Jajodia, P. Ammann, V. Swarup, A weakest adversary security metrics for network configuration security analysis, in: Proceedings of Second ACM Workshop Quality of Protection, ACM, 2006, pp. 31–38.

[55] R. Lippmann, K. Ingols, C. Scott, K. Piwowarski, K. Kratkiewics, M. Artz, R. Cunningham, Validating and restoring defense in depth using attack graphs, in: Proceedings of Military Communications Conference, 2006, pp. 31–38.

[56] L. Wang, S. Jajodia, A. Singhal, S. Noel, k-Zero Day Safety: Measuring the Security Risk of Networks against Unknown Attacks, Springer-Verlag, Berlin, Heidelberg, 2010.

[57] P.K. Manadhata, J.M. Wing, An attack surface metric, IEEE Trans. Softw. Eng. 37 (3) (2011) 371–386, doi:10.1109/TSE.2010.60.

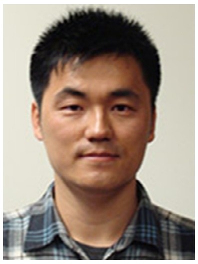
Simon Yusuf Enoch is a lecturer in the Department of Computer Science, Federal University Kashere, Gombe, Nigeria. He received his M.Sc. degree in Computer Science from the University of Ibadan, Nigeria. He is currently a Ph.D. student at the University of Canterbury, New Zealand under the supervision of Dr. Dong Seong Kim. His research interests are security modelling and analysis of enterprise networks, cloud networks and Moving Target Defence.



Mengmeng Ge is an Analytics Engineer in Telstra New Zealand. She received her Ph.D. degree in Computer Science from the University of Canterbury, New Zealand, under the topic of graphical security modeling and assessment for the Internet of Things. Her research interests are graphical security modeling, Internet of Things, software defined networking and data science in cyber security.

Jin B. Hong is a lecturer in the Department of Computer Science and Software Engineering at the University of Western Australia, Australia. He received his Ph.D. degree in Computer Science from the University of Canterbury, New Zealand. His research interests are security modeling and analysis of computer and networks including cloud computing, SDN and IoT, and Moving Target Defense. He is a member of the IEEE.

Hani Alzaid received his B.Eng. in Computer Engineering from King Saud University (Riyadh, KSA) in 2000, his M.S. from University of New South Wales (Sydney, Australia) in 2005, and his Ph.D in Wireless Sensor Networks Security from Queensland University of Technology (Brisbane, Australia) in 2011. Currently, Hani is the co-founder of the National Center for Cybersecurity Technology (C4C) at King Abdulaziz City for Science and Technology (KACST) in which he supervises several national research projects in the Cyber Security domain. Since 2012, Dr. Alzaid is also a board member at the non-profitable Saudi Computers Society and he is also the chairman for the consultancy committee. He also maintained an IT counselor position at SHURA council between 2014 and 2016. Currently, he is an advisor to H.E. the minister of Finance. Hani brings 18 years of experience in the field of information technology and has a diverse background in both academia and industry settings. He has extensive experience in research, instructional, enterprise computing, and cyber security. This hybrid experience has contributed to the winning of many regional awards in the last three years (2015-2017) including ICT and Knowledge Management Leadership Excellence Award and IT Man of the Year.



Dong Seong Kim is the Director of the University of Canterbury Cyber Security Lab. He is a Senior Lecturer (the position is tenured and roughly equivalent to an associate professor in the North American system) in Cyber Security in the Department of Computer Science and Software Engineering at the University of Canterbury, Christchurch, New Zealand. He received Ph.D. degree in Computer Engineering from the Korea Aerospace University in February 2008. He was a visiting scholar at the University of Maryland, College Park, Maryland in the US during the year of 2007 in Prof. Virgil D. Gligor Research Group. From June 2008 to July 2011, he was a postdoc at Duke University, Durham, North Carolina in the US in Prof. Kishor S. Trivedi. His research interests are in security and dependability for systems and networks; in particular, Intrusion Detection using Data Mining Techniques, Security and Survivability for Wireless Ad Hoc and Sensor Networks and Internet of Things, Availability and Security modelling and analysis of Cloud computing, and Reliability and Resilience modelling and analysis of Smart Grid.