Special Issue

JDMS

Journal of Defense Modeling and Simulation: Applications, Methodology, Technology
1–19

DOI: 10.1177/1548512920934551
journals.sagepub.com/home/dms
SAGE

Estimation of cyber network risk using rare event simulation

Alexander L Krall[1], Michael E Kuhl[2], and Shanchieh J Yang[2]

Abstract

Inherent vulnerabilities in a cyber network's constituent machine services can be exploited by malicious agents. As a result, the machines on any network are at risk. Security specialists seek to mitigate the risk of intrusion events through network reconfiguration and defense. When dealing with rare cyber events, high-quality risk estimates using standard simulation approaches may be unattainable, or have significant attached uncertainty, even with a large computational simulation budget. To address this issue, an efficient rare event simulation modeling and analysis technique, namely, importance sampling for cyber networks, is developed. The importance sampling method parametrically amplifies certain aspects of the network in order to cause a rare event to happen more frequently. Output collected under these amplified conditions is then scaled back into the context of the original network to provide meaningful statistical inferences. The importance sampling methodology is tailored to cyber network attacks and takes the attacker's successes and failures as well as the attacker's targeting choices into account. The methodology is shown to produce estimates of higher quality than standard simulation with greater computational efficiency.

Keywords

Cyber security, network risk, rare event simulation, importance sampling

1 Introduction

Security in cyber space has become an increasingly critical concern as the world tends toward omnipresent digital integration. All cyber networks have inherent vulnerabilities, which can be targeted and exploited by any malicious agent. The contemporary cyber-adversary may have a diverse range of intents. Some may seek to steal the private data of key institutions, while others may simply want to demonstrate their abilities. These activities and the selection of targets may or may not have ideological motivations. Regardless, the actions of all attackers result in significant financial loss: damaged systems must be repaired, intellectual property may be lost, etc. Security specialists seek to mitigate these financial impacts and work to protect their networks. Thus, one could say that specialists and attackers are at odds in a so-called ''battle of wits'' where the resulting conflict sees the application of complex tactics as each faction attempts to gain an advantage.[1] During such conflicts, an attacker may succeed in fulfilling their intent, which will impact the targeted institution. The National Institute of Standards and Technology (NIST) represents the synthesis of an adverse impact and its associated likelihood of occurrence as network risk. In other words, risk measures the extent by which an entity is threatened by a potential circumstance or event.[2]

The implicit goal behind mitigating financial impacts due to cyber attacks is minimizing network risk. Therefore, the network must be changed in some way: this can be done through simple reconfiguration or through the implementation of defensive measures. Either approach will change or obfuscate the set of attack actions that are feasible. To this end, deciphering a transformed network's risk

[1]Pennsylvania State University, University Park, PA, USA
[2]Rochester Institute of Technology, Rochester, NY, USA

Corresponding author:
Alexander L Krall, Penn State University, Leonhard Building, 310 South Barnard Street, University Park, PA 16802, USA.
Email: auk999@psu.edu

relative to its original state provides the needed context to a security specialist when assessing a variety of potential security countermeasures.

The complex behavior of attacks on cyber networks and the lack of closed-form methods for analyzing network risk has made computer simulation a leading analysis method. However, given that the objective of network security analysts is to make successful malicious events rare in the system, occurrences of these events in simulated cyber attacks are also rare. As a result, an extraordinary number of simulated attacks may need to be generated to produce enough observations of the rare event of interest to have sufficient data to produce a high-quality estimate of network risk. In this research, we design and develop an efficient rare event simulation (RES) modeling and analysis technique, namely, importance sampling (IS) for cyber networks. The IS method parametrically amplifies certain aspects of the network in order to cause a rare event to happen more frequently. Output collected under these amplified conditions are then scaled back into the context of the original network to provide meaningful statistical estimates and inferences of network risk. The IS methodology is tailored to cyber network attacks and takes the attacker's successes and failures as well as the attacker's targeting choices into account. The methodology is shown to produce estimates of higher quality than standard simulation with greater computational efficiency.

The remainder of this paper is organized as follows. In Section 2 we review the relevant related work in the area of network risk assessment and RES. The IS methodology designed for cyber networks is discussed in Section 3. In Section 4 the experimental evaluation procedure is presented; and the results of these experiments are presented in Section 5. Finally, conclusions and future work are presented in Section 6.

2 Related work

In this section, we discuss relevant related work relative to the assessment of network risk and RES methods.

2.1 Network risk

The risk of an event is defined by its likelihood and impact, which are the focus of this research. However, additional network metrics could be considered alongside risk. McQueen et al.[3] consider the time to compromise a target as a potential metric to track due to its representation of the effort expended by a malicious agent. Their study on Supervisory Control and Data Acquisition (SCADA) systems show that the time to compromise is related to the same machine's risk of being compromised. When the machine's risk decrease, its time to compromise increases.[3]

The likelihood of an event is dependent on three factors pertinent to an attacker: the attacker's intent, capability, and targeting.[2] The intent behind an attack is what an attacker seeks to accomplish with its malicious activity. Capability is synonymous with an attacker's available skillset, while targeting explicitly pertains to an attacker's movement through a network. In addition to these attacker-centric features, there is a temporal nature attached to likelihood. However, if an event is certain to occur at a given rate, the frequency of occurrence can be utilized to replace likelihood in the calculation of risk.[2] Each action performed by the attacker has a given timeframe of execution. According to Dell Secure Works,[4] each of these actions fall within one of 12 sequential categories, known as a kill chain. These categories can be seen in Figure 1. A preliminary study conducted by Rege et al.[5] determined the time consumed by an attacker at various steps in a kill chain. The findings showed that attack reconnaissance and exploitation each took 42% of the time of the attacks.

Within the scope of cyber security, the occurrence of an intrusion event is dependent on the successful exploitation of vulnerabilities present on machines in the network. A vulnerability is defined as a defect in a component or an erroneous or malicious behavior performed by a user.[6] The Common Vulnerability Scoring System (CVSS) rates all known vulnerabilities in terms of their severity on a scale from 1 to 10; higher scores indicate an increased probability of exploitation.[7] CVSS version 3 scores are assigned using four metrics: the access vector (AV), access complexity (AC), privileges required (PReq), and user interaction (UI).[8] AV refers to the context in which the vulnerability's exploitation is feasible. AC acknowledges the necessity of conditions beyond an attacker's control. PReq reflects the level of privileges an attacker must acquire prior to exploiting a vulnerability successfully. UI identifies any necessary interaction from a user that is not the attacker.[8] It is also possible for the attacker

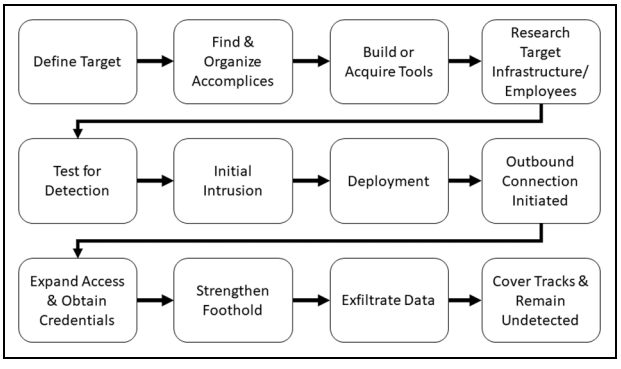

Figure 1. The different stages of the Dell kill chain.

to change a component whose authority is different to that of the vulnerable component. Such an instance is known as a change in scope and can affect the calculation of the PReq metric.[8] FIRST[9] details a methodology by which each of these metrics can be gauged. Furthermore, the use of CVSS scores can inform the development of a probabilistic representation of a network topology,[8] based on the services present on all machines.

In addition to assessing likelihood, a cyber attack can result in a diverse set of impacts. The exfiltration of sensitive information is known as a confidentiality impact. An integrity impact is the result of an asset being placed into a non-recoverable state. Lastly, the placement of an asset into a temporarily inaccessible state is known as an availability impact.[2,10] In addition to its type, an impact can be gauged with respect to certain predefined categorical measures. MITRE[11] assess impact with respect to cost, technical performance, and scheduling as part of their risk management assessment scale. Each category is ranked from minimal to severe on a 1–5 scale. The aggregate of each category's score is utilized as the event's numeric impact score.[11] Given the possibility for multiple events of interest, NIST states that impact can be represented as a vector, which can be combined with a corresponding likelihood vector to produce a risk vector.[2]

One methodology to reduce network risk is through the removal of attack paths.[12] Optimization can be employed to determine the pairwise connectivity between machines, which can then be minimized.[13] In addition, the use of alerts from network sensors in conjunction with knowledge of a network's attack graph can be utilized to correlate isolated alerts to attack scenarios.[14] Attack graphs can also be evaluated with respect to various categorical metrics, identified by Noel and Jajodia.[15] These include victimization, size, containment, and topology. The victimization metric pertains to a network's inherent vulnerabilities. The size family reflects the overall size of the attack graph. Containment refers to the compartmentalization of the network. Topology refers to the interactions between machines.[15] The implementation of advanced defensive measures, such as moving target defense (MTD), may also mitigate a network's risk. The MTD causes a network to periodically reconfigure itself to increase reconnaissance periods and make it necessary for an attacker to regain privileges.[16]

2.2 Rare event simulation

Simulation provides a means to derive performance statistics for stochastic systems. However, assessing risk may be computationally intensive if an event is sufficiently rare. Thus, RES techniques are often employed when dealing with rare events.

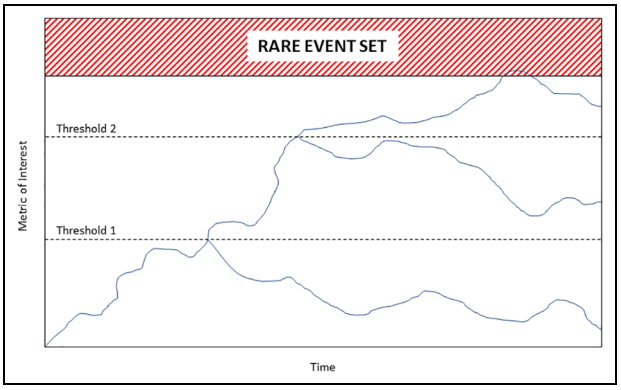

Figure 2. Example of the splitting technique in rare event simulation. The simulation's state is saved at given thresholds of "closeness" to the rare event. Should the state of a particular run drift "further" from the rare event, it can be restarted from the saved state that is "closest" to the event of interest.

There are two RES techniques that have seen widespread historic use. These are splitting and IS. However, each of these techniques has seen limited application to cyber security, as a greater focus has traditionally been placed on attack detection and prevention.[17] Splitting operates by creating copies of the simulation at various states and those able to obtain a sufficient level of ''closeness'' to the rare event are saved. The simulation utilizes these copies as starting points to improve the efficiency of experimentation.[18] Application of splitting to worm attacks shows that the technique yields superior estimations of a rare event's likelihood when compared to standard simulation.[17,19] The overall concept of the splitting technique can be seen in Figure 2.

IS operates on the notion of amplifying features of a network to cause increased incidence of a rare event. Amplification entails increasing the probability of one or more of a system's stochastic features.[20] For example, suppose probability p is equal to 0.15. If one were to amplify the value of p, its amplified counterpart, p', may take some value greater than 0.15, but less than 1. Output that is collected under these enhanced conditions is translated back into the context of the network's original conditions in order to produce usable statistics.[20] The process of feature amplification is performed by altering the probability of obtaining a certain value for a predetermined random variable and is known as a change of measure. Performing a change of measure runs the risk of increasing the likelihood of one event, but not another, or causing the absence of an event of interest to become rare.[20] The cross-entropy (CE) method seeks to address this concern as it can produce optimal changes of measure automatically through an iterative process,[21] but is not suitable for all probability distributions. A preliminary application of IS to model

cyber intrusion attacks, performed by Krall et al.,[22] demonstrates that quality estimates of event likelihood can be obtained with less computational effort than standard simulation.

2.3 Discussion

In this work, we consider a novel application of IS to the assessment of risk in computer networks. The use of simulation allows for explicitly modeling network details and allows for the assessment of risk by evaluating dynamic network attacks. As networks are designed for successful attacks to be rare, the number of simulated attacks needed for a high-quality assessment is large. The IS algorithms that we have developed enable risk assessment with less simulation effort than traditional simulation approaches.

3 Network risk estimation methodology

We have designed the following methodology to evaluate the risk of successful execution of an attack of interest within a cyber network. Within this context, the attack surface is defined by the services present on the various physical machines that comprise the network. We first present an overview of the IS simulation method for assessing network risk. We then present a brief overview of IS followed by a detailed discussion of the application of IS in the context a cyber network. We then present our simulation modeling approach relative to attacker behavior and attack progression through the network, including the selection of target machines and vulnerabilities and likelihood of success. Next, we discuss how the impact assessment is combined with the likelihood to produce the estimate of network risk. Finally, we present our methodology for assessing the performance of the IS method.

3.1 Importance sampling for cyber networks

Consider cyber attacks where an attacker moves through a network. In these cases, the events of interest from the defensive side entail an attacker being able to reach certain machines. Given that networks may be highly interconnected, it may be difficult to distinguish an attacker's ''closeness'' to an event. The obfuscation of event ''closeness'' makes splitting unfavorable for such attack scenarios. Thus, IS is used in lieu of splitting. The path taken by an attacker is determined by the choices of targets as well as the successes and failures during a particular attack scenario. CVSS scores inform the probabilistic nature of the successes and failures when attempting to compromise a machine by means of exploiting its available services. Note that, without loss of generality, all attackers are assumed to start in a single position that is external to the network.

The proposed IS methodology will generate a risk vector for a cyber network. Each element within the risk vector corresponds to the risk of an attacker reaching a machine:

$$\mathbb{R} = \mathbb{L} \times \mathbb{I}. \tag{1}$$

In the above representation, $\mathbb{R}$ is the risk vector while $\mathbb{L}$ is the likelihood vector. The likelihood vector is held with respect to the same time threshold, T. The impact vector is given by $\mathbb{I}$. For the purposes of this methodology, each entry of $\mathbb{I}$ is predetermined and is treated like a parameter. If desired, impact could follow its own distribution. The generalized IS methodology will calculate $\mathbb{L}$ by conducting the following:

1. obtain probabilistic network parameters;
2. assess candidates for amplification;
3. perform the amplification;
4. simulate using the amplified network; and
5. scale the output into the context of the original network.

3.2 General importance sampling

The generalized IS methodology, shown in Figure 3, first considers a metric of interest Y with possible outcomes $Y(x)$. Each outcome, x, occurs with probability $f(x)$. Using standard simulation, one can determine the expected value of Y at density f, which is represented as $\mathbb{E}_f(Y)$.[20] The following equation shows this calculation:

$$\mathbb{E}_f(Y) = \int Y(x) f(x) dx. \tag{2}$$

Network parameters are then assessed as candidates for amplification. Performing a change of measure on these parameters will also modify density f into g. Therefore, under amplified conditions, each outcome $Y(x)$ would occur at probability $g(x)$. Thus, the expected value of Y at density g is given by $\mathbb{E}_g(Y)$.[20] Note that multiplying density f in Equation (2) by $\frac{g}{g}$ is the same as multiplying by one:

$$\mathbb{E}_f(Y) = \int Y(x) \frac{f(x)}{g(x)} g(x) dx. \tag{3}$$

Removing the term $\frac{f}{g}$ from Equation (3) gives a framework by which one would simulate under amplified conditions:

$$\mathbb{E}_g(Y) = \int Y(x) g(x) dx. \tag{4}$$

The ratio of density f to density g, given by W, can then be utilized to translate the output of the amplified

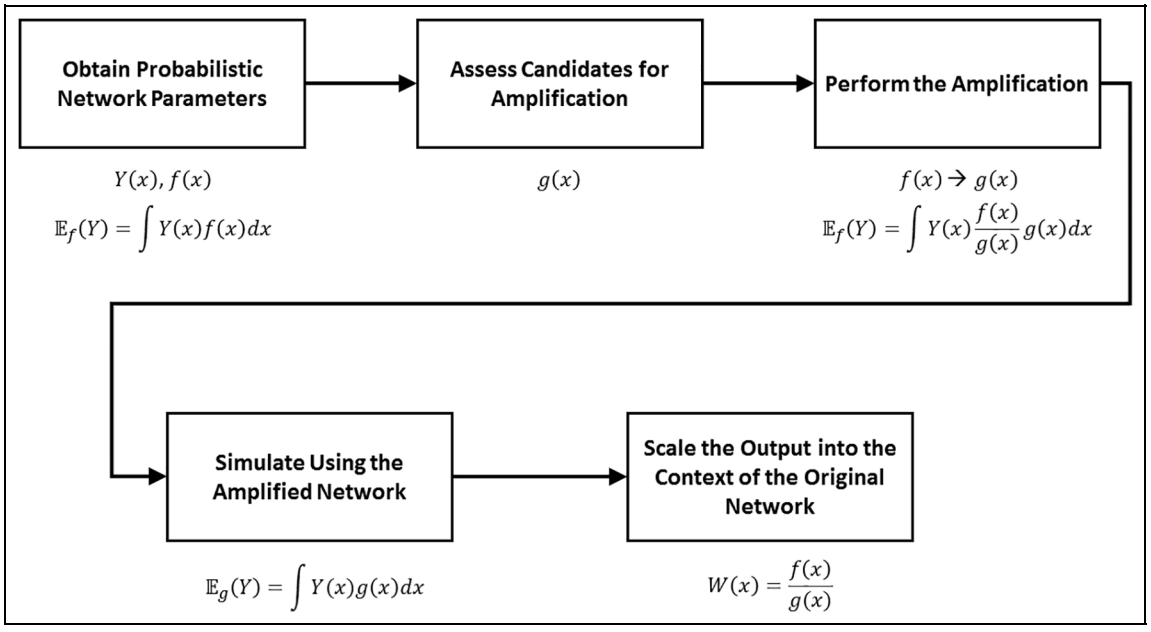

Figure 3. Importance sampling methodology flowchart for a single simulation replication. If utilizing multiple replications, the last two steps are repeated for each new replication.

simulation back into the context of the original network such that:

$$W(x) = \frac{f(x)}{g(x)}. \tag{5}$$

This translation works by multiplying each outcome $Y(x)$ by each corresponding $W(x)$. Essentially, W is utilized to remove the impact of g on Y such that the final output is only held with respect to f, which is the original context.[20] Scaling the amplified output by W produces the same expected value as in Equation (2):

$$\mathbb{E}_g(YW) = \int Y(x)\frac{f(x)}{g(x)}g(x)dx = \mathbb{E}_f(Y). \tag{6}$$

3.3 Cyber security IS framework

The generalized IS methodology is tailored into a security framework that represents an attacker moving through a network. The calculations for the likelihood of an attacker reaching a particular target machine utilizes the following information.

SETS

M set of all machines.

Z set of all target machines, $Z \subseteq M$.

M_m set of machines accessible from machine m, $M_m \subseteq M$.

$M_m^v(j)$ set of vulnerable machines accessible from machine m during attempt j, $M_m^v(j) \subseteq M_m$.

A_{mn} set of services on machine n visible from machine m.

VARIABLES

$u_{mn}(j)$ probability of targeting machine n from machine m on attempt j.

v_{kmn} probability of targeting service k on machine n from machine m.

$q(j)$ machine/service selection probability during attempt j.

$p(j)$ success/failure probability during attempt j.

$$x = \begin{cases} 1 & \text{If service } k \text{ is compromised during attempt } j, \\ 0 & \text{Otherwise.} \end{cases}$$

τ_z compromising time of machine z.

$f_z(\tau_z)$ probability that an attacker has reached machine z at compromising time τ_z.

$g_z(\tau_z)$ probability under amplified conditions.

A visualization of the various defined machine sets is shown in Figure 4.

The adversary leverages attack attempts against the services present on machines in the network. Within the total time horizon, T, the attacker can execute a maximum of J attempts. It is assumed that the attacker has both the skill and desire to continue the attack until time T. Each attempt consumes the same amount of time. During a given replication, there are Ψ trials. Should a target, z, be

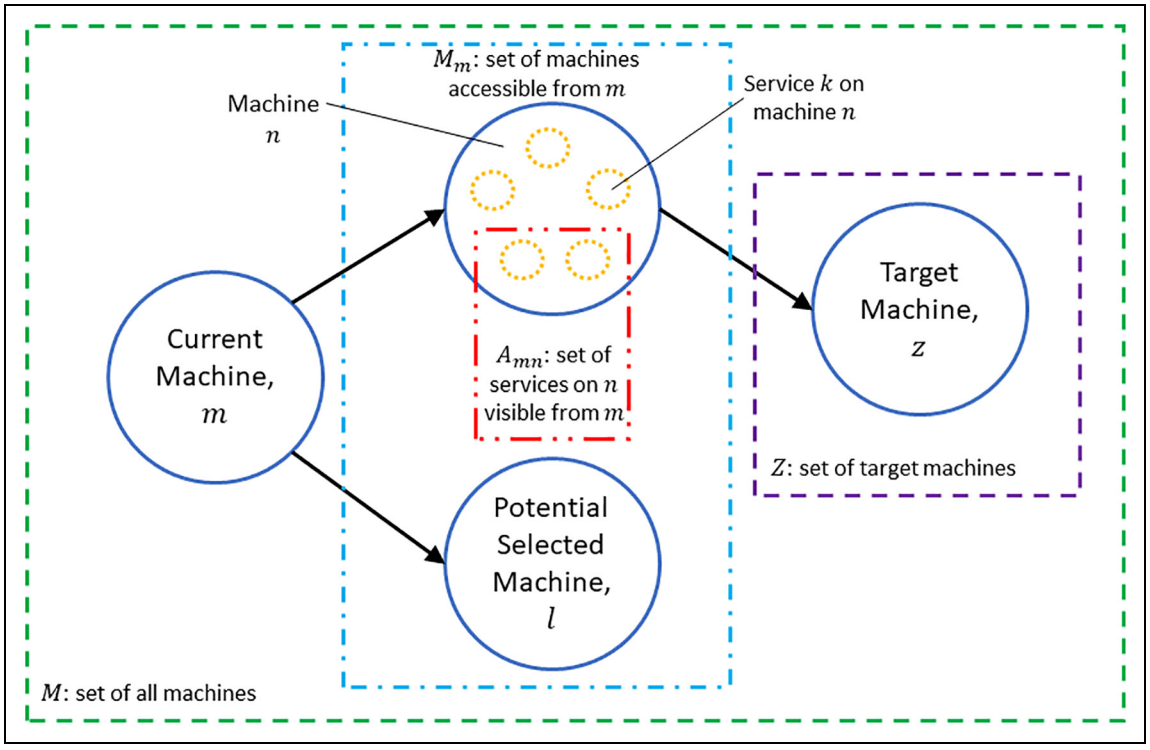

Figure 4. A visualization of the various defined machine sets.

compromised during a particular trial $\psi \in \Psi$, the associated indicator variable will take a value of 1. Otherwise, it will take a value of and 0, such that:

$$I_{z\psi} = \begin{cases} 1 & \text{If target } z \text{ is compromised during trial} \\ & \psi \text{ within time horizon } T, \\ 0 & \text{Otherwise.} \end{cases} \quad (7)$$

When utilizing standard Monte Carlo simulation, the likelihood, L_z, that target z is compromised can be found by calculating the expected value of the indicator variable.[21] It is represented as follows:

$$L_z = \mathbb{E}_f(I_z) = \frac{1}{\Psi}\sum_{\psi=1}^{\Psi} I_{z\psi}. \quad (8)$$

The likelihood of reaching a target machine is held with respect to each trial ψ. Thus, likelihood is dependent on the targeting, successes, and failures an attacker experiences while moving through the network. Targeting refers to how the attacker moves through the network and is always done in two phases. The first phase entails the selection of an available target machine. Available machines are either connected to machines the attacker controls or publicly facing. Given a network movement strategy, the attacker will consider machines that are both accessible and vulnerable. For example, an attacker seeking to maximize the depth of their penetration into the network will only consider machines accessible from the most recent machine that has been compromised. An attacker prioritizing breadth of expansion will not be held to such a restriction and can consider all accessible uncompromised machines.

Each available machine under consideration of the attacker may have a unique selection probability. However, for the purposes of this methodology, each considered machine is given an equal likelihood of selection:

$$u_{mn}(j) = \frac{1}{|M_m^v(j)|} \qquad \forall\, n \in M_m^v(j). \quad (9)$$

Once a machine has been targeted, the attacker will select a vulnerable service. Each service is given a selection weight. These weights correspond to both an attacker's interests and current capabilities. The probability of selecting a service is calculated by dividing the service's individual weight by the aggregated weights of all considered services on the machine:

$$v_{kmn} = \frac{w_k}{\Sigma_{i \in A_{mn}} w_i} \qquad \forall\, k \in A_{mn}, m \in M, n \in M_m. \quad (10)$$

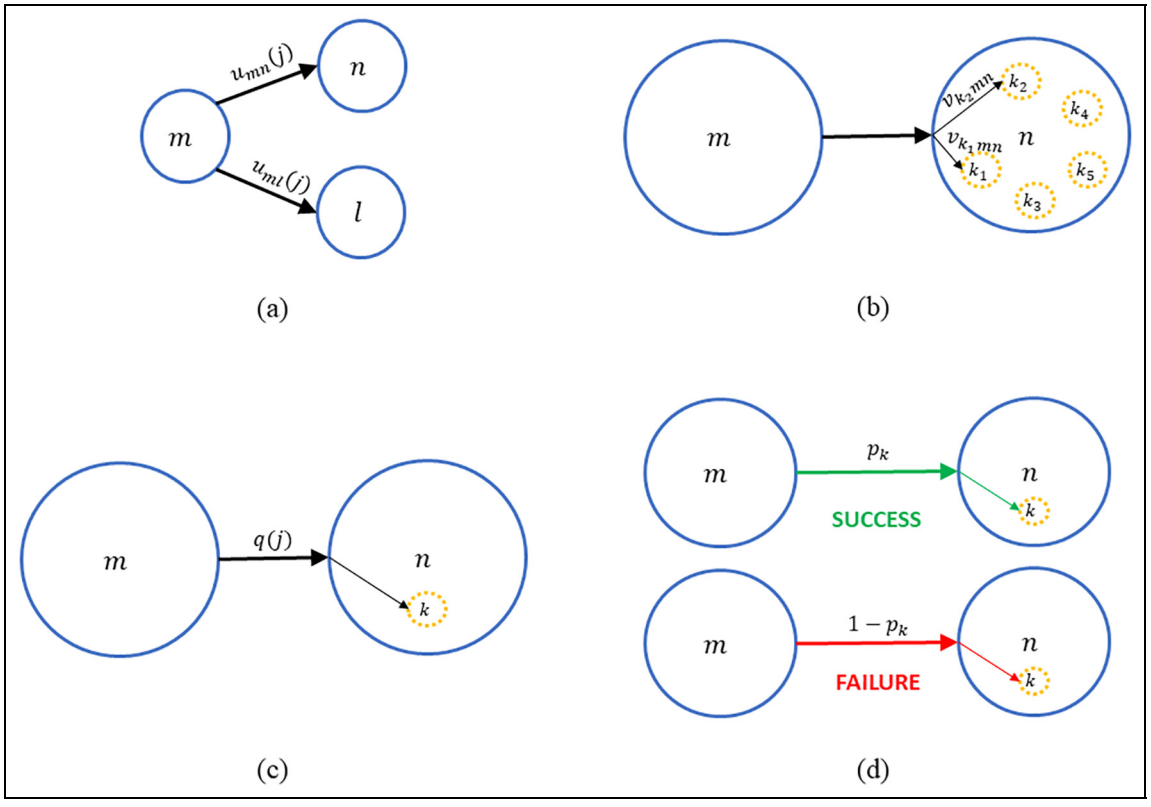

Figure 5. Probabilities for (a) machine selection, (b) service selection, (c) combined machine and service selection, and (d) combined machine and service attack success.

Thus, targeting as a whole can be represented as the product of both the machine and service selection steps:

$$q(j) = v_{kmn}u_{mn}(j) \qquad \forall j \in \{1, 2, ..., J\}. \tag{11}$$

After targeting has been completed, an attack attempt will be leveraged against the selected machine. The attacker will either have a success or failure at each attempt. The probability that an attacker succeeds or fails during an attempt is dependent on the probability of successfully compromising the selected service:

$$p(j) = p_k^x(1 - p_k)^{1-x} \qquad \forall j \in \{1, 2, ..., J\}. \tag{12}$$

Should an attack on a machine's service fail, the attacker is free to perform the same attack again. The probability that an attack succeeds is assumed to be independently and identically distributed. Figure 5 illustrates how probabilities are determined for machine selection; service selection; combined machine and service selection; and combined machine and service attack success.

If given an indefinite timeframe, an attacker would be able to reach every target. Given that the simulation has an established time horizon, T, there will be cases where τ_z is unknown for a particular machine. The number of attempts required to compromise a machine of interest is utilized in calculating the corresponding likelihood value. To this end, when τ_z is unavailable, T is used in its place, which corresponds with J attempts. This concession is done for the sake of practicality when conducting the simulation's calculations. Any incomplete path to a machine will result in $I_{z\psi} = 0$, which will drive the final likelihood calculation to zero. Thus, the number of attempts required to compromise a machine of interest is shown by the following:

$$J_z(\tau_z) = \min\,(\text{Attempts to reach } z \text{ at time } \tau_z,\ J). \tag{13}$$

The attacker's path through the network is represented by the machine/service selections and success/failure of each attempt. The product of the two components $q(j)$ and $p(j)$ gives the probability of adding the particular choice and outcome to the path. Let this product for each attempt be known as the movement factor. The product of all movement factors determines the probability of generating a successful path to a machine of interest:

$$f_z(\tau_z) = \prod_{j=1}^{J_z(\tau_z)} p(j)q(j) \qquad \forall z \in Z. \tag{14}$$

Amplifying any of the associated success/failure or choice parameters will also affect the probability of generating an attack path. When any component of $p(j)$ or $q(j)$ are amplified, then each become $p'(j)$ and $q'(j)$, respectively. The probability of generating an amplified path is given by the following:

$$g_z(\tau_z) = \prod_{j=1}^{J_z(\tau_z)} p'(j)q'(j) \qquad \forall z \in Z. \tag{15}$$

The ratio of probability density f to density g for each target machine is shown by the following:

$$W_z = \frac{f_z(\tau_z)}{g_z(\tau_z)} \qquad \forall z \in Z. \tag{16}$$

The ratio W_z is employed when utilizing IS due to the calculation of L_z being held with respect to density g rather than density f. Scaling the simulation under amplified conditions must make use of each W_z. Thus, a modified version of Equations (6) and (8) produces the likelihood calculation for IS:

$$L_z = \mathbb{E}_g(I_z W_z) = \frac{1}{\Psi} \sum_{\psi=1}^{\Psi} I_{z\psi} W_{z\psi}. \tag{17}$$

Tailoring the probability distribution into this security framework enables the application of IS to this context as it provides a clear framework for conducting parametric amplification. Furthermore, classification of the probability density function enables conversion to the original state from an amplified state. Thus, the likelihood a machine of interest is compromised can be determined. Figure 6 depicts how the IS methodology was formally implemented. The inputs for the methodology are as follows:

1. number of trials and replications;
2. network topology and services;
3. impact and target data;
4. default service selection weights;
5. default service success probabilities;
6. amplifications to weights and success probabilities; and
7. attacker movement strategy.

3.4 Attacker movement

Various machine movement strategies can be implemented as part of the attacker's behavior. The depth-based search (DBS) is one such strategy where targeting is taken with respect to the most recently compromised machine. As depicted in Algorithm 1, several parameters are initialized prior to attacker movement. These include the number of attempts (j), attacker starting position, attacker knowledge, set of target machines, and success/failure indicator. The attacker knowledge is represented as the set of all machines that have been compromised by the attacker at a given number of attempts. Each machine within the attacker's knowledge will remain under the attacker's control until reaching the maximum number of attempts, J. The attacker will scan all outgoing connections from its current position. Should all outgoing connections lead to machines within the attacker's knowledge, the attacker will backtrack through its path so far. Backtracking will continue until there is at least one outgoing machine that has not been compromised. If the attacker backtracks to its starting location, then the infiltration event will terminate.

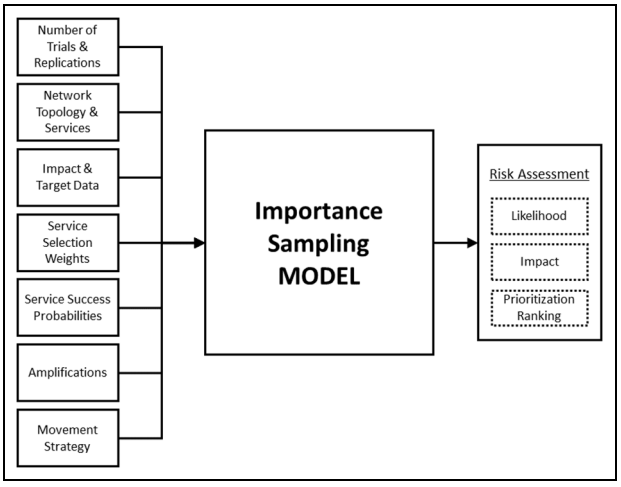

Figure 6. Model implementation structure.

On the contrary, an available machine will be selected if available. Once the machine selection is complete, the attacker will target a service on the machine and initiate an attack. Regardless of the outcome, the success/failure indicator will be updated appropriately. Once the attack has ceased, the total number of attempts will be incremented by one. Should the attack be successful, the attacker's current position will be updated and the attacker knowledge will be updated accordingly. After updating the knowledge, the algorithm will check if the compromised machine is within the set of target machines. Should this be the case, the likelihood of reaching the target will be calculated. if all target machines are compromised, the infiltration event will stop and the replication will come to an end.

The breadth-based search (BBS) is another type of attacker movement strategy and is similar to the DBS. The main difference between the two strategies is the choice of targeting. The BBS does not utilize a single source node as a pivot point. In lieu of this, all machines within the

Algorithm 1: Depth-based movement

```
1: j ← 1
2: SourceNode ← Internet
3: Knowledge ← addKnowledge(Internet)
4: TargetMachines ← addTargets()
5: while j < J do
6:    MachineOptions ← availableConnections(SourceNode)
7:    If isDeadEnd(MachineOptions) then
8:       If isNewSourceAvailable(Knowledge) then
9:          SourceNode ← chooseNewSource(Knowledge)
10:         MachineOptions ← scanConnections(SourceNode)
11:      else
12:         break
13:      end if
14:   else
15:      SelectedMachine ← machineSelection(MachineOptions)
16:      SelectedService ← serviceSelection(SelectedMachine)
17:      AttackStatus ← attackMachine(SelectedService)
18:      j ← j + 1
19:      if isAttackSuccessful(AttackStatus) then
20:         SourceNode ← updateSourceNode(SelectedMachine)
21:         Knowledge ← Knowledge + addKnowledge
            (SelectedMachine)
22:         if isTarget(SelectedMachine,TargetMachines) then
23:            recordLikelihood(SelectedMachine)
24:            if isAllTargetsCompromised(Knowledge,
               TargetMachines) then
25:               break
26:            end if
27:         end if
28:      end if
29:   end if
30: end while
```

attacker's knowledge are treated like a collective source. Thus, every un-compromised machine that is connected to a machine in the attacker's knowledge will be considered for selection. The distinction between attacker targeting strategies is displayed in Figures 7 and 8.

3.5 Impact, selection, and CVSS heuristics

The impact of events of interest is also fed into the IS model as an input. One possible way that these impact ratings can be assigned to machines of interest is through the heuristic defined in Table 1, which is based on similar work done by MITRE.[11] Each category in the heuristic (operational, financial, and schedule) is rated on a scale from 1 to 5, with 5 being the most severe rating. The operational categories refer to the impact of an event on the ability of the organization to perform its core functions. The financial category assesses the direct implications to budgeting, while the schedule category reflects any required adjustments to project timelines.

The probability of successfully compromising a susceptible service can be determined by assessing the service's present vulnerabilities utilizing CVSS version 3.[8] For each CVSS category (AV, AC, PReq, and UI), the vulnerability is assigned a severity value based on Table 2. For the

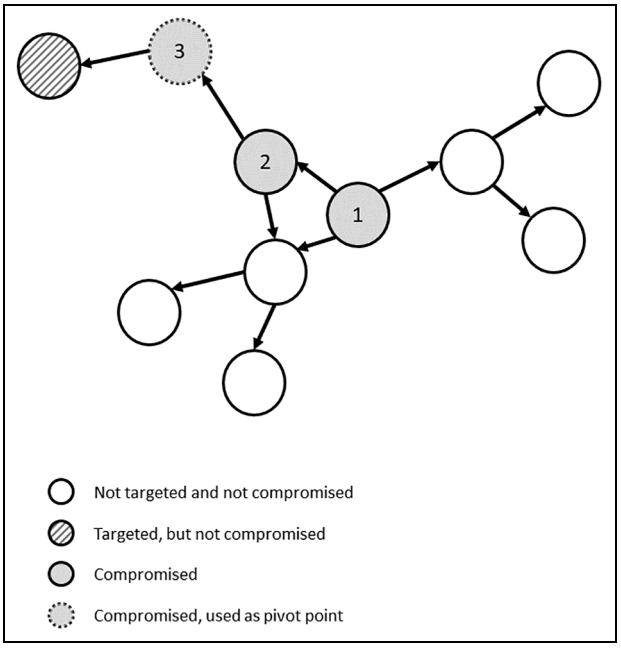

Figure 7. Depth-based machine selection example. The numeric labels indicate the sequence in which machines were compromised. Only machines connected to the most recently compromised machine are targeted.

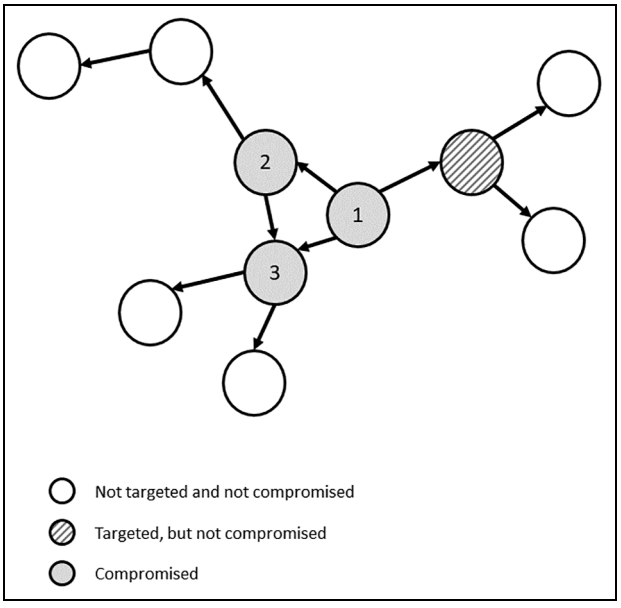

Figure 8. Breadth-based machine selection example. The numeric labels indicate the sequence in which machines were compromised. However, this order does not matter for targeting since all accessible machines can be targeted.

Table 1. Impact heuristic.

Impact	Operational	Financial	Schedule
5 Severe	Ability to perform core business function completely crippled.	Exceptional budget impact.	Exceptional scheduling adjustments required.
4 Significant	Ability to perform core business function is significantly impaired.	Budget significantly exceeds planned amounts.	Major scheduling adjustments required.
3 Moderate	Ability to perform core business function is moderately impaired.	Budget moderately exceeds planned amounts.	Moderate scheduling adjustments required.
2 Minor	Ability to perform core business function is slightly impaired.	Budget slightly exceeds planned amounts.	Minor scheduling adjustments required.
1 Minimal	No impact on ability to perform core business function.	Budget is not affected. No planning adjustments required.	Schedule is not affected. No planning adjustments required.

purposes of this investigation, the vulnerabilities present in a particular service's version are treated as a singularity. FIRST classifies a methodology for assigning categories to vulnerabilities.[9] With regards to AV, if the attacker exploits a vulnerable component via the network stack, the service is either placed into the ''Network'' or ''Adjacent'' classification. Within this domain, if the vulnerability can be exploited from a routed network the classification will be ''Network''; otherwise, the classification will be ''Adjacent.'' If the vulnerable component is not exploited via the network stack, then AV will be classified as ''Local'' or ''Physical.'' If the attack requires physical access to the network, then the classification will be ''Physical.''

AC can take one of two values: it will be High if the attacker cannot exploit the vulnerability at will; otherwise, it will be ''Low.'' PReq can take one of three classifications: ''None,'' ''Low,'' or ''High.'' if the attacker does not need to be authorized, then PReq is ''None.'' If administrator privileges are required, PReq is High. UI is either assigned ''Required'' or ''None.'' The Required classification is assigned only if the attacker requires another user to perform an action to exploit a vulnerability.

Each of the scores seen in Table 2 are derived from Zhang et al.[8] and define one possible way to assigning success probabilities. These scores are given a high and low value to provide a level of stochasticity to the model. A uniform distribution is then utilized to give each service a single score for each of the main CVSS version 3 categories. The final probability is then calculated for each service, 1, ..., K, by the following:

$$p_k = 2.11 \times AV_k \times AC_k \times PReq_k \times UI_k \qquad \forall\, k \in \{1, 2, ..., K\}. \tag{18}$$

An attacker must choose which machine to target while moving through the network. In addition, once a machine is chosen, the adversary must then choose which service on the machine to compromise. The process by which this selection takes place utilizes a heuristic that quantifies the attacker's interest, which is dependent on the attacker's capability and intent. These numeric representations of interest are known as service weights. For the purposes of this investigation, interest is partitioned into four categories: low, medium, high, and very high. Each of these categories has a range of possible values. The minimum and maximum values for each range are shown in Table 3. Service weights are calculated using a uniform distribution with parameter values derived from the corresponding interest category.

Table 2. Common Vulnerability Scoring System categorization values.

Metric	Category	Low value	High value
Access complexity	High	0.4180	0.4620
	Low	0.7315	0.8085
Access vector	Physical	0.1900	0.2100
	Local	0.5225	0.5775
	Adjacent	0.5890	0.6510
	Network	0.8075	0.8925
Privileges required	None	0.8075	0.8925
	Low	0.5890	0.6510
	High	0.2565	0.2835
User interaction	None	0.8075	0.8925
	Required	0.5890	0.6510

3.6 Assessment strategy

The performance of the proposed IS methodology can be assessed via two perspectives. The first perspective contrasts the quality of risk estimates between methods. A $(1-\alpha)$% confidence interval can be produced for a static number of runs per replication, where α is the probability of Type I error. Risk estimates at different degrees of amplification should be roughly the same. Nonetheless, the bounds of their confidence intervals should be

Table 3. Interest rating.

Interest	Lower bound	Upper bound
Very high	6.4	9.9
High	1.6	3.2
Moderate	0.4	0.8
Low	0.1	0.2

different. Confidence intervals with smaller halfwidths are considered to be more accurate estimates.

The second perspective assesses the computational effort required to ascertain a quality estimate of risk. A quality estimate is defined to have a confidence interval that falls within some percentage of the mean. Trials will be run until all risk estimates converge to the quality criteria. Expedient convergence indicates lesser computational effort. To increase the efficiency of the simulation, convergence is not checked after each trial. Instead, a milestone system is employed. These milestones represent a specific number of trials during a replication that will trigger a convergence check. If convergence is not reached at a milestone, another milestone will be computed based on current statistics.

CONVERGENCE
Λ_z set of likelihoods of compromising target z.
$\bar{L}_z$ average likelihood of compromising z.
s_z likelihood standard deviation for machine z.
$t_{\frac{\alpha}{2},|\Lambda_z|-1}$ t-statistic.
θ likelihood quality threshold (%).
ψ_z^c number of trials needed before re-checking convergence for machine z.

The confidence interval for the likelihood must converge to be within some θ% of the mean. Therefore, one can determine the number of trials needed to obtain a quality estimate. For each machine of interest:

$$\bar{L}_z(1+\theta) \geqslant \bar{L}_z + t_{\frac{\alpha}{2},|\Lambda_z|-1}\frac{s_z}{\sqrt{\psi_z^c}}. \tag{19}$$

Solving for ψ_z^c, we get the following:

$$\psi_z^c \geqslant \left(\frac{t_{\frac{\alpha}{2},|\Lambda_z|-1}s_z}{\theta\bar{L}_z}\right)^2. \tag{20}$$

Likelihood convergence must be reached for all machines of interest. Therefore, the milestone will be the maximum of all ψ_z^c.

Trials run from both evaluation perspectives are given additional utility by allowing the attacker to compromise multiple machines of interest within a single path. Under this framework. statistics are collected about all events of interest simultaneously. Since each event is not considered in isolation, fewer experimental replications are required by the simulation.

4 Experimentation

4.1 Experimental network

To evaluate the capabilities and limitation of the IS methodology, we compare a base network (Figure 9) with four alternative network configurations (Figure 10). The base configuration is derived from the Collegiate Penetration Testing Competition held in 2016.[23] Arcs in the network are directed, but come with a few caveats. The presence of an arc means that the two machines are capable of communication with each other. The direction of an arc, however, represents a firewall rule that governs the capability to write data to a machine via one of its services. It is assumed under the conditions of the experiment that an attacker must be able to change another machine's state through writing data under the assumption that the appropriate machine-to-machine permissions are valid.

The network has four interconnected sub networks and represents a hypothetical healthcare-oriented facility. Only certain machines are publicly facing. Subnet 1 contains machines common to a doctor's office. Workstation WRK EMR handles electronic medical records (EMRs), whereas workstation WRK BILL is responsible for billing. Workstation WRK IT deals with information technology-related issues. DC01 is a domain controller that authenticates access to the subnet file share, which is represented as FILES. The network also has a network printer, PRINT, and an x-ray machine, XRAY-13.

Subnet 2 handles the EMR functionality of the entire facility. WEB02 is the EMR application server, where information is stored on the DB02 database. Subnet 3 hosts a diverse set of functionality. WEB01 is a billing application server that stores its information on the DB01 database. OP & WIKI hosts an IT Wiki, while PR operates a public relations Twitter bot that publishes protein folding research.

The final subnet, Subnet 4, handles the aforementioned protein folding research as well as the remote desktop capabilities between workstations. TS01 is a terminal services application server that enables workstation WRK IT remote access into the other workstations on the network. FOLDING represents an application server that conducts the protein folding research. Information from this application server is stored within STORAGE. CI is a continuous integration server and works in conjunction with the GIT repository server for code development. BASTION acts as a protective layer that strictly limits access into the subnet.

All services present on the network must be assigned a probability of becoming compromised if attacked. These

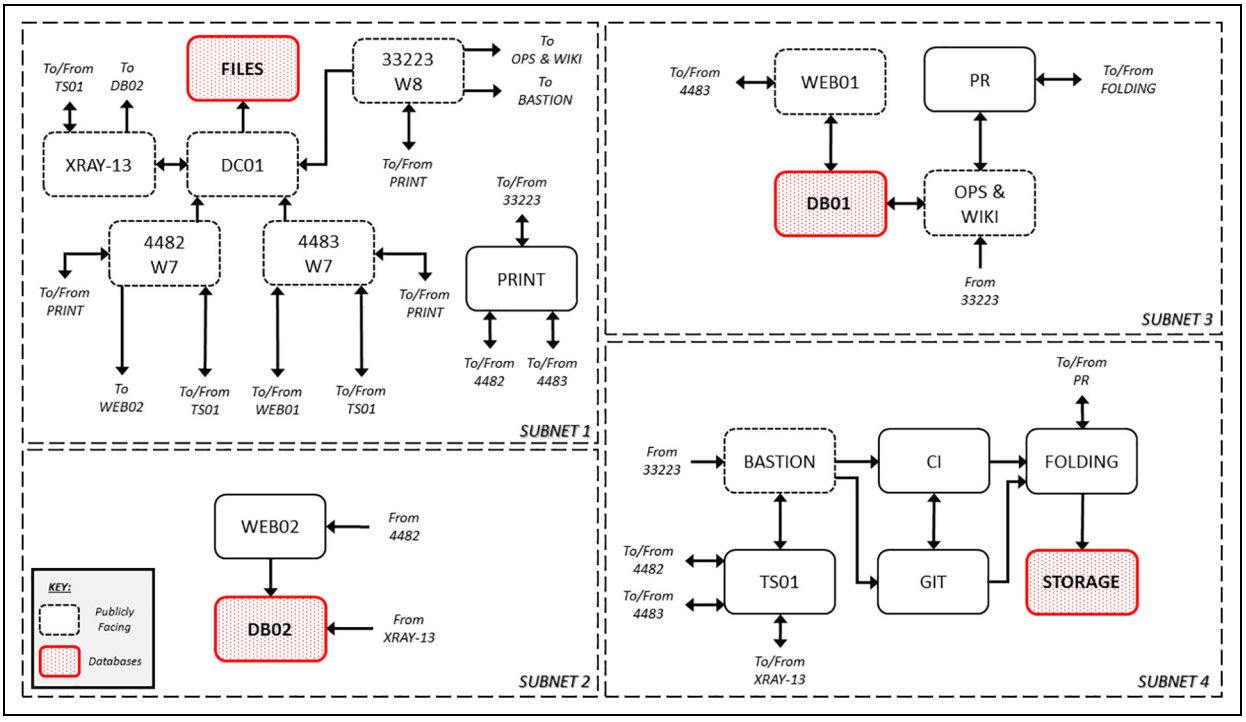

Figure 9. Network example – base case.

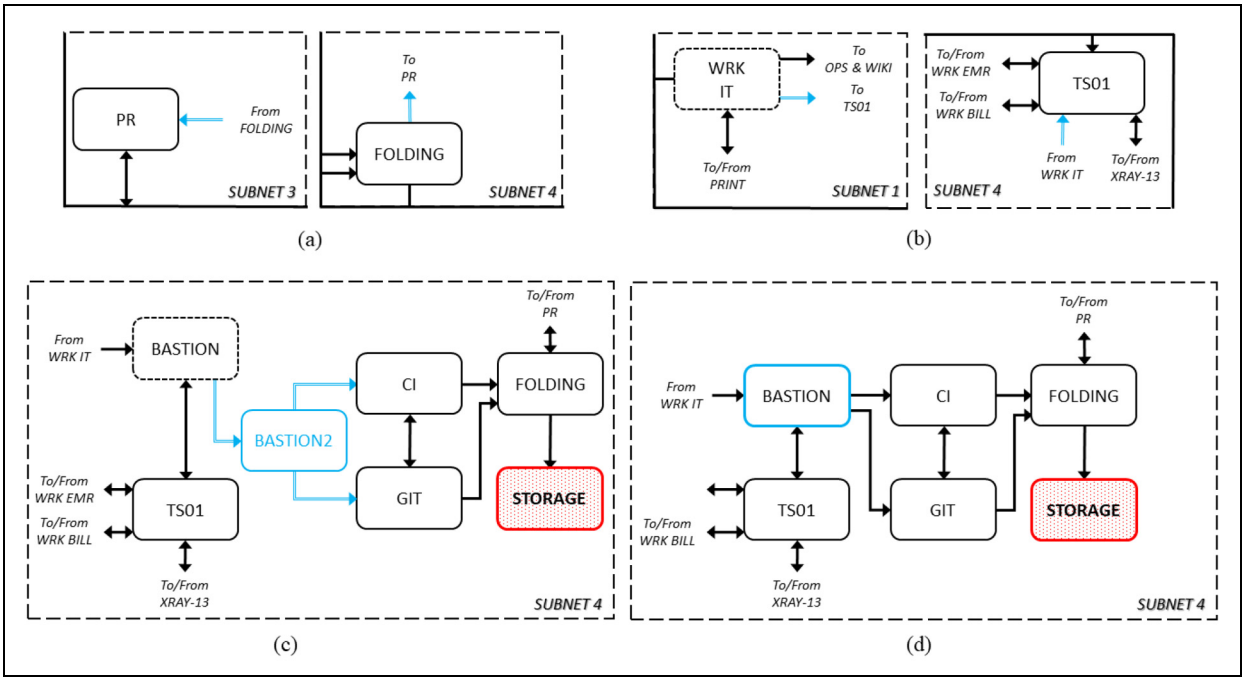

Figure 10. Alternative network configurations: (a) modify connection; (b) move connection; (c) add new machine; (d) change machine access from public to private.

Table 4. Service Common Vulnerability Scoring System categorization.

Services	Locations	Interest	Access complexity	Access vector
Domain controller	DC01	High	Low	Network
Domain file share	FILES, STORAGE	Very high	Low	Adjacent
EMR web application	WEB02	High	High	Adjacent
FreeBSD 9.1	BASTION	Moderate	High	Network
GitLab	GIT	Moderate	Low	Adjacent
Internal IT Wiki	OPS & WIKI	Low	High	Adjacent
Jenkins CI	CI	Low	High	Adjacent
MySQL	DB02, STORAGE	Very high	High	Adjacent
NodeJS web application	PR	Low	High	Adjacent
Non-HIPAA/PCI compliant billing application	WEB01	High	High	Network
Picture archive and communication system	XRAY-13	High	High	Network
PostgreSQL	DB01	Very high	High	Adjacent
Print application	PRINT	Low	Low	Adjacent
Protein folding application	FOLDING	High	High	Adjacent
Remote desktop	TS01	Moderate	High	Adjacent
SSH	BASTION	High	High	Network
Telnet	STORAGE	Low	Low	Adjacent
Terminal services	TS01	Moderate	High	Adjacent
Tomcat	CI	Moderate	High	Adjacent
Ubuntu 16.04-1	WEB02, DB02, DB01, PR, CI, GIT, FOLDING, STORAGE	Low	High	Adjacent
Ubuntu 16.04-2	WEB01, OPS & WIKI	Low	High	Network
Windows 7	WRK EMR, WRK BILL	High	Low	Network
Windows 8	WRK IT	High	Low	Network
Windows Server 2003	XRAY-13	Moderate	Low	Network
Windows Server 2008 R2-1	DC01	Moderate	High	Adjacent
Windows Server 2008 R2-2	FILES	Moderate	High	Network
Windows Server 2012	TS01	Moderate	High	Network

EMR: electronic medical record.

assignments must be compliant with CVSS version 3. Table 4 displays information regarding each network service, including their machine locations and CVSS categorization. UI is set to ''None'' and PReq is set to ''High'' for all machines. Table 4 also shows the interest categorizations utilized for assigning services selection weights.

Within the context of this network example, events of interest are limited to an attacker exfiltrating data stores on the four database servers: FILES, DB01, DB02, and STORAGE. These impact ratings can be seen in Table 5.

Table 5. Impact assignments.

Machine	Impact score		
	Operational	Financial	Schedule
FILES	2	1	3
DB01	4	3	2
DB02	4	4	3
STORAGE	4	5	5

Four configuration alternatives to the base case are explored. The first alternative can be seen in Figure 10(a) and modifies the connection between PR and FOLDING to be unidirectional. Under this ''Modify Connection'' case, data can only be sent from FOLDING to PR. The second alternative is shown in Figure 10(b) and is identified as ''Move Connection.'' The connection between WRK IT to BASTION is moved such that it now exists between WRK IT and TS01. The third case, ''Add Machine,'' sees the addition of a second BASTION server to the network, shown as BASTION2 in Figure 10(c). The data contained on STORAGE is important to the organization, and thus an additional layer of authentication may be desired. The final alternative changes the BASTION server such that it is no longer publicly facing. This ''Public to Private'' case is shown in Figure 10(d).

4.2 Experimental setup

Each experimental case is run with 30 replications to ensure a large number of possible scenarios is observed. When analyzing computational efficiency, reasonable computer memory constraints are considered. Thus, the maximum number of trials is 1×10^8 for both DBS and BBS. The convergence criteria is set to be a confidence

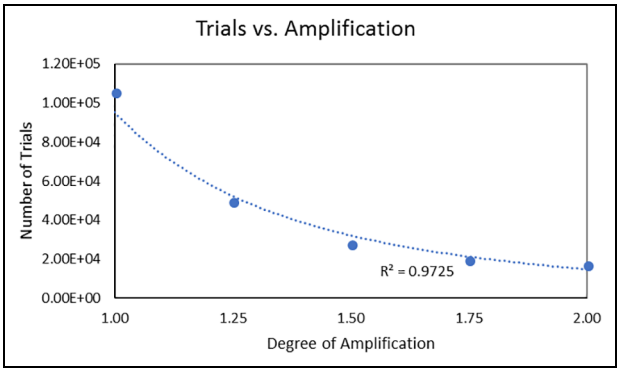

Figure 11. Convergence in the depth-based case – computational savings.

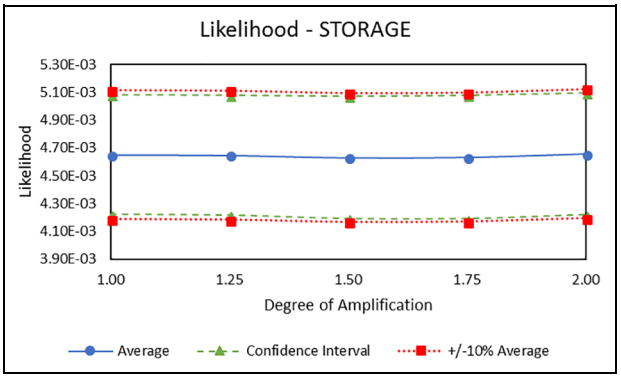

Figure 12. Convergence in the depth-based case – STORAGE likelihood.

interval falling within $\pm 10\%$ of the mean estimate. This criterion ensures the halfwidth stays well within an order of magnitude of the estimate. When testing for estimation quality, the DBS receives a static 50, 000 trials, while the BBS is given 4×10^6 trials. These two numbers are different because the BBS requires significantly more trials before reaching a quality estimate. In all cases, $\alpha = 0.05$ and the maximum number of attack attempts is $J = 10$, which allows for the occurrence of multiple events of interest during a single trial. Different levels of amplification are tested and are carried forth by multiplying all success probabilities by a given factor. Amplification levels of $1\times$, $1.25\times$, $1.5\times$, $1.75\times$, and $2\times$ will be tested. Amplification above $2\times$ runs the risk of assigning probability values greater than one. The $1\times$ case corresponds to no network amplification and is representative of standard simulation.

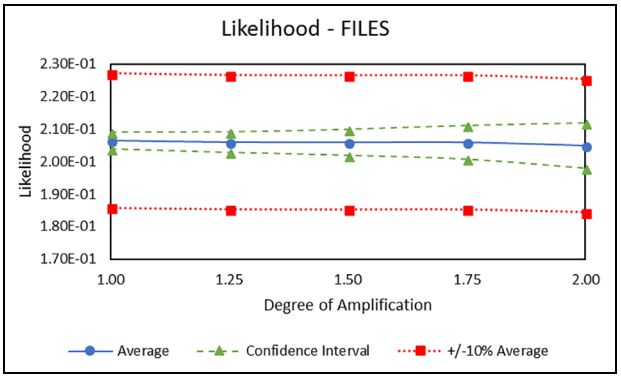

Figure 13. Convergence in the depth-based case – FILES likelihood.

5 Results

5.1 Computational savings versus estimation performance

The number of trials required to converge to a quality likelihood estimation exponentially decreases as the degree of amplification increases. The trend for the DBS is shown in Figure 11. As the degree of amplification increases from $1\times$ to $2\times$, the reduction in computational effort is approximately 83.5%.

Preliminary experimentation revealed that compromising STORAGE happens with the smallest likelihood. Thus, STORAGE is given the primary focus for the remainder of this investigation. Figure 12 shows that the confidence interval of STORAGE remains constant across all degrees of amplification. In all cases, the confidence interval falls within the established bounds of quality. Nonetheless, one can see in Figure 13 that the confidence interval of FILES widens as the degree of amplification

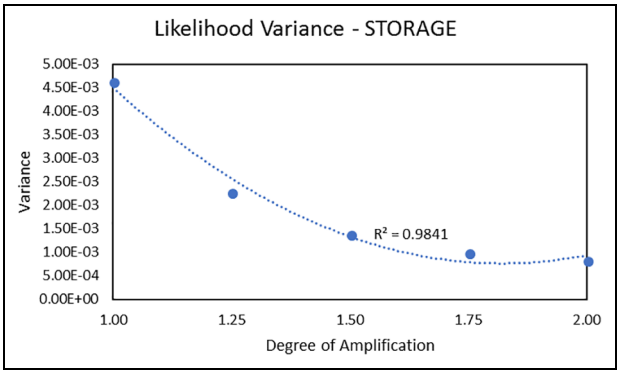

Figure 14. Convergence in the depth-based case – STORAGE variance.

increases. The trend seen in FILES is repeated for DB01 and DB02.

The variance in estimates of STORAGE's likelihood can be seen in Figure 14. As the degree of amplification

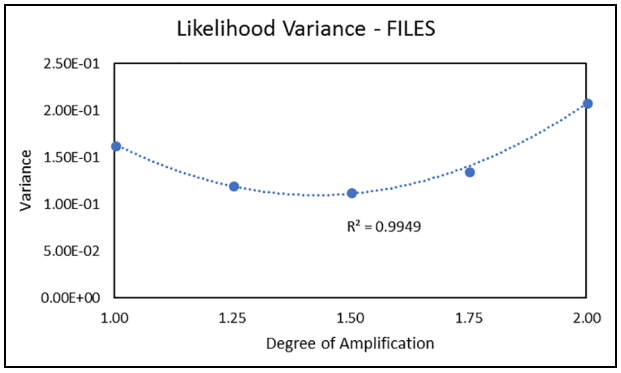

Figure 15. Convergence in the depth-based case – FILES variance.

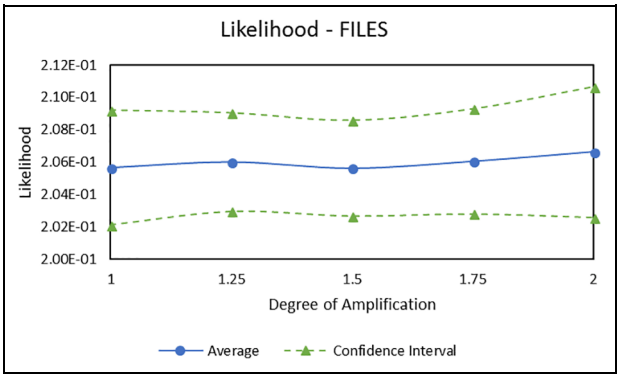

Figure 17. Static in the depth-based case – FILES likelihood.

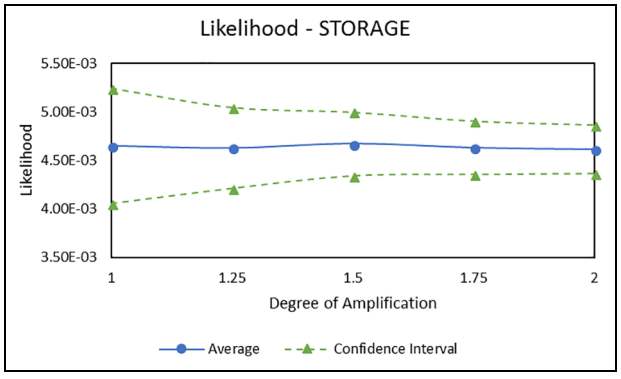

Figure 16. Static in the depth-based case – STORAGE likelihood.

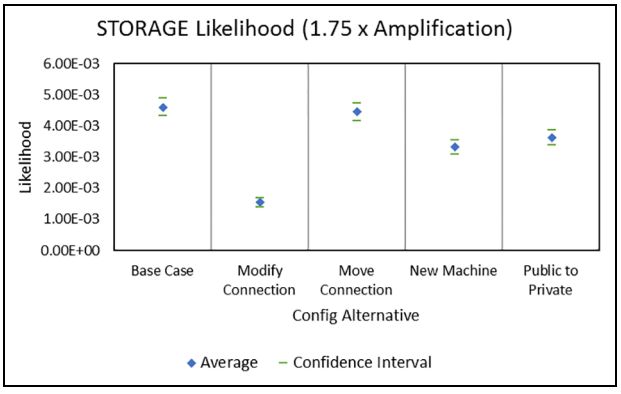

Figure 18. Configuration likelihood comparison in the depth-based case – STORAGE.

increases, the variance exponentially decreases. This variance trend for STORAGE corresponds to the trend seen for the computational efficiency. By contrast, the variance of FILES, seen in Figure 15, appears to be parabolic. Once again, the patterns in variance seen for FILES are repeated for DB01 and DB02.

The same trends in likelihood and variance seen for the DBS are also observed for the BBS, although the BBS yields different estimates. In addition, the number of trials required for convergence is several orders of magnitude greater for the BBS.

5.2 Estimate quality

When utilizing a static number of trials under the DBS, the halfwidth of the confidence intervals for STORAGE's likelihood are shown to decrease as the degree of amplification increases. This trend can be observed in Figure 16. The confidence interval for FILES first narrows and then widens again as the degree of amplification increases, as shown in Figure 17. The pattern in confidence interval seen for FILES is repeated for DB01 and DB02.

The variance patterns seen using a static number of trials mirrors trends seen when assessing computational efficiency. In addition, similar trends in likelihood and variance can be seen for the BBS, albeit with different estimates for each.

5.3 Risk assessment for reconfigurations

The effects of reconfiguration alternatives on the likelihood of compromising STORAGE when using the DBS are seen in Figure 18. An amplification level of 1.75 × is utilized alongside a static number of trials. All cases except the ''Move Connection'' yielded a significant reduction in likelihood when compared to the base case. The ''Modify Connection'' alternative displayed the largest decrease in likelihood. ''New Machine'' and ''Public to Private'' did not differ significantly from each other. Analysis of Figure 19 shows that there can be some unintended consequences to reconfiguration alternatives that seek to offer additional

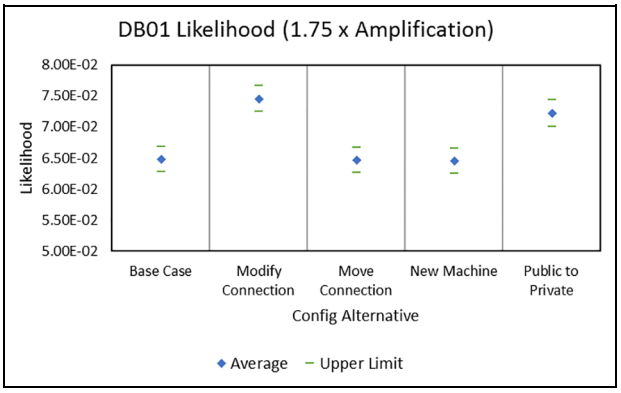

Figure 19. Configuration likelihood comparison in the depth-based case – DB01.

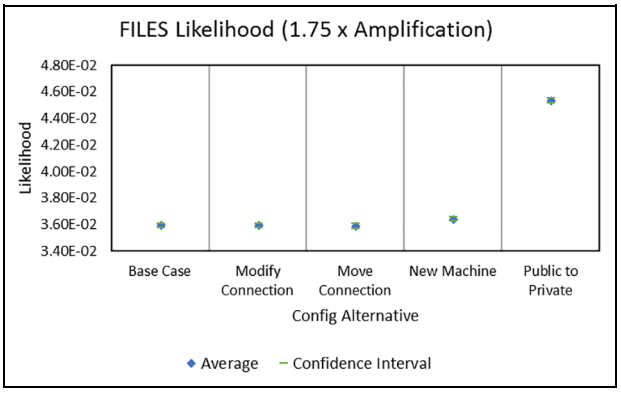

Figure 21. Configuration likelihood comparison in the breadth-based case – FILES.

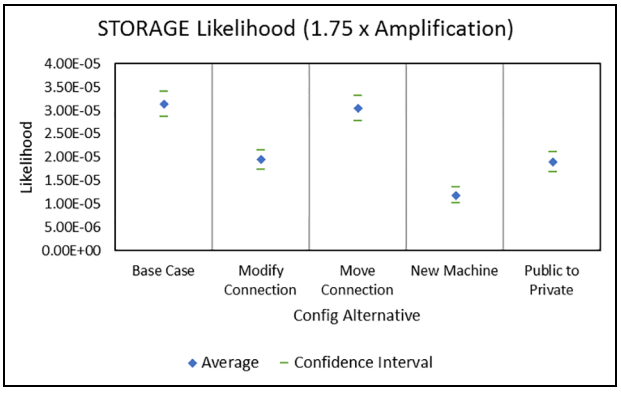

Figure 20. Configuration likelihood comparison in the breadth-based case – STORAGE.

protection to STORAGE. The ''Move Connection'' and ''Public to Private'' alternatives cause the likelihood of compromising DB01 to increase. The ''Public to Private'' case has similar impacts to the likelihood of compromising FILES and DB02.

Similar results are obtained with assessing reconfiguration with respect to the BBS. The primary differences, seen in Figure 20, show that the ''New Machine'' alternative provides the greatest likelihood of reduction. In addition, ''Public to Private'' causes a significant increase in the likelihood of compromising DB01, DB02, and FILES. This trend can be seen in Figure 21. Other reconfiguration alternatives under the BBS did not show a profound an impact on likelihood estimates.

5.4 Discussion

Utilization of the convergence method shows that the rate of convergence is dependent on the rarest event and is due to the simultaneous assessment of all events of interest. As the degree of amplification increases, the rarest event always has a confidence interval that just meets the bare minimum quality requirements. In contrast, all other events experience widening confidence intervals that converge to the same minimum quality requirements. These widening confidence intervals are ultimately impacted by the decreasing number of trials required before convergence. Recall that a halfwidth is equal to $t_{\frac{\alpha}{2},|\Lambda_z|-1}\frac{s_z}{\sqrt{\psi_z^c}}$. Thus, the rarest event can be seen as a limiting factor to the method.

Since the number of trials is static when assessing estimation quality, the confidence intervals are only impacted by variance. Amplification has the greatest impact on the variance of STORAGE's likelihood estimates.

Every reconfiguration alternative has the potential to have no effect or have unintended trade-offs. ''Move Connection'' never produced a meaningful change in the likelihood of compromising STORAGE. Therefore, a network analyst would not implement this alternative. In the case of the DBS, implementing ''Modify Connection'' increased the likelihood of compromising DB01. Making the connection between PR and FOLDING unidirectional caused PR to become a dead-end if not reaching PR from FOLDING. Therefore, an attacker would be more likely to reach a closer machine of interest; in the case of this network example, this would be DB01. A similar phenomenon can be seen when looking at ''Public to Private.'' Making BASTION private means that Subnet 4 is not publicly visible from the attacker's starting position. Thus, STORAGE is placed deeper within the network than the other machines of interest. From these results, one can glean that keeping BASTION publicly facing draws attacks away from the other machines of interest.

The ''Add Machine'' option operates off a similar principle as ''Public to Private.'' However, the key difference

Table 6. Depth – risk reduction.

Machine	Impact	Base case		New machine		% Change
		Likelihood	Risk	Likelihood	Risk	
FILES	6	2.06E-01	1.24E + 00	2.07E-01	1.24E + 00	0.22%
DB01	9	6.49E-02	5.84E-01	6.46E-02	5.82E-01	− 0.40%
DB02	11	3.24E-02	3.57E-01	3.26E-02	3.59E-01	0.48%
STORAGE	14	4.63E-03	6.49E-02	3.34E-03	4.68E-02	− 27.80%

Table 7. Breadth – risk reduction.

Machine	Impact	Base case		New machine		% Change
		Likelihood	Risk	Likelihood	Risk	
FILES	6	3.60E-02	2.16E-01	3.65E-02	2.19E-01	1.33%
DB01	9	2.32E-02	2.09E-01	2.35E-02	2.12E-01	1.39%
DB02	11	1.23E-02	1.36E-01	1.25E-02	1.37E-01	1.13%
STORAGE	14	3.16E-05	4.43E-04	1.20E-05	1.69E-04	− 61.92%

is that the original BASTION server is able to draw in attacks under the ''Add Machine'' case. BASTION2 then acts as a secondary barrier and funnels the attacker into a position where it must keep attacking or abandon its current path. As a result, ''Add Machine'' is the best alternative from the perspective of trade-offs in likelihood.

Tables 6 and 7 show the relative risk reduction when reconfiguring the base case to ''New Machine.'' The DBS and BBS both show a significant reduction in the risk of STORAGE becoming compromised. Nonetheless, there are slight increases in risk of both FILES and DB02 for both the DBS and BBS. In addition, the risk of DB01 decreases slightly for the DBS, whereas it slightly increases for the BBS.

Standard simulation produces wider confidence intervals than IS when utilizing the same number of trials. Thus, to generate estimates of similar quality as IS, more trials must be run under standard simulation. Once again, the rarest event is the limiting factor, since all standard simulation confidence intervals must be either equal in size to or tighter than those produced by IS. Thus, trials must be run until the confidence interval of STORAGE sufficiently narrows. The number of additional trials required by standard simulation is shown in Figure 22 for the DBS.

For comparison purposes, all reconfiguration alternatives must be run with the same number of trials. Therefore, the maximum number of additional trials required among all alternatives is utilized when generating confidence intervals. Figure 23 shows the difference in confidence intervals between standard simulation when running a differing number of trials. The difference in confidence intervals displays the utility of utilizing IS.

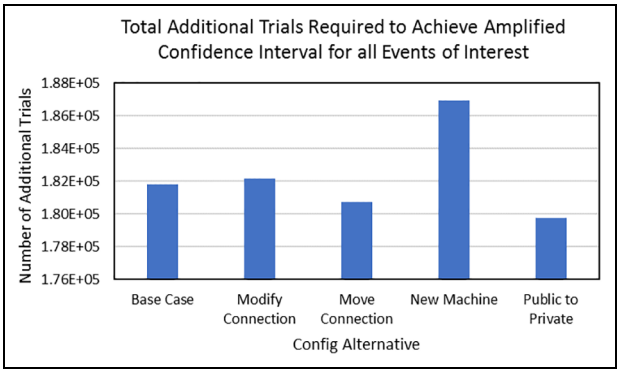

Figure 22. Additional trials required – depth.

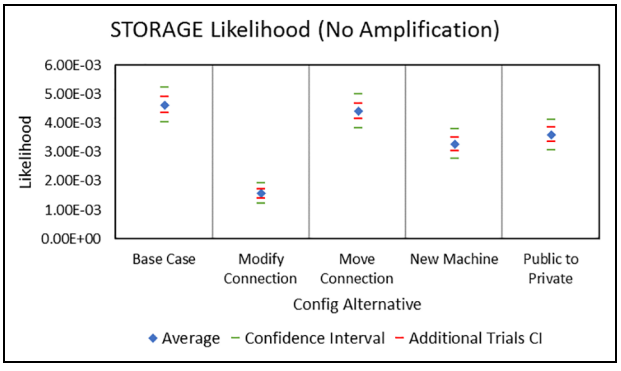

Figure 23. Standard simulation in the depth-based case: comparison – STORAGE.

Essentially, halfwidths can be decreased without needing to run more trials.

6 Conclusions and future work

Determining the risk of cyber attacks is a primary security interest. The investigation has applied a tailored IS methodology to a security framework, which is capable of analytically comparing network configurations against each other. The risk of data theft on a healthcare-oriented network was assessed utilizing this tailored method. Overall, it was able to show that the IS methodology is capable of delivering higher quality estimates with greater computational efficiency when compared to standard simulation.

So far, the IS methodology has only been tested with respect to a single network and a handful of potential reconfigurations. Testing the methodology with other networks, featuring diverse configuration alternatives, would be necessary for further vetting. Future applications may also explore the stochastic assignment of impact and attack success probability that may be dependent on the adversary's actions while moving through the network. The attacker logic may also be upgraded to accommodate more advanced search methodologies, which may reveal new trends between output statistics as a result of amplification. For example, the attacker may probabilistically or contextually shift between different search paradigms. The IS methodology may also be improved by allowing for individual vulnerability selection once a service is selected for targeting. This inclusion would necessitate some modification to the assignment of selection weights as well as the calculations of the aggregated selection probability. These relationships may be utilized in a small pilot study to automate the determination of optimal degrees of amplification.

The current investigation did not assess any performance-based trade-offs associated with reducing network risk as a result of network reconfiguration. Security specialists must strike a balance between network performance and network susceptibility. Thus, future investigations should define some heuristic to quantify the relative impact on network functionality when considering reconfiguration to reduce risk.

The simulation could be configured to store scenarios that result in an attacker reaching a machine of interest. Should reaching the machine be sufficiently rare, only a few scenarios will cause the event to occur. Thus, one can track this limited set of scenarios and gain insight into which configuration alternatives would be worth assessing. This type of additional information would enable a network analyst to make more informed decisions. Nonetheless, the scope of this investigation has established the advantages and potential of utilizing IS to assess network risk.

Declaration of conflicting interests

The authors have no conflicts of interest to declare.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by NSF Award #1526383.

ORCID iD

Alexander L Krall https://orcid.org/0000-0002-9753-1523

References

1. Yang SJ, Du H, Holsopple J, et al. Attack projection for predictive cyber situational awareness. In: *Advances in information security cyber defense and situational awareness*. 2014, pp.239–261. [AQ: 2]
2. Ross RS. Guide for conducting risk assessments. *Special Publication (NIST SP) - 800-30 Rev 1*. September 2012, 95. DOI:10.6028/NIST.SP.800-30r1.
3. McQueen MA, Boyer WF, Flynn MA, et al. Quantitative cyber risk reduction estimation methodology for a small SCADA control system. In: *proceedings of the 39th annual Hawaii international conference on system sciences*, pp.1–11.[AQ: 3]
4. Dell Secure Works. Lifecycle of an advanced persistent threat, http://www.redteamusa.com/PDF/Lifecycle%20of%20an%20Advanced%20Persistent%20Threat.pdf (2012). [AQ: 4]
5. Rege A, Singer B, Masceri N, et al. Measuring cyber intrusion chains, adaptive adversarial behavior, and group dynamics. In: *proceedings of the 12th international conference on cyber warfare and security*.[AQ: 5]
6. Baiardi F, Corò F, Tonelli F, et al. Automating the assessment of ICT risk. *J Inform Secur Appl* 2014; 19: 182–193.
7. Cheng P, Wang L, Jajodia S, et al. Aggregating CVSS base scores for semantics-rich network security metrics. In: *proceedings of the IEEE symposium on reliable distributed systems*, 2012, pp.31–40.[AQ: 6]
8. Zhang H, Lou F, Fu Y, et al. A conditional probability computation method for vulnerability exploitation based on CVSS. In: *2017 IEEE 2nd international conference on data science in cyberspace*, pp.238–241.[AQ: 7]
9. FIRST. Common Vulnerability Scoring System v3.0: User Guide (v1.5), 2018.
10. Muller S, Harpes C, Le Traon Y, et al. Efficiently computing the likelihoods of cyclically interdependent risk scenarios. *Comput Secur* 2017; 64: 59–68.
11. MITRE. Risk impact assessment and prioritization, http://www.mitre.org/publications/systems-engineering-guide/

acquisition-systems-engineering/risk-management/risk-impact-assessment-and-prioritization (2015). [AQ: 8]
12. Noel S, Jajodia S, Wang L, et al. Measuring security risk of networks using attack graphs. *Int J Next Gen Comput* 2010; 1: 135–147.
13. Dinh TN. Assessing attack vulnerability in networks with uncertainty. In: *IEEE conference on computer communications (INFOCOM)*, 2015, pp.2380–2388.[AQ: 9]
14. Wang L, Liu A and Jajodia S. Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts. *Comput Commun* 2006; 29: 2917–2933.
15. Noel S and Jajodia S. Metrics suite for network attack graph analytics. In: *cyber and information security research conference*, volume 9, pp.5–8.[AQ: 10]
16. Zhuang R, Zhang S and DeLoach S. Simulation-based approaches to studying effectiveness of moving-target network defense. In: *national symposium on moving target research*, 2012, (1), pp.1–12.[AQ: 11]
17. Fischer MJ, Masi DMB, Shortle JF, et al. Simulating non-stationary congestion systems using splitting with applications to cyber security. In: *proceedings of the 2010 winter simulation conference*, 2010, pp.2865–2875.[AQ: 12]
18. Shortle JF, Chen CH, Crain B, et al. Optimal splitting for rare-event simulation. *IIE Trans* 2012; 44: 352–367.
19. Masi DMB, Fischer MJ, Shortle JF, et al. Simulating network cyber attacks using splitting techniques. In: *proceedings of the 2011 winter simulation conference*, Jackson, pp.3212–3223.[AQ: 13]
20. Shahabuddin P. Importance sampling for the simulation of highly reliable Markovian systems. *Manag Sci* 1994; 40: 333–352.
21. De Boer PT, Kroese DP, Mannor S, et al. A tutorial on the cross-entropy method. *Ann Oper Res* 2005; 134: 19–67.
22. Krall AL, Kuhl ME, Moskal SF, et al. Assessing the likelihood of cyber network infiltration using rare-event simulation. In: *symposium series on computational intelligence*. [AQ: 14]
23. CPTC. Collegiate penetration testing competition, http://nationalcptc.org/#about (2018).

Author biographies

[AQ: 1]