11-2022

Optimizing Cybersecurity Budgets with AttackSimulation

Alexander Master
Purdue University, amaster@purdue.edu

George Hamilton
Purdue University, hamil132@purdue.edu

J. Eric Dietz
Purdue University, jedietz@purdue.edu

Optimizing Cybersecurity Budgets with AttackSimulation

Alexander Master, George Hamilton, J. Eric Dietz
Department of Computer and Information Technology
Purdue Homeland Security Institute
Purdue University - West Lafayette, USA
Email: {amaster}{hamil132}{jedietz}@purdue.edu

***Abstract*—Modern organizations need effective ways to assess cybersecurity risk. Successful cyber attacks can result in data breaches, which may inflict significant loss of money, time, and public trust. Small businesses and non-profit organizations have limited resources to invest in cybersecurity controls and often do not have the in-house expertise to assess their risk. Cyber threat actors also vary in sophistication, motivation, and effectiveness. This paper builds on the previous work of Lerums et al., who presented an AnyLogic model for simulating aspects of a cyber attack and the efficacy of controls in a generic enterprise network. This paper argues that their model is an effective quantitative means of measuring the probability of success of a threat actor and implements two primary changes to increase the model's accuracy. First, the authors modified the model's inputs, allowing users to select threat actors based on the organization's specific threat model. Threat actor effectiveness is evaluated based on publicly available breach data (in addition to security control efficacy), resulting in further refined attack success probabilities. Second, all three elements - threat effectiveness, control efficacy, and model variance - are computed and evaluated at each node to increase the estimation fidelity in place of pooled variance calculations. Visualization graphs, multiple simulation runs (up to 1 million), attack path customization, and code efficiency changes are also implemented. The result is a simulation tool that provides valuable insight to decision-makers and practitioners about where to most efficiently invest resources in their computing environment to increase cybersecurity posture. AttackSimulation and its source code are freely available on GitHub.**

***Index Terms*—budgeting, computer simulation, cybersecurity, data breaches, evaluation research, probability, risk management, threat modeling**

I. Introduction

Successful cyber attacks often cost organizations significant amounts of money and time. Data breaches, in which valuable non-public information is stolen and subsequently ransomed or published online, are an example outcome of a consequential cyber attack. IBM Security reported the U.S. average total cost of a data breach in 2021 to be 4.24 million dollars, an increase from 3.86 million in 2020 [1]. While mass media tends to focus on incidents involving large corporations, the cost can be proportionally severe for small businesses with fewer than 500 employees. Compared to the previous year, the average cost of a data breach for small businesses grew 26% in 2021 to 2.98 million dollars per incident [1]. The increased proliferation of many aspects of our lives relying on Internet services presents new opportunities for malicious online activity from criminals, state-affiliated advanced persistent threats (APTs), activists, and competitors. At the same time, new cybersecurity firms and security software solutions are increasingly being brought to market. Despite a 58% increase in U.S. corporate spending on cybersecurity over the last five years, 2021 in the U.S. saw 1862 reported data breach incidents [2], [3]. Organizational leaders face the increasingly difficult choice of where to invest their limited resources to reduce cybersecurity risk.

II. Review of Relevant Literature

Academic literature from Gordon and Loeb describes methods for optimizing security budgets for information technology systems [4]. Lerums et al. present an attack simulation tool for modeling the effectiveness of several security controls against phishing attacks [5]. Work from David Bianco presents a taxonomy for understanding the challenges of identifying cyber attacks [6]. The Verizon Security Research Team maintains the VERIS Community Database (VCDB), a corpus of security incidents and data breaches from 2012 to the present [7]. The authors considered all of these past works and how they might contribute to an improved method for small organizations hoping to better understand their cybersecurity posture.

A. Frameworks For Maximizing Security Budgets

Existing research provides several approaches for maximizing the effectiveness of a limited cyber security budget. First, a foundational work in the field from Gordon and Loeb described a simple mathematical model for calculating the expected net benefit of a security control based on its cost and reduction of expected loss. Their research also included performing statistical analysis on the output of their formulae to arrive at some generalized rules about security spending best practices, including that expenditure should generally be less than or equal to 37% of expected loss [4]. Businesses can then use the formulae to calculate the expected net benefit of each security control being considered and rank them accordingly.

Later research expanded on Gordon and Loeb's initial findings by modifying the formulae to incorporate the "wait and see" option and other considerations. The "wait and see" option represents the option of a business to wait for a specified time period before deciding whether or not to invest in a control - in hopes that at a later time there will be less uncertainty about the relevant threats and controls, resulting in a more accurate prediction of potential loss and mitigation [8].

DOI: 10.1109/HST56032.2022.10024984

In addition, researchers expanded on Gordon and Loeb's initial equations by creating a series of equations to maximize the overall expected net benefit, given information on the relevant threats and security controls being considered. Businesses would use the model by entering the likelihood and cost of threats they may face, the cost of each control being considered, and the probability it would mitigate a given threat. Next, the model would determine which controls represent the highest estimated net benefit per dollar [9]. However, this research provided pseudocode, not a functioning program.

The existing frameworks described offer equations, with some providing pseudocode, but leave implementation to businesses or later researchers. They rely on organizations with access to subject matter experts and significant time available to create comprehensive, individualized estimates of likelihood and cost of potential incidents - as well as estimating efficacy and cost of potential controls. These estimates then serve as the inputs for the equations and pseudocode described [4], [8], [9].

B. AnyLogic Modeling

One solution to help organizational leaders prioritize where they spend funds to mitigate cybersecurity risk without dedicating large amounts of time and resources is to model the threat. Analysis done through models and simulations can visually illustrate to technologists and executives the locations of organizational assets, potential courses of action an adversary may take, and mitigations to deter those actions. AnyLogic® is computer simulation software used to identify and propose solutions to problems across several industries. The software is unique in that it offers discrete event modeling, agent-based modeling, and system dynamics approaches to computational models [10]. Once the model is built, inputs to the system such as object (or agent) behavior can be modified to specific scenarios or use-cases. Simulations can be run hundreds (or thousands) of times in succession, with variance in each run and results on the overall system recorded. This allows organizations to rapidly gather data and optimize a key output on a given system. AnyLogic has walk-through demo models to showcase the flexibility of their software in a variety of industry settings [11].

C. Existing Attack Simulation

In 2018, Lerums et al. presented a paper and associated AnyLogic model at the IEEE International Conference on Electro/Information Technology (EIT) in Rochester, Michigan USA [5]. The agent-based model they presented simulates a cyber attack, with the primary example throughout the paper being a phishing attempt. The paper demonstrates a phishing attack's probability of successfully moving through each machine, NetworkNode, in a victim's environment to reach a target machine called the "flag". The flag may represent the primary domain controller of the network, which commonly holds the most sensitive information, or any other machine deemed most important to the victim. The model also displays the cost and effectiveness of several possible controls with realistic default values based on research from Lerums et al. [5].

As presented, the model effectively generalizes a small enterprise network and many of the most common components necessary for daily operations in modern organizations. It utilizes AnyLogic agents of the type "NetworkNode" to represent the devices along the physical and logical paths that an attack must traverse before compromising a machine with the organization's most sensitive information, referred to in this paper as an attack path. In the original model the attack path is presented left to right and top to bottom as: the organization's email server (Exchange Server), the end user's computer (Workstation), a layer 3 router (Router), a departmental network segment, an internal network segment, and the domain controller (flag). At each node, the model allows for various inputs, including mitigation measures, referred to in this paper as controls, such as the presence of a firewall, an intrusion detection system, email scanning software, or antivirus solution. The model allows for manual input of the probability and variance of each control stopping the attack at each NetworkNode. This customization is significant because it creates the flexibility to compare the efficacy of different controls and adapt the model as future research and better data emerge. Additionally, each control has a cost field which can be accounted for by service subscription, one-time fee, or cost per user/platform. Nodes can be turned off for a specific environment's simulation or cost values set to zero if the intent of a particular simulation run is to show a proposed additional control, not overall operational cost [5].

Lerums et al. posit that phishing attacks are one of the most effective ways for a threat actor to gain an initial "foothold" into a victim's network and are often revealed to have been successful in data breach investigations [5]. The National Institute of Standards and Technology (NIST) defines phishing as the "use of deceptive computer-based means to trick individuals into disclosing sensitive personal information." [12]. While adversarial methods vary, email is one of the most common delivery mechanisms for phishing attacks. Lerums et al. simulated a phishing email attempt using their model and a set of proposed controls based on their research of the efficacy of cybersecurity tools in defense against phishing emails. A successful attack is represented when each node's security control is bypassed; nodes are highlighted in red when the phishing email bypasses all of the node's controls. For their simulation, it was assumed that a software "backdoor" would be installed on the end-user victim's machine, resulting in the compromise of at least one system in the network. The overall probability is calculated by the product of the probability of an attack bypassing each control at each node; overall cost is the addition of all costs associated with each control at each node. Lerums et al. concluded when the model was run 150 times (n = 150) with the control attributes based on their research, attacks were 0% effective at the internal network node and thus never reached an end-user workstation [5].

The AnyLogic model presented by Lerums et al. succeeds in providing a framework to simulate and measure the effective-

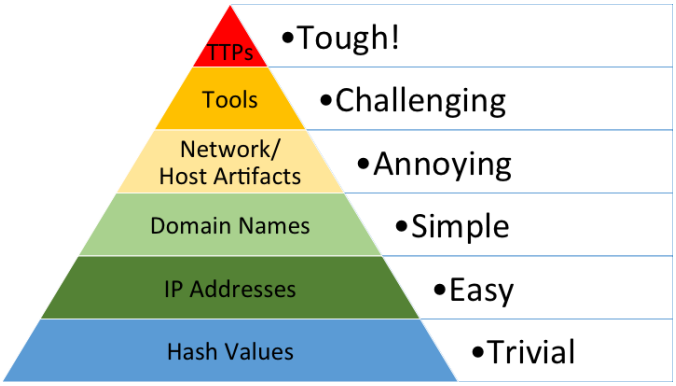

Fig. 1. The Pyramid of Pain visualizes difficulty levels in detection of cyber attacks [6]

ness of cybersecurity controls against a cyber attack technique while also depicting costs associated with their implementation. Simulation runs give an organization's technicians and leaders insight into where to prioritize resources to reduce cybersecurity risk [5]. The additions and improvements that follow build on this existing work. The research goals were to ensure that the model outputs were appropriately tailored and accurate, and that the inputs and assumptions made by the simulation were inclusive of current adversarial understanding.

D. Challenges In Threat Analysis

The Pyramid of Pain by David Bianco provides a conceptual model of the difficulty posed by aspects of cyber attacks, shown in Fig. 1 [6]. Starting from the bottom of the pyramid are portions of analysis that are easy for humans and software alike. Sorting through lists of hashes and internet protocol (IP) addresses to determine whether something is malicious is generally trivial but often has little value in determining what happened. These aspects of cyber attacks are also easy for threat actors to modify to thwart defenders relying on them. Threat actors can recompile their tools using obfuscators or adding randomness when compiling them to ensure cryptographic hashes of their tools (if discovered) do not match blocklist databases or crowd-sourced threat feeds like those available on virustotal.com. They can also use virtual private networks (VPNs) or proxy servers to redirect their traffic, ensuring attribution to their IP addresses is difficult [13].

The top of the pyramid addresses the more challenging aspects of the cyber kill chain. Tactics, Techniques, and Procedures (TTPs) of adversaries are often the most cost-intensive resource investment made by an adversary. Subject matter expertise, cost of cyber-physical and logical infrastructure, and development of tactics and techniques make formidable forces for defenders. These aspects of attack methodology are also the most difficult to detect and protect against for defenders. Unfortunately, many commercially available cybersecurity products only address the bottom four categories of the Pyramid of Pain, leaving the difficult job of hunting for malicious cyber actors to human analysts [2]. It is important to consider that not all organizations have dedicated security staff or security operations center (SOC) to facilitate such analysis.

E. Shortcomings of The Existing Attack Simulation

The notion of reducing the risk of a cyber attack down to zero percent is intuitively misleading, given the number of data breaches and ransomware incidents publicly disclosed and media coverage surrounding them. The authors do not take issue with the probability calculations of the Lerums et al. model but rather assumptions leading to some of the inputs [5]. Solely using efficacy data provided by vendors of cybersecurity products as a baseline for probabilistic estimates creates a monolith of all threat actors and likely skews the results towards that of less sophisticated, high-volume attacks.

Building on the phishing example of Lerums et al., low-budget scammers will have vastly different levels of success than a state-affiliated APT group. A scammer TTP may include sending out large volumes of phishing emails to various targets, exploiting whoever they can for monetary gain. If this is the threat model a particular organization faces, the technical controls and attack effectiveness probabilities represented in Lerums et al.'s original AnyLogic model may be sufficient in reducing the number of successful attempts to near-zero probability [5]. However, an APT group's TTPs will likely include specific targeting of the email recipient and tailoring the message's content to increase the likelihood of the target believing the message to be genuine. APTs will also likely utilize advanced techniques to bypass the victim network's technical controls.

For example, some email server administrators, such as those in the Department of Defense, utilize software that strips hyperlinks out of all emails to reduce the likelihood of malicious links making their way to user inboxes. Some systems even filter content against large lists of known command and control servers or malware domains. However, in the case of hyperlink stripping, this inadvertently trains users to copy and paste the plaintext hyperlinks they receive in their emails. If an APT is armed with this information, given its advanced resources, it could carry out a phishing campaign. It would begin by acquiring new legitimate web domains (defeating blocklists), utilize plain-text links to convincing URLs with HTTPS redirects with legitimate TLS certificates (users will copy the phishing URLs into their browsers), and ultimately defeat the technical mitigations in place. Threat actors with a higher probability of attack success, such as those with advanced techniques described here, should be accounted for in cyber attack simulation modeling.

While reporting on data breaches covers many different industries, detailed data resulting from the investigations of these attacks is often limited [3]. Government and military breaches may encounter classification issues, healthcare breaches may involve personally identifiable information (PII) or legal ramifications involving HIPAA, and specific industries with proprietary data concerns result in less publicly available information for study. Despite these limitations, some groups prioritize attribution of threat actors in incident reporting, such

as the Verizon Security Research Team. The data scientists at Verizon maintain a corpus of incidents and breaches, which informs the annual Data Breach Investigations Report (DBIR) [14]. Their data set can provide insight into threat actor effectiveness for the purpose of simulation modeling.

The model presented by Lerums et al. is narrowly scoped in the attacks it seeks to simulate. Specifically, it considers a single phishing attack per iteration that travels through a fixed set of NetworkNodes before reaching the most important machine in a victim's network, the "flag" [5]. The model could be made more robust by adding additional NetworkNodes commonly observed in breaches, including web application servers which appeared in 56% of all breaches in Verizon's 2022 DBIR [14]. The existing NetworkNodes in the simulation could also be made more useful by allowing users to re-arrange them to represent an attack path specific to their own network. Finally, the existing model could be improved by providing users with aggregate statistics based on a large number of attacks, such as the frequency of attacks the user's organization is likely to face in a year, rather than presenting results one attack at a time.

III. Methods

The authors sought to improve the model presented by Lerums et al. by incorporating relevant threat actors as an influence on security control efficacy. A model should provide users flexibility in the attack paths they can simulate, accurately reflect the kinds of NetworkNodes that exist in modern networks, and present users with aggregate results based on many attack iterations.

A. Threat Actor Data Analysis

The authors incorporated the influence of a threat actor by using two different rates derived from data in the VERIS Community Database (VCDB) [7]. First, the proportion of all incident reports in which the victim was a small business[1], which involved a given threat actor (a_p). The authors label this proportion "prevalence rate", which is based on (n) the number of incident reports involving that threat actor where the victim was a small business and (N) the total number of incidents where the victim was a small business: $a_p = \frac{n}{N}$. The authors defined success for a threat actor as the actor being part of an incident with confirmed data disclosure, and failure as the threat actor being part of an incident with a confirmed lack of data disclosure. Incidents coded with "unknown" or "maybe" values for data disclosure were excluded. A threat actor's "fail rate" (a_f) is represented as the inverse of the threat actor's "effectiveness", and was calculated based on the number of (s) successful and (f) failed incidents from the perspective of the threat actor: $a_f = \frac{f}{s+f}$.

After determining the prevalence rates (a_p) and fail rates (a_f) of threat actors, the results were filtered and categorized for applicability to the model and clarity to the end-user. Threat actors with minimal perceived impact on small businesses, such as "Unknown," were removed. The model's primary focus is cyber attacks perpetrated by external actors, for which cybersecurity products are primarily designed to prevent or reduce harm. Given this constraint, many actors considered "insider threats" were also removed from consideration. Examples in the VCDB include "Doctors," "Guard," "Cashier," "Executive," "Human Resources," and others, all of whom would have at least privileged access to their organization's computer systems. Additionally, some threat actors were excluded because the authors determined the sample size too small to be representative; the "Terrorist" threat actor is one such example (n = 3).

In addition to exclusion of some threat actors, the authors made the decision to group data for similar actors. "Nation-State" and "State-Affiliated" were combined into a single group called "State-Affiliated". A combined prevalence rate (a_{pc}) was computed from the prevalence rate of the "State-Affiliated" (a_{ps}) and "Nation-State" (a_{pn}) threat actors using the following formula: $a_{pc} = a_{ps} + a_{pn}$. The combined threat actor fail rate (a_{fc}) was computed using the number of incidents in which the threat actor failed and succeeded in breaching data for both "State-Affiliated" (f_s and s_s respectively) and "Nation-State" (f_n and s_n respectively): $a_{fc} = \frac{f_s+f_n}{(f_s+s_s)+(f_n+s_n)}$. The results of the overall threat actor analysis are shown in Table I.

There are limitations inherent in this approach. The model assumes the accuracy of VCDB incident reporting. The VCDB maintainers verify incident reports, but the majority are self-reported or submitted by the security community. The AnyLogic model inherits any data bias present in VCDB. Under reporting may be present given the large variety of jurisdictions represented in the data, some of which do not have mandatory reporting requirements. Skew toward particular industries may also occur. Data collection over many years (VCDB events include reports from 2012 to the present) may help mitigate some bias issues. Nonetheless, these proportions provide valuable insight into threat actor activity – and the model can easily be given other input data as adversarial understanding increases.

Algorithm 1 Selected Threat Actor Data Incorporation

$appliedFailRate$ = 1
for Each Selected Threat Actor Accessed In Random Order **do**
  $actorIsApplied$ = $True$ with probability a_p, otherwise $False$
  if $actorIsApplied$ **then**
    $appliedFailRate$ = a_f
    Break
  end if
end for

B. Threat Actor Model Additions

After performing the necessary analysis to determine threat actor prevalence and fail rates, the data were incorporated into

[1]The DBIR defines a small business as an organization with 1000 or fewer employees. [14]

Algorithm 2 Relay Attack Algorithm

$totalCost$ = 0
$probabilityNodeCompromised$ = 1
for Each Control **do**
  $totalCost$ += cost of this control
  $controlProbability$ = probability control stops the attack
  $controlVariance$ = variance in the calculation of
  $effectiveVariance$ = random number [-$controlVariance$, $controlVariance$]
  $effectiveControlProbability$ = $controlProbability$ + $effectiveVariance$
  $effectiveControlProbability$, constrained to [0, 1]
  $probabilityNodeCompromised$ *= 1 - $effectiveControlProbability$
end for
$probabilityAttackStopped$ = (1 - $probabilityNodeCompromised$) * $appliedFailRate$
$probabilityAttackStopped$, constrained to [0, 1]
$probabilityNodeCompromised$ = 1 - $probabilityAttackStopped$
$nodeIsCompromised$ = True with probability $probabilityNodeIsCompromised$, otherwise False
Update running statistics tracking number of compromises for this node
if $nodeIsCompromised$ **then**
  Display this node as red, indicating it was compromised
  Relay the attack to the next node
end if

the model by allowing users to indicate relevant actors in a simulation run with a set of checkboxes (see Fig. 2b). Threat actors represent a new AnyLogic agent type within the model. A population of all relevant actors is created at runtime from an Excel sheet containing each threat actor's prevalence rate and fail rate. This allows users to update or modify threat actor data at runtime without requiring an AnyLogic license or rebuilding the model. For each attack simulated, a fail rate is applied based on the user-selected threat actors using Algorithm 1.

Algorithm 1 begins by random selection of an actor among those the user has chosen to include. The prevalence rate of the algorithm-selected actor is considered; if chosen that actor will be relevant for the particular attack iteration. If not, another actor will be assessed based on their prevalence, and so on. If no selected actor is identified based on prevalence, "Unidentified" will be reflected for that iteration. The output of algorithm 1 is $appliedFailRate$ and serves as a modifier to control efficacy.

After $appliedFailRate$ is determined, the attack begins at the first node based on the position values for each node entered by the user. Algorithm 2 occurs at the node, where the next node represents the subsequent node in the attack path based on position values for each node entered by the user (see Fig. 2a).

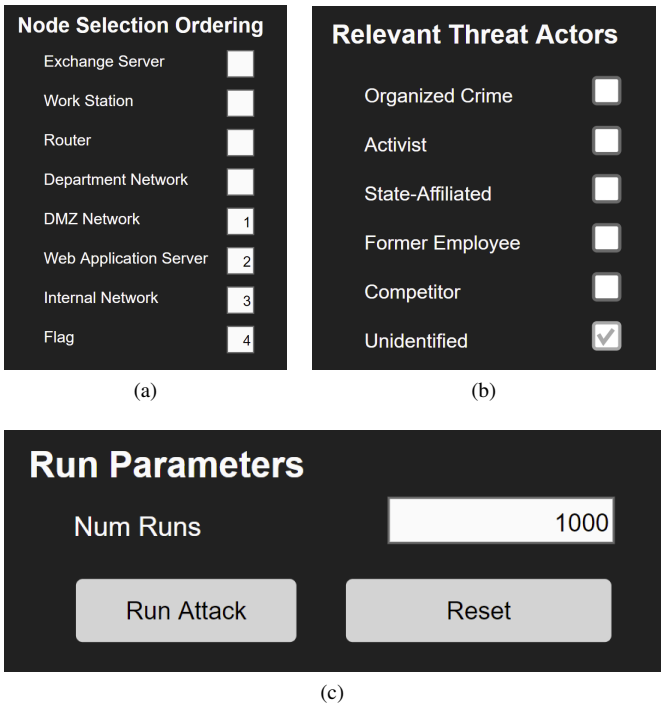

Fig. 2. Updated (a) node ordering; (b) threat actor selection; (c) run configuration menus

C. Alternate Attack Paths and New Nodes

The original model from Lerums et al. allowed users to simulate a single attack path, in which the attack was relayed from one node to the next with a fixed set of nodes in a fixed order [5]. This was an effective strategy for simulating a phishing attack in a single network topology, given the assumption that a phishing attack would traverse NetworkNodes in a specific order. However, it is limiting when simulating arbitrary attacks or varied network topologies. To increase flexibility, the authors implemented a position selection feature, which allowed users to designate the NetworkNodes in the simulated attack path and the order in which they appeared. Algorithm 2 represents the progression of the attack through each node in the AnyLogic code, and the impact threat actor effectiveness,

control efficacy, and variance have on attack outcomes.

D. Result Aggregation and Visualization

The model was expanded to allow users the option to simulate a specified number of attacks up to 1,000,000 with a single click and an unlimited number of attacks by repeating the process. Bar charts are included to demonstrate attack outcomes and success probabilities of security controls over many iterations (see Fig. 2c). The model also includes a reset button which is used to clear all aggregate data and graphs to run subsequent attacks from a clean state.

E. Retention

The attack simulation model maintains several core components from the original created by Lerums et al. While the NetworkNode agent's inner code components were changed significantly, the user-facing appearance of the NetworkNode remains highly similar. In addition, the controls available for user testing and how they enter the controls' probabilities, variances, and costs remain the same. Lastly, the fundamental idea of a simulated attack path, in which an attack begins at one node and traverses to another until a control stops the attack, remains unchanged.

IV. Analysis and Results

The authors completed the additions and modifications using the methods described in section III and shown in Fig. 2. The result was a simulation model which allowed users to easily configure custom attack paths by assigning each NetworkNode a position of their choosing within the path they wanted to simulate, as shown in Fig. 2a. This also increased the number of unique attack paths which could be simulated by the user from 1 to $\sum_{i=1}^{d} i! = \sum_{i=1}^{8} i! = 46233$, where d was the number of available NetworkNodes (8 in the updated model). While some of these attack paths are unlikely in real-world network architectures, the large number of available configurations allows users to make their simulation more representative of their networks.

Results also included the successful incorporation of threat actors into the model. The data controlling each threat actor's impact on probabilities is loaded from an Excel sheet at runtime. Users may modify or update this data as they see fit. The data shown in Table I is populated into the model by default; this threat actor effectiveness analysis was conducted as described in section III and used the VCDB events from 2012 to February 2022.

Threat actors applied to a given run of the simulation may be tailored to fit the organization's threat model more appropriately. This is accomplished using the relevant threat actor selection controls added to the model shown in Fig. 2b.

The model was also successfully restructured to allow simulating a number of attacks specified by the user, rather than a single attack. The user could configure this by setting run parameters to specify the number of attacks to simulate, and reset aggregated statistics to run the subsequent attack from a clean state as shown in Fig. 2c.

TABLE I
THREAT ACTOR PREVALENCE AND FAIL RATES: VCDB ANALYSIS
(DEFAULT SIMULATION RUNTIME VALUES)

Actor	Fail Rate	n	N*	Prevalence Rate	n	N**
Organized Crime	0.182	570	7283	0.131	375	2861
Activist	0.53	447	7283	0.0308	88	2861
State-Affiliated	0.1051	257	7283	0.00664	19	2861
Former Employee	0.0469	64	7283	0.0136	39	2861
Competitor	0.125	16	7283	0.00245	7	2861

*Population for Fail Rate utilizes the entire VCDB data set
**Population for Prevalence Rate includes only incidents pertaining to organizations with 1000 or less employees
Note: Fail Rate is the inverse of the *p*-value threat actor effectiveness

A set of graphs were created to visualize the results of many attack iterations. The first displays the percentage of attacks that compromised a NetworkNode. The second displays the number of iterations in which each NetworkNode was compromised. An example of output graphs is shown in Fig. 3.

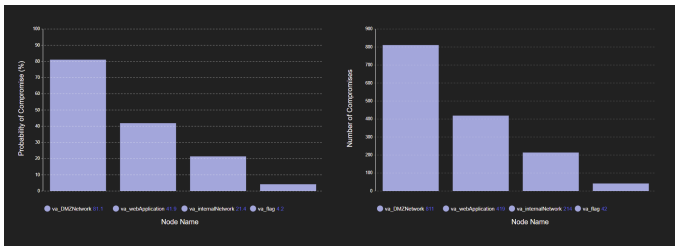

Fig. 3. Graphs to visualize aggregate statistics across all attack runs

The authors successfully added the web application NetworkNode as discussed in section III, and a DMZ node to represent the additional network segment in which Web Applications are often placed by businesses. For comparison, the original model presented by Lerums et al. is shown side-by-side with AttackSimulation in Fig. 4.

This model is intended for small organizations without dedicated cybersecurity teams, or anyone trying to make more informed risk decisions while allocating a limited cybersecurity budget. The model will not represent every organization with pinpoint accuracy; it is meant to provide a useful tool accurate enough to improve decision-making [15].

All AttackSimulation code and input data have been made open source, and are freely available at https://github.com/gjhami/AttackSimulation/. Standalone executable versions of the model, which do not require the AnyLogic software or license to run, are available under the "Releases" page. Additionally, a python parser script is included in the repository to aid in the automation of VCDB analysis.

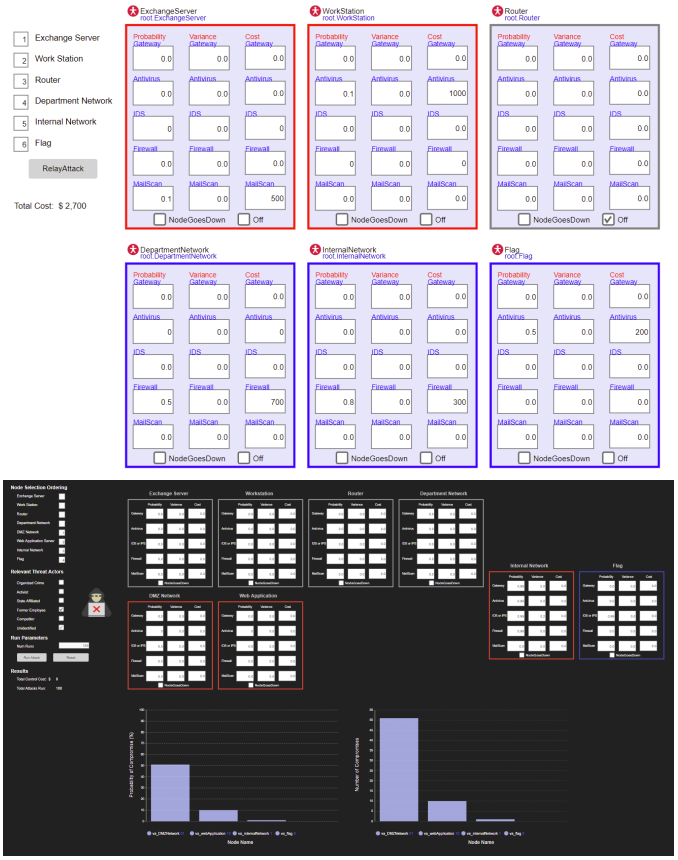

Fig. 4. (Top) Lerums et al. Model (Bottom) AttackSimulation

V. Conclusion & Future Work

Cybersecurity threats are an increasingly prominent aspect of the risk management decisions made by organizations. Businesses have a responsibility to secure the consumer data with which they are entrusted. If not adequately mitigated, cyber attacks can result in significant detrimental outcomes, such as data breaches. Small businesses and non-profits have limited resources and need an effective means to prioritize where to invest them - from among the myriad of cybersecurity solutions available on the market today. Simulation modeling can be a cost-effective means for organizations to quantitatively assess their cybersecurity posture, especially when their limited budget makes more in-depth assessments cost-prohibitive. The model presented in this paper builds on the existing literature of cyber attack simulations. The model assists with the visualization of probabilities of attack success and compares the efficacy of security controls at various levels of a computer network. The authors' contributions include: allowing user input for threat actor selection, implementation of threat actor effectiveness scoring in the probability of attack success, the addition of web application nodes, node re-ordering functionality, automation of simulation iterations with visualization of results, and numerous code efficiency improvements.

Regarding future work, the authors invite researchers and industry users to provide feedback or contributions to the project on its GitHub page. Input from small businesses and users testing the model in their organization is encouraged and welcome. Several lines of research effort, as well as new features for implementation into the model, would be helpful towards increasing the veracity and scope of attack simulations. Specific recommendations are as follows:

1) Implementation of default data importing for security control efficacy, security control variance, and security control cost from comma separated value files at runtime
2) Integration of data related to overall costs of data breaches, to allow for return on investment (ROI) predictions, as well as user-imposed budget constraints
3) Further research into security control efficacy rates from a broader range of cybersecurity products
4) Expand the model to allow multiple instances of each NetworkNode to simulate more complex and layered network topologies

Acknowledgment

The authors are grateful to Dr. James Lerums for his collaboration and sharing the source code of the model this project was based on. PHSI thanks The AnyLogic Company for providing software licensing to enable our research.

References

[1] I. Security, "Cost of a data breach report," tech. rep., IBM, 2021. https://www.ibm.com/security/data-breach.
[2] J. Hubback, "Cybersecurity technology efficacy report: Is cybersecurity the new "market for lemons"?," tech. rep., Debate Security, October 2020.
[3] ITRC, "End-of-year data breach report," tech. rep., The Identity Theft Resource Center, January 2021. https://www.idtheftcenter.org/.
[4] L. Gordon and M. Loeb, "The economics of information security investment," *ACM transactions on information and system security*, vol. 5, no. 4, pp. 438–457, 2002.
[5] J. E. Lerums, D. P. La'Reshia, and J. E. Dietz, "Simulation modeling cyber threats, risks, and prevention costs," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pp. 0096–0101, IEEE, 2018.
[6] D. Bianco, "The pyramid of pain." https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html, 2014.
[7] Verizon Security Research Team, "The veris community database." https://github.com/vz-risk/VCDB, 2022. Commit Used: c1af86f on February 23, 2022.
[8] L. A. Gordon, M. P. Loeb, and W. Lucyshyn, "Information security expenditures and real options: A wait-and-see approach," *Computer Security Journal*, vol. 19, no. 2, 2003.
[9] T. Sawik and B. Sawik, "A rough cut cybersecurity investment using portfolio of security controls with maximum cybersecurity value," *International journal of production research*, no. ahead-of-print, pp. 1–17, 2021.
[10] Grigoryev, *AnyLogic in Three Days, Fifth Edition.* AnyLogic, 2021. https://www.anylogic.com/upload/al-in-3-days/anylogic-in-3-days.pdf.
[11] "Anylogic: Simulation modeling software tools & solutions for business," 2022. https://www.anylogic.com/.
[12] M. Souppaya and K. Scarfone, "Guide to malware incident prevention and handling for desktops and laptops." https://dx.doi.org/10.6028/NIST.SP.800-83r1, July 2013.
[13] T. Steffens, *Attribution of Advanced Persistent Threats*. Springer, 2020.
[14] Verizon, "Data breach investigation report," 2022. https://www.verizon.com/business/resources/reports/dbir/.
[15] D. Hubbard and R. Seiersen, *How To Measure Anything in Cybersecurity Risk*. Hoboken, NJ: Wiley, 2016.