Assessing Security and Dependability of a Network System Susceptible to Lateral Movement Attacks

Hongyue Kang[a*], Bo Liu [a*], Jelena Mišić[b+], Vojislav B. Mišić[b+], Xiaolin Chang[a*]
[a]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, P. R. China
[b]Ryerson University, Toronto, ON, Canada
Email: [*]{18120474, 13112067, xlchang}@bjtu.edu.cn; [+]{ jmisic, vmisic}@ryerson.ca

Abstract—Lateral movement attack performs malicious activities by infecting part of a network system first and then moving laterally to the left system in order to compromise more computers. It is widely used in various sophisticated attacks and plays a critical role. This paper aims to quantitatively analyze the transient security and dependability of a critical network system under lateral movement attacks, whose intruding capability increases with the increasing number of attacked computers. We propose a survivability model for capturing the system and adversary behaviors from the time instant of the first intrusion launched from any attacked computer to the other vulnerable computers until defense solution is developed and deployed. Stochastic Reward Nets (SRN) is applied to automatically build and solve the model. The formulas are also derived for calculating the metrics of interest. Simulation is carried out to validate the approximate accuracy of our model and formulas. The quantitative analysis can help network administrators make a trade-off between damage loss and defense cost.

Keywords—Lateral movement attack; Transient analysis; Survivability model; Security; Dependability.

I. INTRODUCTION

Lateral movement refers to an attacking path through the network, hopping between computers to find an intended goal. It is widely used in most sophisticated attacks, including advanced persistent threats (APTs) [1]. See Fig. 1, which summarizes APT means. Lateral movement attack is very common. As is reported in [2], around one-third of all trespassers conducted lateral movement attacks. Lateral movement attacks are not easy to be detected and have caused huge damage due to their features of concealment and complexity. Catching and resisting lateral movement are critical to resist various sophisticated attacks [3]. Accordingly, it is especially important to understand the principle of lateral movement attacks.

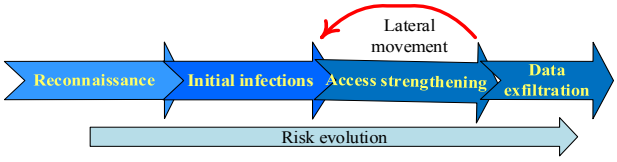

Fig. 1. Process of APT attack

In recent years, a lot of studies have been carried out to detect lateral movement attacks including graph-based methods [4]-[5], machine learning approach [6], framework [7]-[8] and various tools [9]-[10]. However, few of them investigated the detailed lateral movement attack defense rather considered the whole APT attack defense [11]-[12] mostly. The reason is that it is not easy to capture the system states under lateral movement attack for the uncertainty of attacker's behavior. Nearly two years, the researchers in [13]-[14] characterized the lateral movement attacks as an epidemic model according to its feature of propagation, but they restricted the adversary traversed the network via credential chaining, where the adversary stole credentials from one system, used them to certify to another, and repeated the process. Actually, there are other methods for attackers to conduct lateral movement attacks. Our model mainly concentrates on the system behavior under lateral movement attacks and there is no limitation to the attack mode of the attacker. The authors in [15] quantitatively studied the transient security of a vulnerable network system under lateral movement attacks, which explored state-space modeling techniques to construct a survivability model for quantitative analysis. But they assumed that the intruding capability of the adversary was unchanged. However, in real systems the rate of intrusion propagation can be related to the number of intruded computers. Similarly, the mean rates of data leakage and completing data leakage were equal in any situation in that paper. These assumptions may not be true in a realistic network system.

This paper relaxes these assumptions and aims to quantitatively analyze the transient security and dependability of a vulnerable network system under lateral movement attacks. In the system, the attacking capability (e.g. intruding, exfiltrating) of the adversary increases with the increasing number of attacked computers and each attacked computer can cause damage independently and randomly. We develop a survivability model for capturing the system and adversary behaviors from the time that the attack touches successfully the system until defensive mechanisms are deployed. We also consider the case where the number of active computers may vary due to physical damage, which is studied in this paper. Stochastic Reward Nets (SRNs) is applied to build and solve the survivability model of the complicated behaviors of the system and the adversary. We also can develop the formulas for calculating the metrics of interest. The model and formulas can help the network administrator make a trade-off between damage loss and defense cost. Simulation is also conducted to verify the accuracy of our model and formula.

The paper is organized as follows. Section II introduces the system description and the proposed analytical model.

Numerical analysis is given in Section III. Section IV concludes the paper.

II. System Description and Model

This section first describes the system. Then the model and the formulas for calculating metrics of interest are presented. Table II defines the variables to be used in the rest of the paper and gives their default settings.

A. System Description

The vulnerable network system is assumed to have m computers. Each computer has vulnerabilities and is susceptible to be intruded. Each computer is at one of four different states (described in Table I) from the time that the attacker gets into the system until defense mechanisms are deployed. The first state is *Vulnerable*, denoting that the computer has vulnerabilities but has not been targeted by attackers. The second state is *Intruded*, denoting that the computer has been intruded by attackers exploiting the vulnerabilities in itself but data leakage has not happened. The third state is *Exfiltrated*, representing that information leakage is occurring on the computer after this computer is at state *Intruded*. The last state is *Secure*, denoting that the defensive mechanisms are deployed and the computer is immune to attack.

Here, we assume that the fault positive rate is very small, namely, all detected abnormities are attacked-related. Once the aggressive behavior is detected, the administrators begin to design the defensive mechanisms immediately. If the defensive mechanisms are deployed, the system can resist those attack and turn into *Secure* state.

TABLE I. COMPUTER STATE

Computer state name	*Definition*
Vulnerable	Computer has vulnerabilities but has not been discovered by attackers
Intruded	Computer has been intruded by attackers exploiting the vulnerabilities in itself but data has not been stolen.
Exfiltrated	Information leakage occurs on the *Intruded* computer.
Secure	Defensive mechanisms are deployed and the computer is immune to attacks.

TABLE II. NOTATIONS AND DEFAULT SETTINGS OF VARIABLES

Notation	*Definition*	*Default Value*
m	The number of computers in the system	3
i	Denote that i computers are at state *Intruded*	-
j	Denote that j of i *Intruded* computers are at state *Exfiltrated*	-
$1/\beta$	Mean time to intrude one computer	2 days
$1/\lambda_j$	Mean time to compromise j *Intruded* computers	4+2·j days
$1/\mu_j$	Mean time for finishing data exfiltration on j *Intruded* computers	2·j days
$1/\gamma$	Mean time for fixing	2 days
$1/f$	Mean time for damaging	10 days
$1/g$	Mean time for repairing	2 days

When the attacker enters the vulnerable system, he begins to intrude *Vulnerable* computers. The attacker is assumed to be able to only intrude one computer with rate β at an instant time. But the rate of next *Vulnerable* computer being intruded is related to the number of *Intruded* computers in the current system. For example, there is one *Intruded* computer in the system and it has the ability to intrude another *Vulnerable* computer. We assume the intruded rate of one *Vulnerable* computer is β when there is one *Intruded* computer in the system. Then the intruded rate of another *Vulnerable* computer is 2β when there are two *Intruded* computers in the system. After intruding computers in the system, the attacker can continue to intrude the left *Vulnerable* computers or find valuable data on the *Intruded* computers. When data begins being leaked, the computer enters *Exfiltrated* state. Note that an *Exfiltrated* computer can return to *Intruded* state after the attacker completes stealing data. We assume the attacker can carry out data exfiltration on multiple computers simultaneously but with different data exfiltration rate λ_i for the reason of that stealing data on more *Intruded* computers needs more time. The rate of completing data exfiltration with rate μ_i is similar to data exfiltration. Besides, to fully consider the actual situation, during the attack process, the computers can be damaged at any time and at any state with rate *f*. Thus, the number of active computers in the system will change. We assume only one computer can be damaged at a time. When the network administrator finds the computer is damaged, he will repair the damaged computer with rate *g*. And then, the state of damaged computer in the system will return to its previous state.

The metrics of interest in this paper include:

***Metric m*1)**: Probability that only i computers are at state *Intruded* at time *t*.

***Metric m*2)**: Accumulated time that only i computers at state *Intruded* in time interval [0, *t*).

***Metric m*3)**: Probability that j of i *Intruded* computers are at state *Exfiltrated* at time *t*.

***Metric m*4)**: Accumulated time that j of i *Intruded* computers are at state *Exfiltrated* in time interval [0, *t*).

***Metric m*5)**: Probability that the system is at state *Secure* at time *t*.

B. System Model

In this subsection, we describe the survivability model and define the notations used in the paper. A two-tuple (i, j) is defined to present the system states, where i represents the number of *Intruded* computers and j stands for the number of *Exfiltrated* computers in the system. For example, (2,1) denotes that there are two *Intruded* computers and one of them has been data-exfiltrated in the system. Now we give an example of three computers in the system. The model has eleven system states, shown in Fig. 2. At first, no attack is detected in the system, so the initial state is (0,0). When the attacker enters the system, there is one *Intruded* computer. The system state turns into (1,0). The whole process of attacking can be divided into six processes:

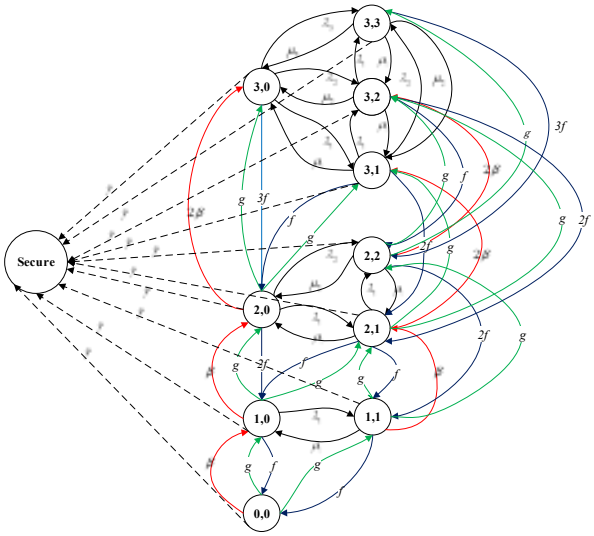

Fig. 2. Survivability model with different intruded, data exfiltration and completing data exfiltration rates

Intruding process: Once there is one intruded computer, the *Intruded* computers have the ability to intrude other *Vulnerable* computers. As is shown in Fig. 2 with the red line. The first element i of (i, j) increases, which represents the intruding process occurred.

Exfiltrating process: After a computer is intruded, the attacker can search sensitive data on it. If the second element of (i, j) increases, which denotes the exfiltrating process happened. We define the data exfiltrating rate as λ_i and use the example in Fig. 2 to illustrate. For example, there are two *Intruded* computers, so the system state is (2,0). When the attacker steals data on one computer, the exfiltrating rate is λ_1. System state moves from (2,0) to (2,1). When the stealer leaks data on two computers at the same time, the exfiltrating rate is λ_2. System state moves from (2,0) to (2,2). The reason of $\lambda_1 < \lambda_2$ is that the mean time of data leaking on one computer is fewer than two computers under the condition of limited resources.

Completing data leakage process: This process is similar to the exfiltrating process. If the second element of (i, j) decreases, which means the completing data leakage process took place. The rate of completing data leakage is defined as μ. Similarly, different rates are set, shown in Fig. 2.

Damaging process: During the process of attacking, the computer can break down due to some physical factors. Then the system state will change. For example, the current system is (3,2). There are three *Intruded* computers and two of them have been data-*Exfiltrated*. If the *Intruded* but non-*Exfiltrated* computer is damaged, the system state turns into (2,2) with rate *f*. If one of two *Exfiltrated* computer is damaged, the system state will change from (3,2) to (2,1) with rate *2f*. Because any one of the two *Exfiltrated* computers is damaged, the system state can enter (2,1). The detailed damaging process is shown in Fig. 2 with the blue line.

Repairing process: When the computer is damaged, the administrator can find the problem and repair it. Then the damaged computer can return to its previous state because the attack code still exists in the computer and is not removed. Therefore, the system state returns to its previous state. The specific repairing process is presented in Fig. 2 with the green line.

Fixing process: When the defense mechanisms are deployed well, the computer can defend against the attack and the computer can return to the state of *Non-Intruded*. Therefore, no matter which state the system is at, once the defense mechanism is deployed, the system enters *Secure* state with same rate γ, which is shown with the grey line in Fig. 2.

Now we use the SPNP software package [16] to calculate the above metrics as follows:

***Metric m*1)**: Expected number of tokens at (1,0), (2,0), (3,0) at time *t*.

***Metric m*2)**: Expected accumulated number of tokens at (1,0), (2,0), (3,0) in time interval [0, *t*).

***Metric m*3)**: Expected number of tokens at (3,1), (3,2), (3,3) at time *t*.

***Metric m*4)**: Expected accumulated number of tokens at (3,1), (3,2), (3,3) in time interval [0, *t*).

***Metric m*5)**: Expected number of tokens at *Secure* at time *t*.

When there are *m* computers in the system, the metrics are as follows:

***Metric m*1)**: At state (*i*,0), expected number of tokens of #(*i*,0) at time *t*, where $1 \le i \le m$.

***Metric m*2)**: At state (*i*,0), expected accumulated number of tokens of #(*i*,0) in time interval [0, *t*), where $1 \le i \le m$.

***Metric m*3)**: At state (*m*,*j*), expected number of tokens of #(*m*,*j*) at time *t*, where $1 \le j \le m$.

***Metric m*4)**: At state (*m*,*j*), expected accumulated number of tokens of #(*m*,*j*) at time *t*, where $1 \le j \le m$.

***Metric m*5)**: At state *Secure*, expected number of tokens of #*Secure* at time *t*.

Let $P_{ij}(t)$ denote the probability that the system is at state (i, j) at time *t*. Also, let R_{ij} denote the loss of that the system at state (i, j). Let $L_{ij}(t)$ denote the expected system loss of that the system is at state (i, j) at time t, and then the system loss $L_{ij}(t)$ at time *t* can be defined as:

$$L_{ij}(t) = R_{ij} P_{ij}(t)$$

III. Numerical Analysis

This section presents detailed evaluations of our model and formulas for computing metrics by comparing analytical and simulation results. All the time intervals are exponentially distributed. Table II gives the default settings of parameters. The value of $\beta, \lambda_1, \mu_1, \gamma$ are set according to [15]. Other values are set to highlight the effectiveness of our model. SPNP software package [16] is used to quantitative analysis. The results of *Metric m*1-*m*6 under the condition of four

computers in the system are shown in Fig.3-Fig.7, respectively. The detailed explanation of these results is given as follows.

***Metric m*1)** Fig. 3 indicates the transient probability of i computers at *Intruded* state. There are four computers in the network system. Pi0 denotes the probability that i computers are intruded but all of them have no data loss. The probability increases from the 1st day and then it starts to decrease from the 2nd day. The reason is that once the attacker begins the intrusion to the system, more computers in the system will receive intrusion. And the mean fixed time is 2 days, so the system can be secure quickly. Once the system is secure, there is no intruding and exfiltrating activity. Also, we can see that in the first two days, P10 > P20 > P30 > P40. That means with the increasing number of *Intruded* computers in the system, the intruding probability decreases. The reason for that is that the attacker can only intrude one computer at a time instant and intruding more computers needs more time. After two days, P10 < P20 < P30. That is because the mean fix time is 2 days and the fewer computer at *Intruded* state can be fixed more quickly. The reason of P40 > P30 is that every *Intruded* computer has the ability to intrude others and more *Intruded* computers have a higher probability to intrude the *Vulnerable* computers successfully.

***Metric m*2)** Fig. 4 represents that the intruding accumulated time is 0.399 for (1,0), 0.232 for (2,0), 0.175 for (3,0), 0.339 for (4,0) respectively. That means by the 10th days, the total time that the system is at state (1,0) is longest. And (1,0) > (2,0) > (3,0). Obviously, the system will spend more time at the state that the number of *Intruded* computers is small. Because the state, a small number of *Intruded* computers, is easy to reach. However, the total time at (4,0) is bigger than (3,0) and (2,0) sometimes. The reason for that is that more and more computers are likely to be intruded over time and the mean time of fixing time is not enough. Because when there are more *Intruded* computers in the system, there is a larger probability to intrude another *Vulnerable* one.

***Metric m*3)** Fig. 5 presents the probability of j computers at *Exfiltrated* state. We assume that stealing data on different numbers of computers with different exfiltrating rate λ_i. P4i denotes that there are four *Intruded* computers and i computers that have been data-exfiltrated. P41, P42, P43, P44 represent the result of $1/\lambda = 6$ days, 8 days, 10 days, 12 days respectively. We can see that P41 < P42 < P43 < P44. That is because we define the different rates of data stealing. That means the rate of stealing one computer is smaller than the rate of stealing two computers. Namely, the smaller the number of stealing computers, the bigger the rate, which is suitable in a real system, since the attack resources are limited. Stealing data on more computers needs more time. Besides, in the first three days, the probability of data exfiltration increases and then decreases. That is because the defense mechanisms are deployed. After the fixed system is ready, the attacker cannot steal data on any computer. Therefore, the probability decreases after three days.

***Metric m*4)** Fig. 6 shows that the exfiltration accumulated time. That is 0.161 for (4,1), 0.095 for (4,2), 0.059 for (4,3), 0.036 for (4,4) at the 10th day respectively. We can see (4,1) > (4,2) > (4,3) > (4,4) at any time during the 10 days. Similarly, we can understand that the state (4,1) is easy to reach than others. Because it needs more time to steal data on more *Intruded* computers.

***Metric m*5)** Fig. 7 shows the probability that system is at state *Secure*. We find that the probability that the system at state *Secure* keeps increasing during the 10 days and the probability at state *Secure* at the 10th day will be close to 1. That means when the fixed system is ready, the whole vulnerable system will be at state *Secure* finally.

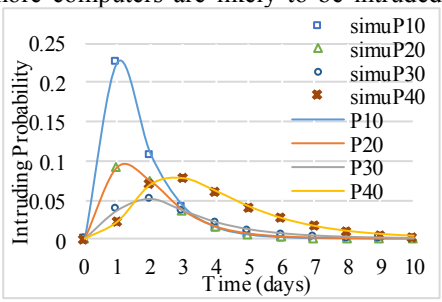

Fig. 3. Intruding probability

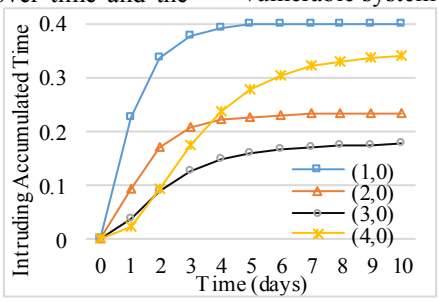

Fig. 4. Intruding accumulated time

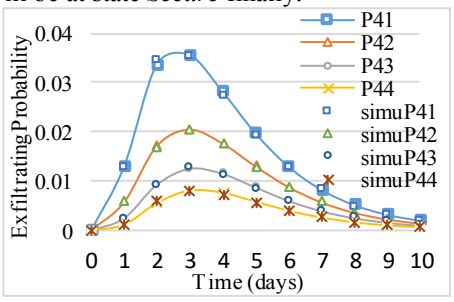

Fig. 5. Exfiltrating probability

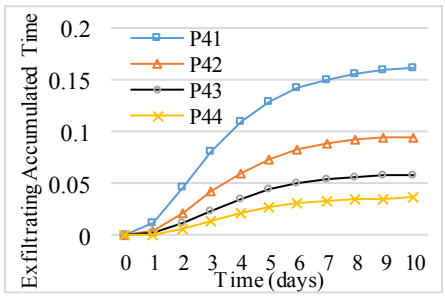

Fig. 6. Exfiltrating accumulated time

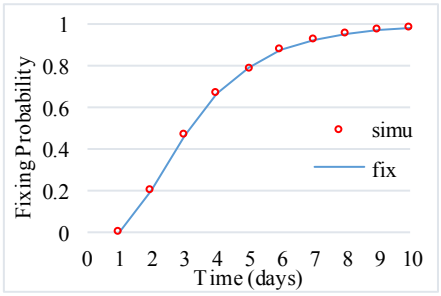

Fig. 7. Fixing probability

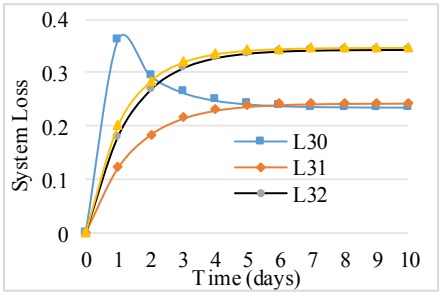

Fig. 8. System loss

When the computers are intruded or data leaked, there will be some loss to the system. After knowing the system state after being attacked, we can calculate the system loss timely, shown in Fig. 8. We assume there are three computers in the system and all of them have been intruded. Let $L_{ij}(t)$ denote the loss that the system at state (i, j) at time t. We assume that when the system is at state (3,0), the expected loss of the system is 0.5. When the system is at state (3,1), the expected loss is 1. When the system is at state (3,2), the expected loss is 2. When the system is at state (3,3), the expected loss is 3. From the picture, we can see that the loss is maximum when the system is at state (3,0) in the first two days. The reason for that is stealing data on computers needs more time. The attacker has not had enough time to leak the data and most of computers in the system are at *Intruded* state. Then when the system is at (3,2) and (3,3), the loss of the system is maximum after two days. That is because when time is long enough, more computers data can be stolen and at state *Exfiltrated* and cause more losses to the system. That indicates that in the early stages of an attacker attacking the system, *Intruded* state will cause a bigger loss to the system. And then in the later stages, the more the number of *Exfiltrated* computers, the larger the system losses. Then for the network administrator, he should first handle the *Intruded* computers at the early stage and deal with the *Exfiltrated* computers at the later stage. And we also give the specific loss value at any time shown in Fig. 8. We just abstractly define the system state loss. The network administrator can make the system state loss according to its own actual situation. After obtaining the whole system loss, the network manager can balance the system loss and defense cost. If the system loss is too large when the network administrator detects it, the manager may worry funds and give up fixing it. If the system loss is controllable and the defense cost is within affordable range, he could make plans to defense the attack.

IV. Conclusions and Future Work

This paper develops a survivability model for quantitatively analyzing the security and dependability of a vulnerability network system under lateral movement attacks from the time that the attack touches successfully the system until defensive mechanisms are deployed. The proposed model can capture the behaviors of the scenario where (i) the rate of *Vulnerable* computers being intruded increases with the increasing number of *Intruded* computers, and (ii) each *Intruded* computer can perform its compromising activity independently and randomly. We also introduce the formulas for computing metrics. The proposed model and formula are verified to be approximately accurate by comparing numerical results and simulation results.

References

[1] Atul Bohara, Mohammad A. Noureddine, Ahmed M. Fawaz, William H. Sanders: An Unsupervised Multi-Detector Approach for Identifying Malicious Lateral Movement. SRDS 2017: 224-233.

[2] Daniel Fraunholz, Daniel Schneider, Janis Zemitis, Hans Dieter Schotten: Hack My Company: An Empirical Assessment of Post-exploitation Behavior and Lateral Movement in Cloud Environments. CECC 2018: 3:1-3:6.

[3] Smokescreen Technologies Pvt. Ltd. The Top 20 Lateral Movement Tactics. Technical report, August 2016

[4] Qingyun Liu, Jack W. Stokes, Rob Mead, Tim Burrell, Ian Hellen, John Lambert, Andrey Marochko, Weidong Cui: Latte: Large-Scale Lateral Movement Detection. MILCOM 2018: 1-6.

[5] Ahmed M. Fawaz, Atul Bohara, Carmen Cheh, William H. Sanders: Lateral Movement Detection Using Distributed Data Fusion. SRDS 2016: 21-30.

[6] Mingyi Chen, Yepeng Yao, Junrong Liu, Bo Jiang, Liya Su, Zhigang Lu: A Novel Approach for Identifying Lateral Movement Attacks Based on Network Embedding. ISPA/IUCC/BDCloud/SocialCom/SustainCom 2018: 708-715.

[7] Harinder Pal Singh Bhasin, Elena Ramsdell: Data Center Application Security: Lateral Movement Detection of Malware using Behavioral Models. SMU Data Science Review 1(2): 10 (2018).

[8] Zhihong Tian, Wei Shi, Yuhang Wang, Chunsheng Zhu, Xiaojiang Du, Shen Su, Yanbin Sun, Nadra Guizani: Real Time Lateral Movement Detection based on Evidence Reasoning Network for Edge Computing Environment. CoRR abs/1902.04387 (2019).

[9] Airull Azizi Awang Lah, Rudzidatul Akmam Dziyauddin, Marwan Hadri Azmi: Proposed Framework for Network Lateral Movement Detection Based On User Risk Scoring in SIEM. TAFGEN 2018: 149-154.

[10] Jain Utkarsh: Lateral movement detection using ELK stack[D] (2018).

[11] Ramchandra Yadav, Raghu Nath Verma, Anil Kumar Solanki: Defense-in-Depth Approach for Early Detection of High-Potential Advanced Persistent Attacks. Soft Computing: Theories and Applications: 205-216 (2019).

[12] Mohammad A. Noureddine, Ahmed M. Fawaz, William H. Sanders, Tamer Basar: A Game-Theoretic Approach to Respond to Attacker Lateral Movement. GameSec 2016: 294-313.

[13] Luxing Yang, Pengdeng Li, Xiaofan Yang, YuanYan Tang: A risk management approach to defending against the advanced persistent threat. IEEE Transactions on Dependable and *Secure* Computing (Early Access) (2018).

[14] Brian A. Powell: The epidemiology of lateral movement: exposures and countermeasures with network contagion models. CoRR abs/1903.07741 (2019).

[15] Yu Shi, Xiaolin Chang, Ricardo J. Rodríguez, Zhenjiang Zhang, Kishor S. Trivedi: Quantitative security analysis of a dynamic network system under lateral movement-based attacks. Rel. Eng. & Sys. Safety 183: 213-225 (2019).

[16] Gianfranco Ciardo, Jogesh K. Muppala, Kishor S. Trivedi: SPNP: Stochastic Petri Net Package. PNPM 1989: 142-151.