Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

DISCRETE EVENT SIMULATION OF CYBER MAINTENANCE POLICIES ACCORDING TO NESTED BIRTH AND DEATH PROCESSES

Akshay Krishna Murali
Enhao Liu
Theodore T. Allen

Department of Integrated Systems Engineering
The Ohio State University
1971 Neil Avenue-210 Baker Systems
Columbus, OH 43221, USA

ABSTRACT

This article proposes a novel discrete event simulation model for predicting cyber maintenance costs under multiple scenarios. In this study, our model of the evolution of computer hosts is similar to the Susceptible-Infected-Removed (S-I-R) epidemiological model. A double or "nested birth and death" construction is used for the hosts and the vulnerabilities on the hosts. The objectives of the model are to study the benefits and drawbacks of current scanning policy and maintenance policy, evaluate cost-effective alternatives, and investigate the significance of celebrity vulnerabilities.

1 INTRODUCTION

Discrete event simulation has been used to model cyber-security related costs in various fields ranging from power grids (Nguyen et al. 2015) to politics (Naugle et al. 2016). In an organizational setup, forecasting of cyber maintenance costs using simulation has been explored by Allen and Liu (2018) focusing on evolution of a computer host through different states. Computer hosts can be any system that has access to a shared network, such as personal computers, servers etc. Each of these devices can have software vulnerabilities in their system, which are weaknesses that a hacker can exploit.

In this paper, we propose a simulation model to extend the assumptions for maintenance policies by adapting a Susceptible-Infected-Removed (S-I-R) model from epidemiology. The computer hosts are generated or "born" at certain times, acquire vulnerabilities "get sick" and the vulnerabilities are patched "recovered" and disposed at the end of their life-cycle "die". While prior studies consider an SIR model for the evolution of viruses or vulnerabilities (Kephart et al. 1993) we use it to design the host evolution in the proposed model. Hernández-Suárez et al. (2010) show that the inter-arrival time of an S-I-R model follows a non-stationary Poisson process (NSPP). We therefore approximate the computer hosts arrival and vulnerabilities generation to a NSPP.

The common maintenance policy for these unobservable hosts has been "out-of-sight is out-of-mind" (OSOM), i.e., auto-patching applied to the hosts which are missing despite actual vulnerabilities and associated risks of exploiting could be hidden on them (Allen and Liu 2018). In this paper, we factor in the observation level as a decision variable to find the best alternative. Recently, experts have begun tagging certain vulnerabilities having qualities similar to Heartbleed and Shellshock as notorious or as "celebrity vulnerabilities" (Allen et al. 2017). Each specimen is recognized to have a high potential of being exploited and therefore require special attention. In the proposed simulation model, we extend upon the past methods by accounting for the celebrity vulnerabilities and penalizing the model for mishandling.

With respect to the development of cyber vulnerabilities maintenance policy, Afful-Dadzie and Allen (2014) proposed a data-driven Markov decision process (DD-MDP) framework in which the security state of host is measured by CVSS and whether or not host being compromised. Roychowdhury (2017) proposed a partially observable MDP (POMDP) framework to address the unobservability issue in hosts compromised state. Allen et al. (2018) developed a Monte Carlo-Bayesian reinforcement learning framework in which the security state is the combination of actual security state and the scenarios of parameters uncertainties. Although these MDP-based/POMDP-based framework could provide cost-effective policies, the evolution of hosts status and whether or not vulnerabilities being patched are only accounted for indirectly.

Here, we are using simulation to shed light on the internal mechanism of such a system. Through simulation we consider a vulnerability-based system where individual actions are applied for each vulnerability based on its severity level. Studies show that typically 70% of the distinct hosts are missing from the monthly scans for vulnerabilities (Allen and Liu 2018). In the proposed simulation model, we also consider the consequences of manually altering the proportion of hosts observed in monthly scans, which allows us to propose a hybrid policy based on the resource constraints of the organization. Resources of organizations can include a variety of factors ranging from manpower in the form of cyber security experts and analysts, and capital invested in cyber maintenance operations. In this paper, we implicitly consider these resources in the form of scan levels and maintenance actions because greater number of resources would result in more extensive cyber maintenance policies.

The remainder of this paper is organized as follows. In section 2, the structure of the proposed model is described. Section 3 discusses the experiments involving common policy and four alternatives. In section 4, the implications for decision-makers and opportunities for future research are given.

2 THE PROPOSED MODEL

The hosts and the vulnerabilities are generated independently, and the active vulnerabilities are assigned to the host depending on host attributes. The observed hosts are scanned once a month and patching is attempted on the vulnerabilities. These vulnerabilities accumulate in the system till patching or an exploit or the disposal of the host. Thus, the vulnerabilities are approximately borne and disposed within the evolution of the host.

Data sources from a university tells that there are approximately 2000 distinct hosts per department on average. Cyber security experts indicate that the recommended period of usage of a computer to be 4 years on average before replacement. Therefore, by Littles Law, we can estimate the number of hosts per year to be $\lambda = L/W$. (i.e.) 2000/4 = 500 distinct hosts per year.

The Nessus scans from the major university usually have about 2,300 distinct vulnerabilities. Typically, the severity level of vulnerabilities is measured by Common Vulnerability Scoring System (Mell et al. 2007). We take the 100 most frequent vulnerabilities grouped into four severity levels namely: Low (CVSS score $0 \sim 3.9$), Medium (CVSS score $4 \sim 6.9$), High (CVSS score $7 \sim 9.9$) and Critical (CVSS score 10), across 500 hosts. Additionally, there are four types of hosts in the proposed simulation model. These are Windows, Linux, Mac and Other (such as general hosts, servers etc.). We also grant administrator privilege to some hosts and also give weights to hosts that may have confidential information stored.

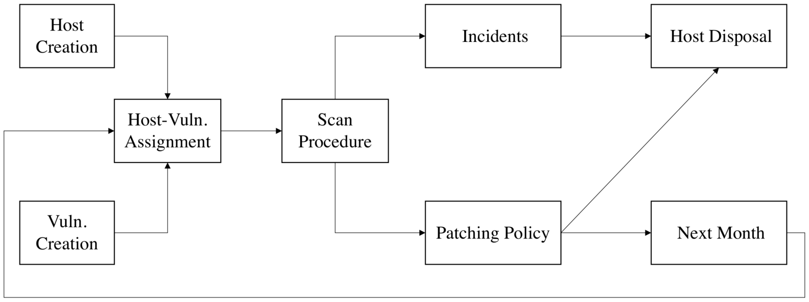


Figure 1: Logic Diagram of ARENA Simulation Model.

2.1 ARENA Simulation Model

The workflow of the proposed simulation model is depicted in Figure 1. The host entities are generated on a monthly basis according to a non-stationary Poisson process in the “Host Creation” module (see Figure 5). Host attributes include operating system, administrator privileges, “whether or not the host has restricted data”, and “whether or not the host entities are clones” are assigned to the entity in a Boolean format. If hosts are clones, they will be batched together since they would have the identical attributes. Once these attributes are assigned, the hosts are then transferred to the vulnerability assignment loop.

We define the attributes of vulnerabilities in the “Vulnerability Creation Module” (see Figure 7). We select 100 most frequent vulnerabilities split into four levels for each severity level in Table 1. There are four distinct vulnerabilities which are known to be celebrity vulnerabilities (i.e., Heartbleed and Shellshock). The other 96 vulnerabilities are also assumed to have a small probability (5%) of bei celebrity vulnerabilities. In addition, vulnerability attributes in Table 1 contain CVSS score, patch availability, active-inactive status, and a Boolean factor for compatibility with OS type. From Nessus scans of the major university, the lifetime of the vulnerabilities was derived based on the time difference between the first appearance date of the vulnerability on hosts to the time of patching vulnerability. Figure 2 presents the histogram of the vulnerability life span (days) for four OS types. According to Hou (2015), the vulnerability life span for several vendors are approximately within 250 days. Therefore, the vulnerability life span in the proposed model is scaled to be uniformly distributed between 60 to 180 days. The average time for patch availability is also scaled to be uniformly distributed between 30 to 90 days. The active-inactive status of the vulnerabilities is decided in the vulnerability creation module where the vulnerabilities are generated as entities based on the severity level. These entities are then updated in a 2D-array which acts as the master list of vulnerabilities. The vulnerability creation module acts as a sub-model where the vulnerabilities are generated, updated, and stay in the system using a “delay” unit till the end of their assigned lifetime, and are then disposed. Once disposed, the activity status is toggled back to 0 in the master list.

Next, we need to assign active vulnerabilities to created hosts. A subset of the master list of vulnerabilities is assigned to each host entity as an attribute array in the “Vulnerability Assignment Loop” (see Figure 6). Here, the system loops through the list of active vulnerabilities, and probabilistically assigns them to the hosts based on OS compatibility. To maintain traceability, the assigned vulnerabilities are stored as a 2D-array attribute in every host, and are dynamically updated.

From the vulnerability assignment loop, the hosts are then routed to the “Nessus scan station”(see Figure 9) where the maintenance path of the host is decided. The host entities are directed based on the determined scan level. We tested six scan levels (10%, 30%, 50% 70%, 90%, and 100%) which corresponds to the percentage of hosts observed in the monthly scans. For a scanned host, the assigned vulnerabilities will be discovered and there is a probability of being exploited. The probability of an exploit is varied based on highest severity level of vulnerabilities on the host. For an unscanned host, there is a higher probability of being exploited and a small probability of being auto-patched by system updates. These hosts tend to accumulate vulnerabilities over months and result in higher costs. When a host is exploited, it is then sent to the “Incident Module” (see Figure 12) where the cost of the exploit is calculated based on whether or not the host has restricted data, and then the host is disposed as it is assumed to be compromised. In the event of no exploits, the hosts are sent to the “Patching station” (see Figure 10). The host entities that arrive at the station are directed based on whether or not they were scanned. The patching station loops through all the active vulnerabilities on the host and action is taken based on the pre-programmed maintenance policy for different severity levels of vulnerabilities. Since knowledge of celebrity vulnerabilities is known in advance, the same policy for celebrity vulnerability is followed irrespective of host being scanned or unscanned.

The maintenance actions for the policies are defined in the “Patching Actions” module (see Figure 11). In our proposed model, four levels of maintenance actions have been considered. Action 1 (Auto-patching) involves software updates whose costs are borne by the software developer. Action 2 (Research-Accept) involves inspection by cyber-maintenance staff. If no patch or solution available for the vulnerability, we accept the risk. Action 3 (Research-compensate) forces the vulnerability to be fixed as risk cannot be accepted. Action 4 (Remediation/Replacement) forces the host to be taken offline (i.e.) the host is disposed.

The hosts are then transferred to the subsequent month through the "Transfer Station" (see Figure 8) where the costs of patching are accumulated, and the model repeats until the hosts are disposed due to remediation or replacement.

Table 1: Master list of Vulnerabilities with known celebrity vulnerabilities: Heartbleed and Shellshock are shown in bold.

Vulns Index	Nessus Plugin	CVSS	Linux	Windows	Mac	Others	Active	Patch	Celebrity
1	10407	2.6	1	1	1	1	0	0	0
2	10759	2.6	1	1	0	1	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
26	11213	4.3	1	1	1	1	0	0	0
27	12217	5	1	1	0	1	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
51	10264	7.5	1	1	0	1	0	0	0
52	23842	7.5	1	1	0	1	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
74	**73404**	**9.4**	**1**	**1**	**1**	**1**	**0**	**0**	**1**
75	**73412**	**9.4**	**1**	**1**	**1**	**1**	**0**	**0**	**1**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
97	84824	10	0	1	0	1	0	0	0
98	86542	10	0	1	0	0	0	0	0
99	**77829**	**10**	**1**	**0**	**1**	**1**	**0**	**0**	**1**
100	**79147**	**10**	**0**	**1**	**0**	**1**	**0**	**0**	**1**

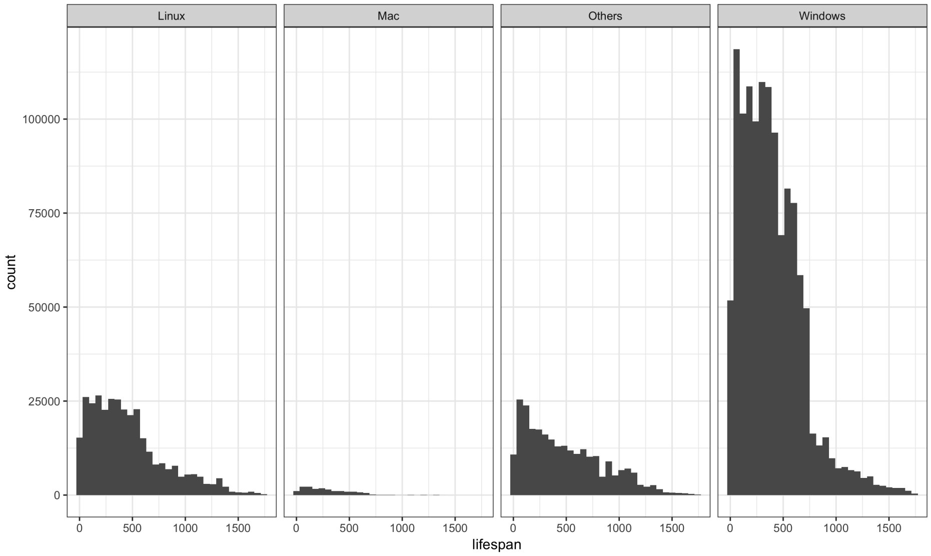


Figure 2: Vulnerability life span (days).

3 RESULTS AND DISCUSSION

Five replications are used in each scan level to get the average expenditure over five years for each particular setup of maintenance policies. The organization does not incur any expenses for Action 1 (auto-patching) according to Allen et al. (2018) and Afful-Dadzie and Allen (2014). However, there is a cost associated for manually patching each vulnerability. Therefore, the estimated costs for Action 2 and Action 3 are $3.5

and $17.5 per vulnerability respectively which are scaled down from Hou (2015) based on patch purchasing cost and labor cost data provided by the chief IT staff of the university. Action 4 accounts for remediation and replacement and has a one-time cost of $1000 per host.

Based on the Cost of Data Breach Study (Ponemon Institute, LLC 2017), the average cost for each record lost in a data breach is $221, and the average total cost paid by organizations is $7.01 million. Nevertheless, the data breaches are extremely rare events. In this paper, we consider a more general class of security events called incidents. The resulting incident costs are dependent on whether the host has any restricted data. According to Hou (2015), incidents where restricted data is compromised has a penalty of $20,000 per host, and for other incidents on normal hosts the penalty is $3000 per host. The total expenditure for each scenario is estimated by summing up monthly expenses with a discount factor of 0.99.

3.1 Comparison of Alternatives

In our analysis, we consider the current OSOM policy (Policy 1_0), along with five alternatives (Policies $0 \sim 4$) and forecast the cyber-maintenance costs over a five-year period. The actions for scanned and unscanned hosts in each policy are summarized in Table 2. We see that in policies 1 through 4, the aggressiveness of patching actions increases. While policies 1 through 4 provide alternatives to the current policy, policy 0 explores the significance of celebrity vulnerabilities by only taking manual action on the celebrity vulnerabilities.

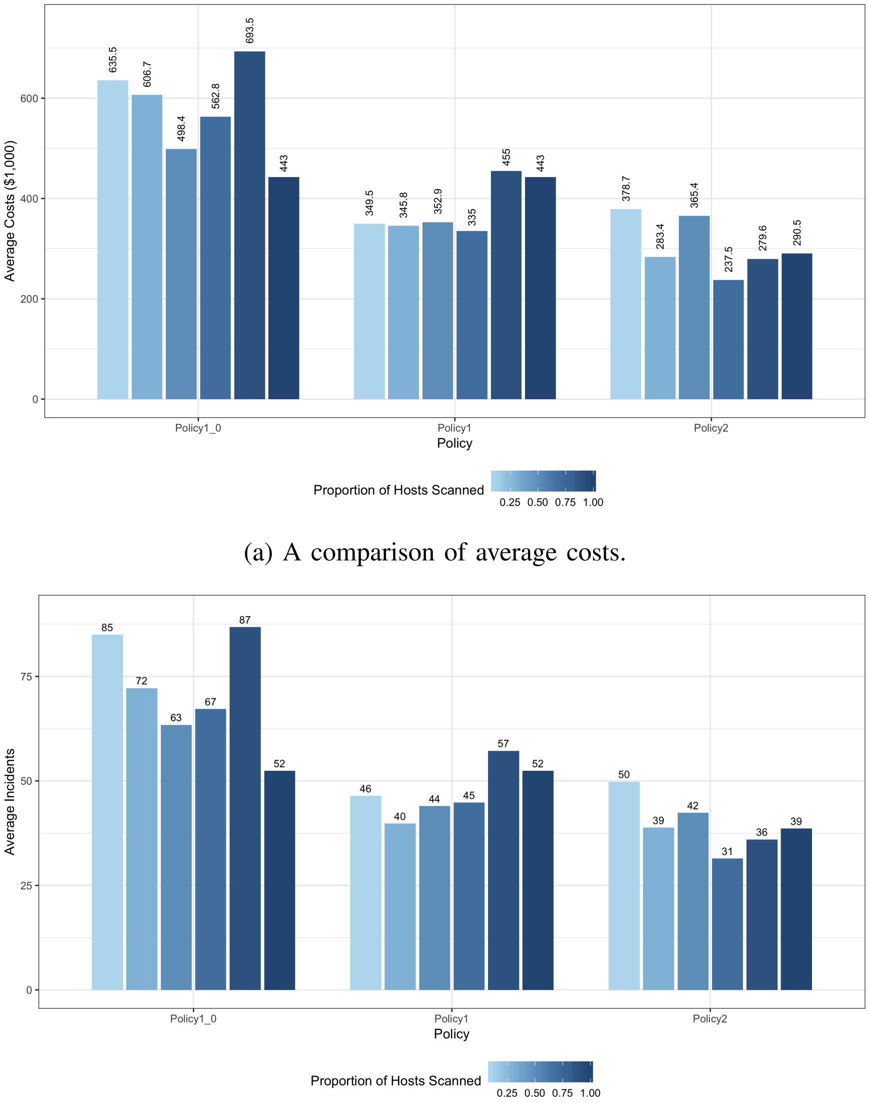
The most popular policy followed in organizations is called as "policy 1_0" at a scan level of 30% (Afful-Dadzie and Allen 2014; Allen and Liu 2018). (i.e.) For 30% scanned hosts, Action 1 is applied to hosts with Low and Medium severity vulnerabilities, and Action 2 is applied to hosts with High or Critical severity vulnerabilities and Action 3 for celebrity vulnerabilities. For 70% unscanned hosts, we follow the OSOM policy, i.e., do not take any manual actions.

Table 2: Cyber-maintenance policies explored (action for scanned hosts, action for unscanned hosts).

Vulns. Level\Policy	Current Policy 1_0	Policy 0	Policy 1	Policy 2	Policy 3	Policy 4
Low	(1,1)	(1,1)	(1,1)	(1,1)	(1,1)	(2,1)
Medium	(1,1)	(1,1)	(1,1)	(1,1)	(2,1)	(2,1)
High	(2,1)	(1,1)	(2,1)	(2,1)	(3,1)	(3,1)
Critical	(2,1)	(1,1)	(2,1)	(3,1)	(3,1)	(3,1)
Celebrity	(3,1)	(4,4)	(3,4)	(4,4)	(4,4)	(4,4)

The results of the simulation are illustrated in Figure 3 and Figure 4. Figure 3a and 3b explores the significance of celebrity vulnerabilities and improved analytics. The only difference between policy 1_0 and policy 1 is the action taken on celebrity vulnerabilities. At the currently observed 30% scan level, the associated maintenance costs are nearly halved from $606,000 to $345,800 by paying attention to celebrity vulnerabilities and the corresponding number of incidents goes down from 72 incidents to 40.

A similar trend is observed between policy 1 and policy 2. By choosing Action 3 (Research-compensate) on critical vulnerabilities, although the average number of incidents are almost same, policy 2 reduces the total expenditure even further by approximately $62,400. Figure 3a and 3b also shows that for the current policy 1_0, improving scanning efforts to observe 50% of the hosts can decrease the expenses of the by $108,300 with a 12.5% reduction in the average number of incidents.



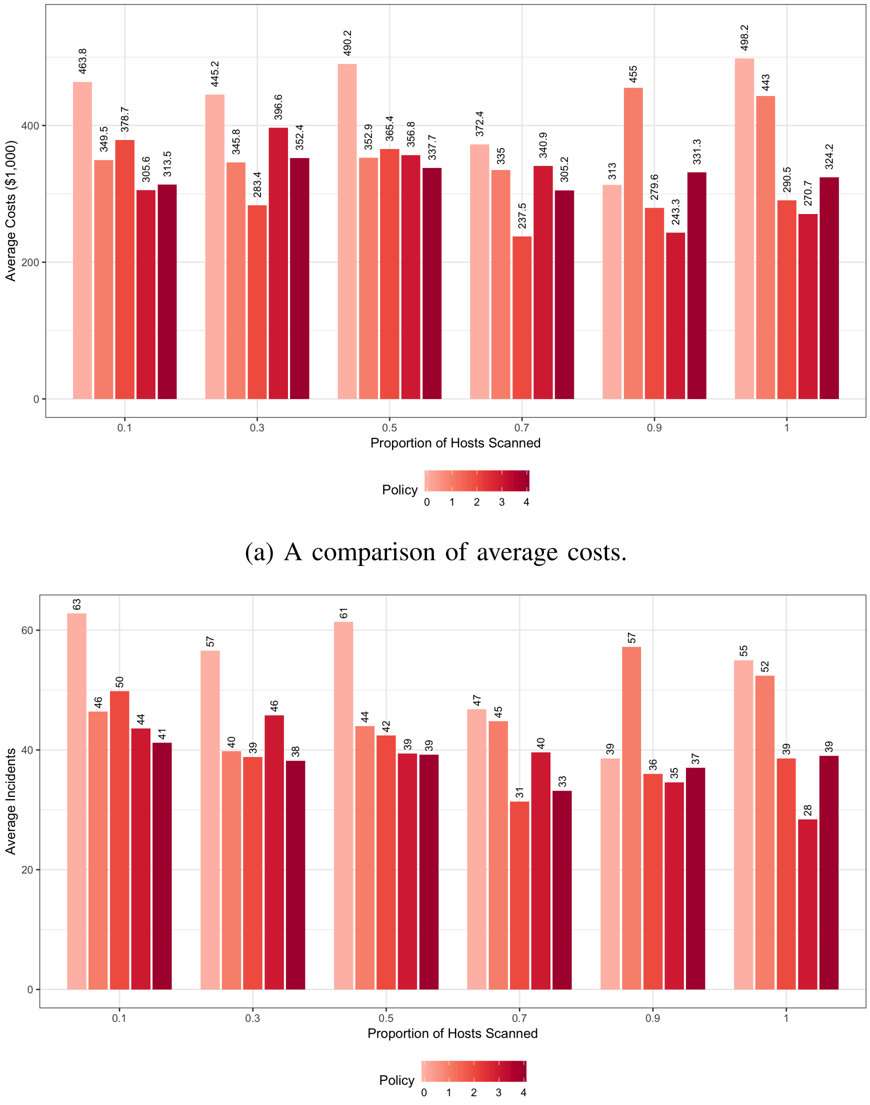
(a) A comparison of average costs.

(b) A comparison of average incidents.

Figure 3: A comparison among current policy and proposed policies.

Figures 4a and 4b provide a more comprehensive review of the policy benefits at different scan levels. While the best policy varies for each scan level, we can see that all the alternatives are more cost effective than the current OSOM policy 1_0. Specifically, policy 2 at a scan level of 70% provides the best objective results, with an average expense of $237,000 and 31 incidents on average. From Figures 3 and 4 we can also draw conclusions about policy 0. By comparing policy 0 and policy 2, we can see that its poorer performance compared to the other alternatives can be attributed to the penalty of not acting on high or critical severity vulnerabilities. It is still interesting to note that policy 0 performs significantly better than the current policy at the 30% scan level, with an associated cost of $445,200 and 57 incidents on average.

Therefore, it is evident that improving the current scanning methodology and scraping for evidence of celebrity vulnerabilities can drastically reduce the expenditure on cyber maintenance and improve security.



(a) A comparison of average costs.

(b) A comparison of average incidents.

Figure 4: A comparison from Policy 0 to Policy 4.

4 CONCLUSIONS AND FUTURE WORK

This paper puts forth a "nested birth and death" simulation model for predicting and forecasting of cyber maintenance costs. The article discusses the shortcomings of the current OSOM policy and scanning methodologies. Results from the simulation show that celebrity vulnerabilities are highly significant, and targeted actions on them can result in up to 50% reduction in cyber maintenance expenditure.

The article also proposes improved scanning methodologies and alternate policies for improving cyber security policies in organizations with resource constraints. The results of the simulation give the costs associated with cyber maintenance for multiple scan levels and policies chosen, and could be used to evaluate and improve current policies in organizations based on their available resources. It is observed that we can do better at the current scanning level using policy 2 which only costs one-third of the current estimated expenditure and reduces the number of incidents by 46%.

However, there are extrapolations in estimating the scaled down costs of actions and limitations with the simulation software. While the model proposed here addresses the concept increased scanning, the concept of optimal scanning can result in providing better policies at lower expenses. The optimal scanning could be derived from more sophisticated decision making models such as Markovian Decision Processes (MDPs). MDPs could be used to predict the probability of a particular host having high, critical, or celebrity vulnerabilities which in turn helps in improving scanning priorities and patching policies. Future work could include customized actions for different host types, improved costing methods with scanning and replacement costs, and dynamic pricing based on host and vulnerability attributes.

A APPENDICES

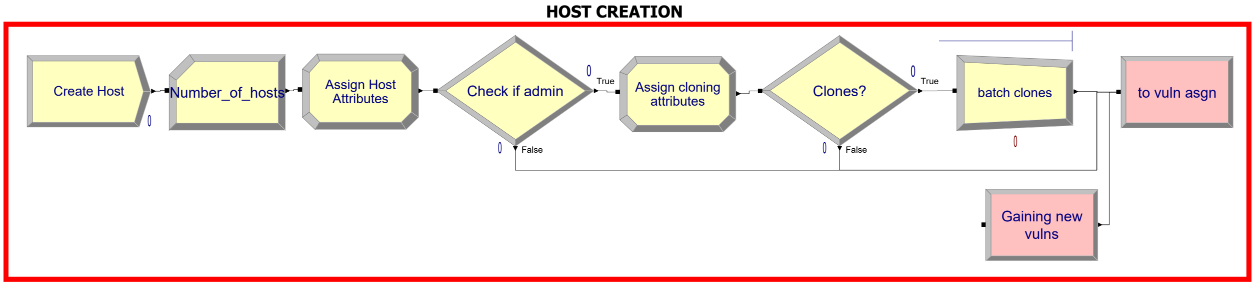


Figure 5: Host creation module.

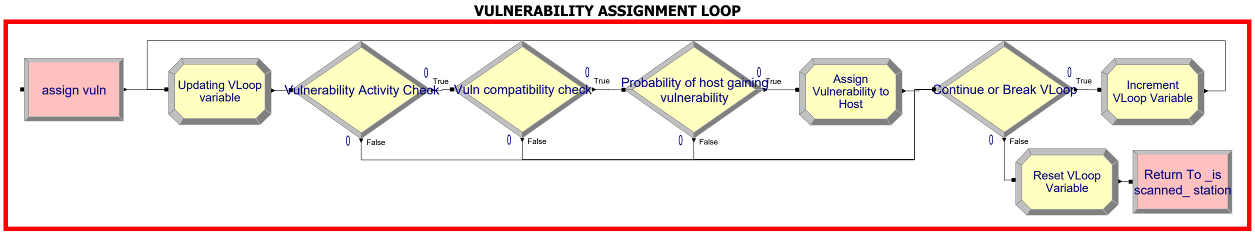


Figure 6: Vulnerability assignment module.

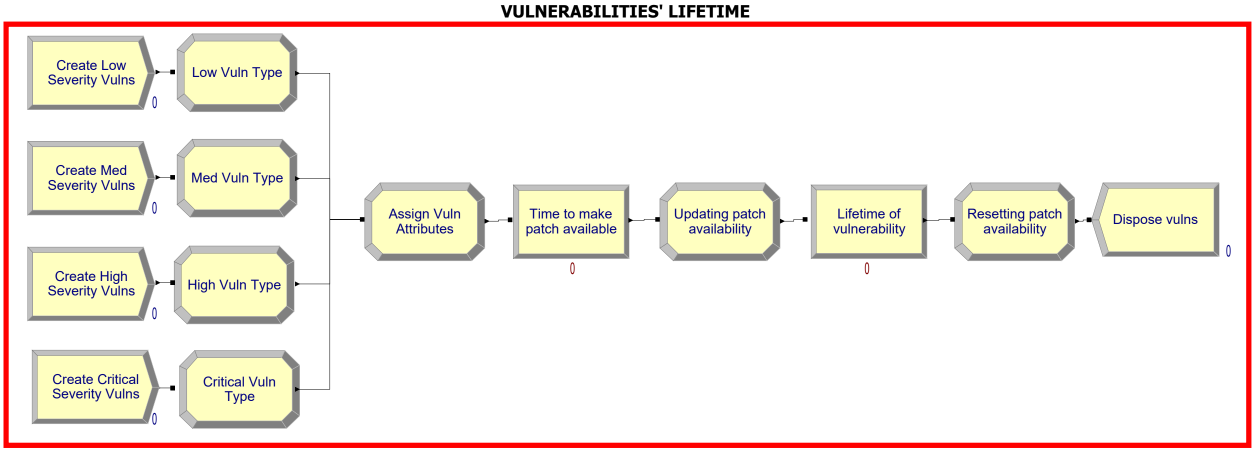

VULNERABILITIES' LIFETIME
Create Low Severity Vulns
0
Low Vuln Type
Create Med Severity Vulns
0
Med Vuln Type
Create High Severity Vulns
0
High Vuln Type
Create Critical Severity Vulns
0
Critical Vuln Type
Assign Vuln Attributes
Time to make patch available
0
Updating patch availability
Lifetime of vulnerability
0
Resetting patch availability
Dispose vulns
0

Figure 7: Vulnerability creation module.

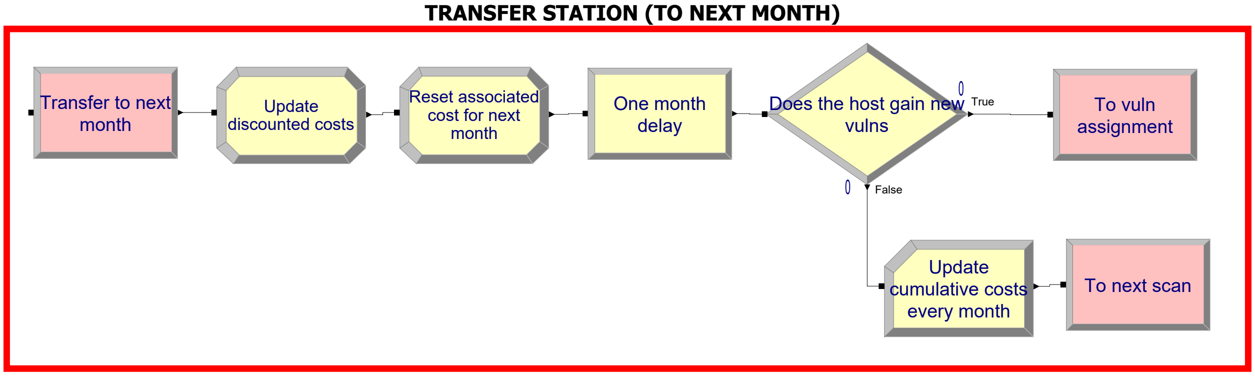

TRANSFER STATION (TO NEXT MONTH)
Transfer to next month
Update discounted costs
Reset associated cost for next month
One month delay
Does the host gain new vulns
0
True
To vuln assignment
0
False
Update cumulative costs every month
To next scan

Figure 8: Transfer station module.

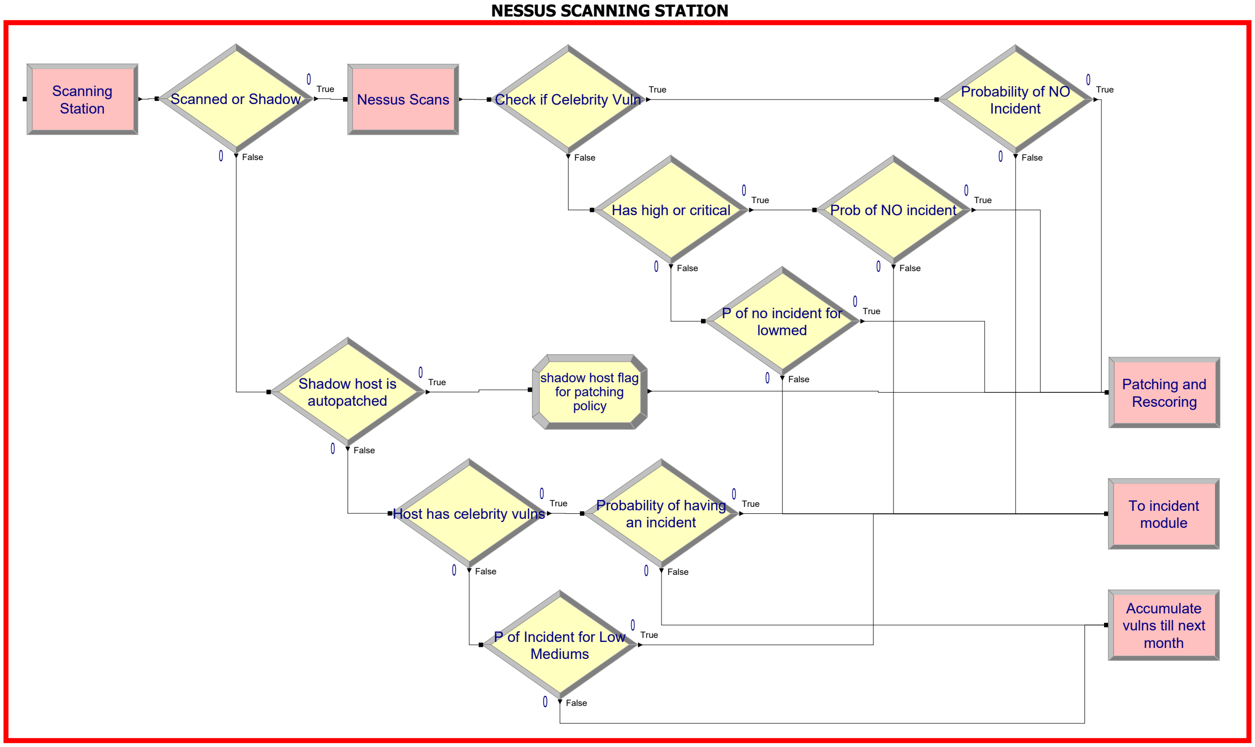


Figure 9: Scanning station module.

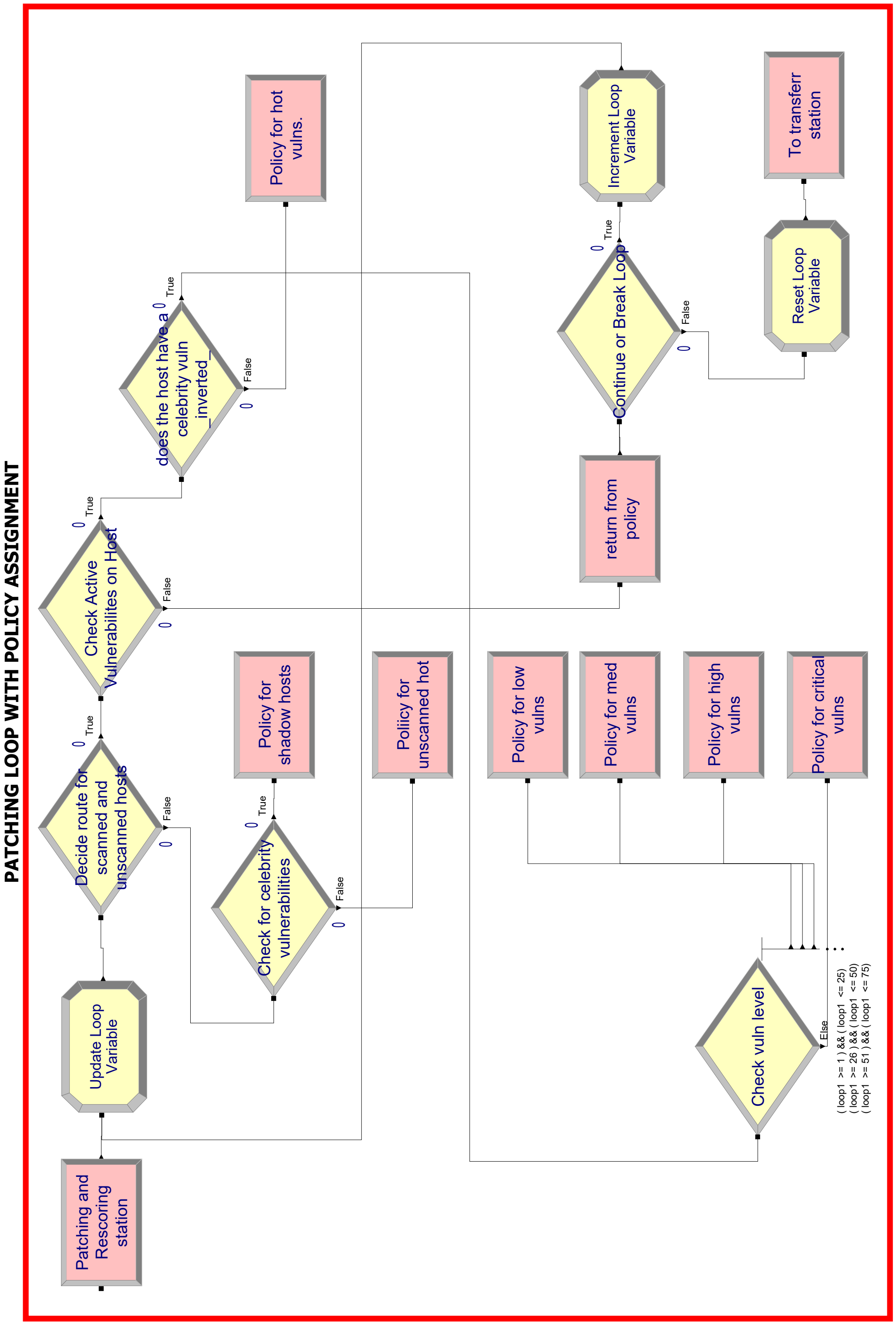

PATCHING LOOP WITH POLICY ASSIGNMENT
Patching and Rescoring station
Update Loop Variable
Decide route for scanned and unscanned hosts
True
False
0
0
Check for celebrity vulnerabilities
True
False
0
0
Policy for shadow hosts
Policy for unscanned hot
Check Active Vulnerabilites on Host
True
False
0
0
does the host have a celebrity vuln _inverted_
True
False
0
0
Policy for hot vulns.
Policy for low vulns
Policy for med vulns
Policy for high vulns
Policy for critical vulns
Check vuln level
Else
(loop1 >= 1) && (loop1 <= 25)
(loop1 >= 26) && (loop1 <= 50)
(loop1 >= 51) && (loop1 <= 75)
return from policy
Continue or Break Loop
True
False
0
0
Increment Loop Variable
Reset Loop Variable
To transferr station

Figure 10: Patching station module.

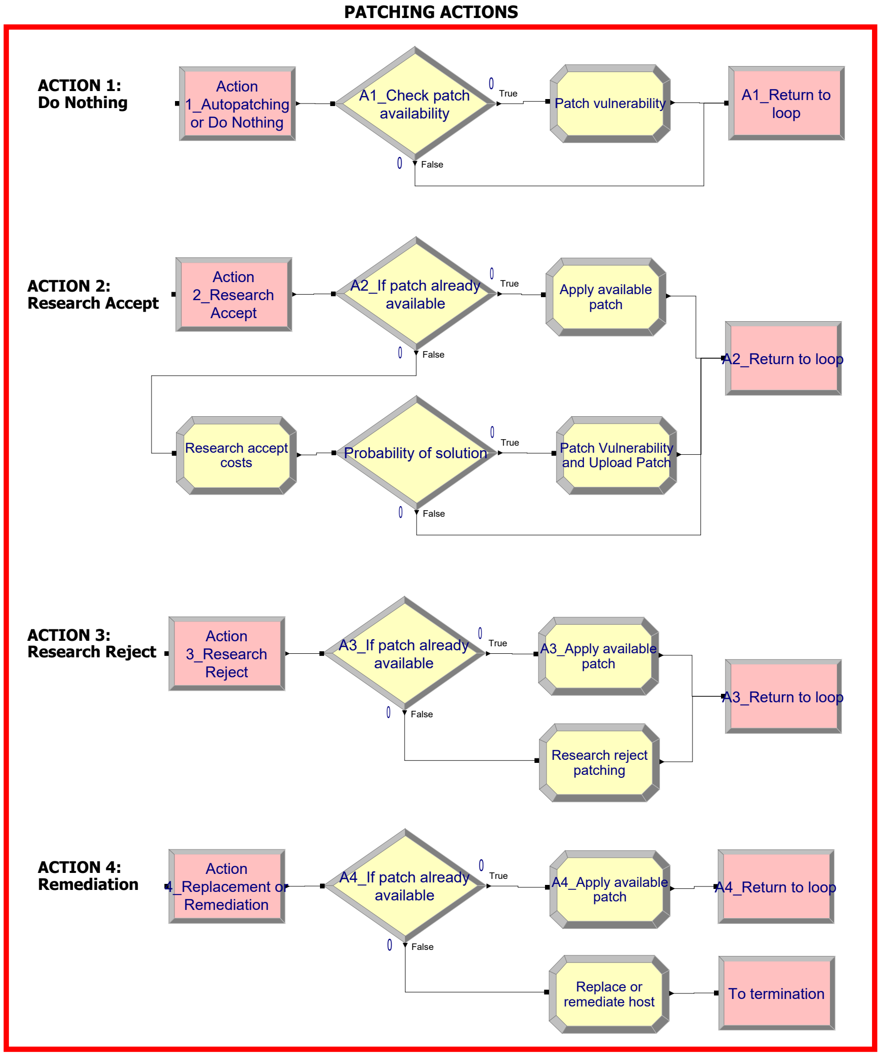


Figure 11: Patching Actions.

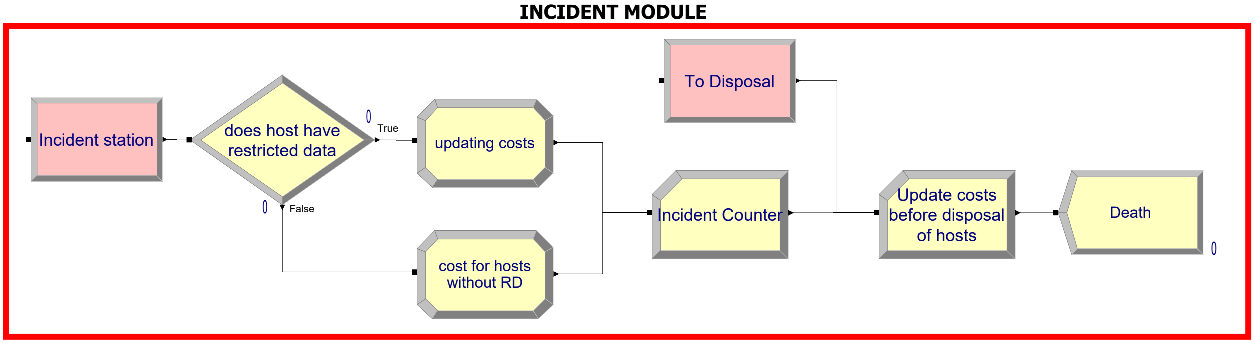

INCIDENT MODULE
Incident station
does host have restricted data
True
False
updating costs
cost for hosts without RD
To Disposal
Incident Counter
Update costs before disposal of hosts
Death
0

Figure 12: Incident module.

REFERENCES

Afful-Dadzie, A. and T. T. Allen. 2014. “Data-driven cyber-vulnerability maintenance policies”. *Journal of Quality Technology* 46(3):234–250.

Allen, T. T. and E. Liu. 2018. “Forecasting cyber maintenance costs with improved scan analytics using simulation”. In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 1218–1225. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Allen, T. T., S. Roychowdhury, and E. Liu. 2018. “Reward-based Monte Carlo-Bayesian reinforcement learning for cyber preventive maintenance”. *Computers & Industrial Engineering* 126:578–594.

Allen, T. T., Z. Sui, and N. L. Parker. 2017. “Timely decision analysis enabled by efficient social media modeling”. *Decision Analysis* 14(4):250–260.

Hernández-Suárez, C. M., C. Castillo-Chavez, O. M. López, and K. Hernández-Cuevas. 2010. “An application of queuing theory to SIS and SEIS epidemic models”. *Math. Biosci. Eng* 7(4):809–823.

Hou, C. 2015. *Dynamic programming under parametric uncertainty with applications in cyber security and project management*. Ph.D. thesis, The Ohio State University, Columbus, Ohio. https://etd.ohiolink.edu/pg_10?::NO:10:P10_ETD_SUBID:105526, accessed 9th August 2018.

Kephart, J. O., S. R. White, and D. M. Chess. 1993. “Computers and epidemiology”. *IEEE Spectrum* 30(5):20–26.

Mell, P., K. Scarfone, and S. Romanosky. 2007. *A complete guide to the common vulnerability scoring system version 2.0*. FIRST: Forum of Incident Response and Security Teams. https://www.nist.gov/publications/complete-guide-common-vulnerability-scoring-system-version-20, accessed 26th July 2018.

Naugle, A. B., M. L. Bernard, and I. Lochard. 2016. “Simulating political and attack dynamics of the 2007 Estonian cyber attacks”. In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 3500–3509. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Nguyen, C., J. Dietz, S. Liles, V. Raskin, J. Springer, and L. Yilmaz. 2015. “Cyber Defense Econometric of a power grid distribution infrastructure”. In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 978–1. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Ponemon Institute, LLC 2017. *2017 Cost of Data Breach Study: United States*. Ponemon Institute, LLC. https://www.ponemon.org/blog/2017-cost-of-data-breach-study-united-states, accessed 14th September 2018.

Roychowdhury, S. 2017. *Data-Driven Policies for Manufacturing Systems and Cyber Vulnerability Maintenance*. Ph.D. thesis, The Ohio State University, Columbus, Ohio. https://etd.ohiolink.edu/pg_10?::NO:10:P10_ETD_SUBID:150554, accessed 5th August 2018.

AUTHOR BIOGRAPHIES

AKSHAY KRISHNA MURALI is a M.S. student in the Integrated Systems Engineering department at the Ohio State University. He received his B.S. from SSN College of Engineering, Anna University in Chemical Engineering (2017). His interests are related to cyber security, operations research and analytics (murali.53@osu.edu).

ENHAO LIU is a Ph.D. candidate in the Integrated Systems Engineering department at the Ohio State University. He received his M.S. from the Ohio State University (2017), his B.S. from Jinan University in Electrical Engineering and Automation (2015). His interests related to cyber security, operations research, and reliability engineering (liu.5045@osu.edu).

THEODORE T. ALLEN is an Associate Professor in the Integrated Systems Engineering department at the Ohio State University. He received his B.A. from Princeton, his M.S. from UCLA, and his Ph.D. from the University of Michigan (1997). He is currently the president of the Social Media Analytics section of INFORMS and the simulation area editor of *Computers & Industrial Engineering* (IF: 3.2). He has published over 60 refereed publications and received over 30 grants as PI including from NSF, ARCYBER, and GE Appliances. His research on simulation optimization for voting machine allocation has received national attention and he has contributed to millions of voters avoiding hours of waiting and effective or actual law changes in North Carolina, Ohio, and Michigan. He has also served as associate editor for the *Journal of Manufacturing Systems* and *Quality Approaches in Education* and as a reviewer for *Operations Research*, *Technometrics*, and many other journals (allen.515@osu.edu).