Quantitative Analysis of the Mission Impact for Host-Level Cyber Defensive Mitigations

Neal Wagner, Cem Ş. Şahin, Diana Hanson, Jaime Pena, Era Vuksani and Brady Tello
MIT Lincoln Laboratory
244 Wood Street Lexington, MA 02420, USA
{neal.wagner, cem.sahin, dhanson, jdpena, era, brady.tello}@ll.mit.edu

ABSTRACT
Network devices and user accounts often provide an initial entry point for attackers who wish to gain a foothold on a network, pivot to other hosts, and ultimately disrupt an organizational mission and/or steal valuable network resources. Several host-level cyber defensive measures have been proposed to mitigate this threat. Although these mitigations are intuitively appealing from a security perspective, there is a lack of quantitative analysis addressing their effectiveness with respect to the network as a whole and the mission that the network supports. Testing these mitigations in an operational setting is prohibitively expensive, and thus modeling and simulation approaches are sought, due to their relative low cost. Our goal is to investigate the network-scale effects of various host-level defensive mitigations both from the standpoint of cyber security and mission impact. Our approach utilizes a hierarchical framework to model a complex cyber system at multiple, appropriate scales. Experiments serve to provide quantitative assessment of host-level mitigations from a complete network system perspective.

Author Keywords
Cyber defense; Hierarchical modeling; Mission impact; Multiscale systems; Network security

ACM Classification Keywords
C.2.0 COMPUTER-COMMUNICATION NETWORKS: Security and protection; I.6.3 SIMULATION AND MODELING: Applications

1. INTRODUCTION
As evidenced by the seemingly constant stream of news releases detailing the damage done by successful cyber compromises, government and private industry computer networks are under serious attack [8,29]. To help protect these systems and improve the overall state of cyber security, several defensive measures and mitigations have been proposed by various agencies such as the SANS Institute [9], Microsoft [19], and the Information Assurance Directorate of the National Security Agency [1], among others. These mitigations address different cyber threats and are designed to protect different aspects of a computer network. In a perfect environment, all recommended mitigations could be deployed to potentially maximize the security posture of a network. However most environments are resource-constrained and network administrators face the challenge of deciding which mitigations to deploy. Unfortunately, recommended mitigations are not ranked or prioritized and are not quantitatively evaluated with respect to the cyber threats that they are meant to provide protection against. This means that admins must rely on judgment to decide which mitigations are best for their network. Additionally, computer networks exist to support an organizational mission, and thus mitigations should be evaluated in the context of a complete network system where the goal is to maximize network security and minimize impacts to the mission.

Cyber systems are a mix of computerized processes and human actors, and thus are not deterministic. As such, a cyber environment under a given set of conditions (i.e. users, attackers, and defensive mitigations) is associated not with a single outcome, but rather with a distribution of outcomes representing the range of possible results. To properly assess a mitigation, it is therefore necessary to execute numerous tests to understand which are likely versus which are unlikely. Executing a large number of tests on a live network system can be prohibitively expensive. We therefore seek a modeling and simulation approach due to its relative low cost.

This paper examines two proposed mitigations designed to protect network hosts (i.e. devices) from exploitation: (i) Microsoft’s Enhanced Mitigation Experience Toolkit or EMET and (ii) Administrative Privilege Control. The purpose is to provide a quantitative assessment of the mitigations in the context of a complete network system and to consider mitigation effectiveness with respect to two fundamental network concerns, security and mission impact. To this end, a multiscale, hierarchical modeling framework is utilized to integrate relevant dynamics at the host and network scales. We also discuss the value of having a single performance measure to evaluate mitigations and describe a proposed unified metric to fulfill this purpose.

2. RELATED WORK
There have been several modeling and simulation (mod/sim) studies that have focused on network security in recent years. A large body of research has used mod/sim to investigate network intrusion detection and prevention. Some examples include [12, 16, 24, 25]. While these studies employ mod/sim to conduct an initial exploration of the efficacy of proposed models for intrusion detection, the focus is on network situational awareness rather than investigative network modeling.

Another area of research has focused on mod/sim applications for the purpose of investigative network modeling. A simulation to analyze a modeled network with respect to a...
specific network security protocol to automatically determine protocol compliance is given in [7]. A study combining discrete event simulation with meta-heuristic optimization to simulate network attacks and optimize network defenses is provided in [13]. An agent-based model investigating cooperative botnet attacks and corresponding defenses is presented in [14]. A Markov model is used to simulate worm attacks with simulation splitting techniques for efficient simulation of rare catastrophic network states are discussed in [17]. A model built using OMNeT++ to simulate distributed denial-of-service attacks on networks is presented in [20]. [23] utilizes an agent-based model to evaluate the performance of candidate security techniques that rely on a moving target strategy to defend against cyber attack. [26] and [28] provide epidemiological models to simulate malware propagation over networks. An agent-based simulation examines the effectiveness of security policies seeking to mitigate the threat posed by unauthorized hardware on a network in [27].

We utilize a mod/sim approach to investigate the network-scale effects of two widely known cyber defensive mitigations protecting at the host level. Our main contribution is to provide a quantitative assessment of these mitigations considering two fundamental network concerns: network security and mission impact. We also provide a single unified performance metric for measuring these concerns that is convenient and easily accessible to network analysts and admins.

3. HOST-LEVEL CYBER DEFENSIVE MITIGATIONS

Sophisticated and targeted cyber intrusions have increased in recent years. Understanding defensive mitigations and quantifying their performance help network administrators improve the security posture of networks against attack.

This paper investigates two defensive mitigations seeking to protect hosts on a network from exploitation, EMET [19] and Administrative Privilege Control [9]. Before we delve into the details of these mitigations, it is useful to define two key terms: vulnerability and exploit. A security vulnerability is a weakness in a product that could allow an attacker to compromise the integrity, availability, or confidentiality of that product [5]. Here, the term product can refer to software, hardware, firmware, or communication protocols among others but for the purposes of this paper we will restrict our focus to vulnerabilities in software. An exploit is then a method by which an attacker takes advantage of a vulnerability to breach the security of a computer system in violation of security policy [4]. Both mitigations considered provide protection by attempting to block a particular category of exploits. Ideally, when one of these mitigations is in place, all exploits of the addressed category are ineffective. We now describe these mitigations and the exploits that they are meant to block.

3.1 EMET Mitigation

The EMET mitigation is meant to protect unpatched software vulnerabilities from exploitation [11]. Here, the idea is that even when software vulnerabilities are present on a device, EMET may prevent an attack attempting to exploit these vulnerabilities. EMET consists of 12 security technologies focusing on preventing memory-based exploits [19]. A memory-based exploit is one that attempts to corrupt the memory of a device to cause unintended behavior and allow the attacker to gain control. When EMET is configured to utilize all of its component technologies and to protect all software on the device, then it potentially can render memory-based exploits ineffective. This means, in a best-case scenario, EMET can nullify all exploits of this category.

Of course in real-world situations, the best case is not always possible. While EMET can be theoretically configured to protect any Windows application, there are examples that are not compatible with one or more components of EMET [2] and, thus are not fully protected. Additionally, security researchers have proven that attacks can be crafted to bypass all EMET components [10]. Therefore, against a sufficiently sophisticated attack, EMET may provide little protection.

For this paper we consider the best-case scenario, where EMET is fully configured and faces common unsophisticated exploits. The goal here is to provide an upper bound on the effectiveness of EMET to protect unpatched vulnerabilities. For the remainder of this paper, when we refer to EMET, we are referring to this best-case scenario. Note that while EMET is designed to protect unpatched vulnerabilities from memory-based exploits, we evaluate its effectiveness, at the network scale, in protecting against all host-level exploits, memory-based or not, as there exist other types of exploits such as those that attack software or Internet browser vulnerabilities.

3.2 Administrative Privilege Control Mitigation

The Administrative Privilege Control (APC) mitigation is concerned with placing restrictions on accounts with specialized privileges and access to resources that are beyond those associated with a standard user account [9]. Administrator and server accounts are common examples of such privileged accounts. The goal of this mitigation is to control the number and types of users who are given access to such privileged accounts, to protect these accounts from invalid or fraudulent authentication, and to restrict the privileges that these accounts have to reduce the amount of damage that can be done in the event that they are compromised. APC consists of nine different suggested threat-mitigating features including disallowing standard users from access to admin privileges on their local devices, requiring longer and more complex passwords for privileged accounts, and limiting the number of devices that are allowed to host privileged accounts, among others. We focus on one of the nine suggested mitigation features: Ensuring privileged accounts do not have access to email and are disallowed from browsing the Internet, which we refer as Restricted Email and Internet Access (REI).

The idea is that privileged accounts should not be allowed to do tasks that are not necessary for their intended purpose and expose them to additional risk of compromise. Accessing emails and browsing the Internet are general privileges that are usually associated with standard user accounts. Email and Internet browsing privileges expose an account to higher risk because they may involve processing malicious content, for example by downloading and executing a malicious email attachment or by browsing to a malicious website. When a device processes malicious content, that content can exploit...
software vulnerabilities present on the device and allow the attacker to gain control. When a device with a privileged account is allowed to access Email and browse the Internet, it has a higher chance of being compromised and, once compromised, allows an attacker to access additional privileges and resources that wouldn’t normally be accessible if the compromise had taken place on a device with a standard user account.

The purpose of the REI mitigation is to protect devices that have privileged accounts by preventing exploits arriving from external sources via Email and Internet browsing. When the REI mitigation is enforced for privileged accounts, these accounts are not susceptible to such exploits and, thus are less likely to be compromised and used by an attacker to spread to other network devices. Note that REI does not prevent all exploits, only those that arrive as a result of a user accessing Email or browsing the Internet. Other exploits take advantage of software vulnerabilities in a more direct fashion, for example by exploiting vulnerabilities present in server software or the environment in which this software is executed. A well-known example of such a direct exploit is Shellshock, which targeted a vulnerability in the shell environment that is commonly used to execute Apache server commands [3].

In this paper we classify the exploits that REI prevents and those that REI does not prevent as client-side and server-side exploits, respectively as done in [15]. When REI is enforced on a device with a privileged account, this device is protected against client-side exploits (i.e. those that arrive via Email and Internet browsing) but is not protected against server-side exploits (i.e. those that do not require a user to access Email or browse the Internet to enable exploit execution).

4. MULTISCALE, HIERARCHICAL MODEL

We wish to quantitatively assess the effectiveness of the EMET and REI mitigations in the context of a complete network system. We employ a multiscale model characterizing dynamics at the host and network scales. A hierarchical framework is leveraged in which host-scale dynamics are captured separately in a single model and simulation results from this model are then used to inform a network model.

Fig. 1 gives a graphical overview of this framework and presents a host model that is used to characterize host-scale dynamics of attack and defense. Simulation runs are executed on this host model, results are aggregated, and these results are used to parameterize our network model capturing an abstracted network system complete with attacker and defender actors, mission users and associated operations. The hierarchy of this multiscale model allows us to substitute different components at the host model without having to change the underlying implementation of the network model. It also provides for quicker simulation execution times (due to reduced complexity at the larger network scale) and less model implementation effort (due to the modularity gained by splitting the overall model into host and network components). The following describes the host and network models.

Host Model: At the host scale we characterize a host with a set of vulnerabilities, an attacker attempting to exploit these vulnerabilities, and the host-level defensive mitigation used to protect against this threat. The attack model is based on the threat of exploitation of known software vulnerabilities on a target host. The threat characterizes an attacker who is assumed to have exploits for known vulnerabilities at his/her disposal and utilizes these to attempt to compromise the host. Devices with more vulnerabilities or more severe vulnerabilities are more likely to be compromised when attacks arrive. We use the probabilistic model given in [15] to compute the probability of device compromise ($p_{comp}(device)$) when an attack arrives at a host with one or more vulnerabilities.

$$p_{comp}(device) = 1 - \prod_{v \in V} (1 - p_{comp}(v))$$

where $v$ is a single vulnerability and varies over all vulnerabilities ($V$) on the device, $p_{comp}(v) = (\frac{CVSS(v)}{10})^2$ is the probability of compromise for a given vulnerability $v$, CVSS($v$) is the CVSS score for a vulnerability $v$, and $p_{comp}(device)$ is the probability of device compromise. The defense model characterizes a reduction in $p_{comp}(device)$ when a host-level mitigation is in place. When emet is modeled, the reduction is specified by the type of exploits blocked by emet, namely memory-based exploits (described in Section 3). When REI is modeled, the reduction is specified by exploits blocked by REI, that is client-side exploits (also in Section 3).

Network Model: At the network scale we utilize an agent-based model to characterize network dynamics of attack, defense, and mission operations. The attack model is based on the network-scale threat developed in [15], which characterizes an attacker who scans the network, probabilistically compromises vulnerable devices, and upon success utilizes compromised internal devices to further attack other network devices. The defense model characterizes a defender scan that detects and cleanses compromised network devices. Fig. 2

![Figure 2. Network Model Defense: Network scanning for compromised devices. Compromise detection triggers a cleansing process.](image)

provides a graphical depiction of this model. Here, the network is periodically scanned to detect compromised devices. Detected devices are taken offline for cleansing and then returned to the network in their original (uncompromised) state.

A mission model characterizes a network-supported, time-sensitive mission allowing us to examine a defensive mitigation’s ability to protect the mission from attack. Any delay to the mission’s completion is undesired. This model is based on a military-style Air Operations Center (AOC) in which requests for air operations are gathered and processed into flight plans for air operations. A mission team involves three mission users and three database servers existing on different network devices. We assume each mission device has at most one mission role. The abstracted AOC mission shown in Fig. 3, where the mission users pass a payload from Database 1 to Database 3. The network includes $N_m$ mission user teams sharing three mission servers, meaning that there are $3N_m + 3$ total mission devices. Mission users require a fixed amount of uninterrupted time, $t_M$, to operate on the payload before passing it to the next step. Non-mission operations, such as sending and receiving benign traffic can occur during this time, but suffering a compromise to a mission device will delay the mission until that device is cleansed.

![Figure 3. Abstracted AOC Mission Model: Three mission users utilize three network hosts to interact with three database servers to execute the mission.](image)

Host-Network Model Integration: As mentioned above the host and network models are separate models in a hierarchy in which host model simulation results are used to set parameters in the network model. For the simulations conducted in this paper, we execute three host model scenarios: one where a vulnerable host has no defensive mitigation, one in which the host is protected by EMET, and one with the host protected by REI (the details are provided in Section 6). We execute thousands of runs on the host model for each scenario, record the number of successful attacks, and compute the expected probability of host compromise when attacked. This result is then used to specify the expected probability of compromise for hosts of that type in the network model. When a device is scanned by the attacker in the network model, it is probabilistically compromised based on this parameter setting.

5. EFFECTIVENESS MEASURES

Our multiscale model captures attacks on device availability. A successful attack on a device cause that device to be unavailable while it is compromised and while it is being cleansed. We measure the mitigation effectiveness with respect to two aspects: (i) system security (given as a index) and (ii) mission impact (measured as delay).

Definition 1. System security index, $s_i$, is the expected ratio of device up-time to total time, normalized to $[0, 1]$.

The higher the security index, the better the security performance is. It is calculated as:

$$s_i = \frac{\sum_{i=1}^{N} T - t_{\text{down}}^i}{N}$$  (2)

where $t_{\text{down}}^i = t_{\text{comp}}^i + t_{\text{cleanse}}^i$ is the total down time for device $i$, $T$ is the total simulation time, and $N$ is the number of devices. A network device $i$ can be down due to either compromise or cleansing. $t_{\text{comp}}^i$ and $t_{\text{cleanse}}^i$ are total compromise time and cleansing time for device $i$, respectively.

Definition 2. Mission delay, $m_d$, is the expected total time of device compromise $t_{\text{delay}}(\text{comp})$ and cleanse time $t_{\text{delay}}(\text{cleanse})$.

As discussed in Section 4, a crucial component of the network-scale model is the ability to run time-critical missions. Each user has a mission task to complete and incurred delay is not desirable. If a mission-critical device is compromised, the mission halts until the device is cleansed. To observe the effects of the mitigation on the time-critical missions, we calculate the mission delay as shown below:

$$m_d = t_{\text{delay}}(\text{comp}) + t_{\text{delay}}(\text{cleanse})$$  (3)

The expected total time of device compromise and cleanse time in Eq. 3 is calculated as:

$$t_{\text{delay}}(\text{comp}) = \sum_{i=1}^{M} \hat{t}_i(\text{comp}),\quad M$$  (4)

$$t_{\text{delay}}(\text{cleanse}) = \sum_{i=1}^{M} \hat{t}_i(\text{cleanse}),\quad M$$

where $\hat{t}_i(\text{comp})$ and $\hat{t}_i(\text{cleanse})$ are the delays for mission $i$ due to compromise and cleansing, respectively. $M$ is the number of missions executed.

5.1 Unified Measure for Mitigation Effectiveness

The system security index and the mission delay evaluate different effects of a mitigation. When we want to evaluate the effectiveness of a mitigation and/or to compare the effectiveness of multiple mitigations, it is convenient to have single measure for evaluation.

Definition 3. The unified metric, $m_g$, is a measure to characterize the security and mission impact inherent to a simulated network environment. The metric includes effects of mean, median, and variance of results from multiple simulation runs, normalized to $[0, 1]$.

To unify the metrics, we propose the following approach:

$$s_M = \int f_M(s_M)ds_M$$  
$$s_{\text{coef}} = \int f_M(s_{\text{coef}})ds_{\text{coef}}$$  
$$m_M = \int f_M(m_M(t))dt$$  
$$m_{\text{coef}} = \int f_M(m_{\text{coef}}(t))dt$$

$$s_g = \frac{s_M - s_{\text{coef}}}{\max(s_M, s_{\text{coef}})}$$  
$$m_g = \frac{m_M - m_{\text{coef}}}{\max(m_M, m_{\text{coef}})}$$

$$m_g = f_M(w_1, s_g, w_2, m_d)$$  (5)
where $s_{i,M}$ and $s_{i,noM}$ represent $s_i$ (Eq. 2) with and without mitigation, respectively and $m_{d,M}$ and $m_{d,noM}$ are $m_d$ (Eq. 3) with and without mitigation, respectively. $f_1$ is a function $f_1: X \rightarrow f_1^*$ that takes an arbitrary input $X$, which might be $s_{i,M}$ and $s_{i,noM}$, and outputs an approximation function $f_1^*$. $f_2$ is a function $f_2: f_1^* \rightarrow f_2^*$ that takes an arbitrary function $f_1^*$, and maps into an approximation function $f_2^*$. $f_3$ is a function $f_3: Y \rightarrow f_3^*$ that takes an arbitrary input $Y$, which might be $m_{d,M}$ and $m_{d,noM}$, and outputs an approximation function $f_3^*$. $f_4$ is a function $f_4: f_3^* \rightarrow f_4^*$ that takes an arbitrary function $f_3^*$, and maps into an approximation function $f_4^*$. $s_y$ and $m_{d,y}$ are normalized enhanced security index and mission delay, respectively. $f_5$ is a function $f_5: \{w_1, s_y, m_{d,y}\} \rightarrow m_y$ that takes the user defined importance factors $(w_1, s_y, m_{d,y})$, and, $m_y$ generates and the unified performance measure $m_y$. Note that $w_1 + w_2$ should be one so that $m_y \in [-1, +1]$.

The proposed measure given in Eq. 5 combines $s_i$ (see Def. 1) and $m_{d,y}$ (see Def. 2) with/without a mitigation to provide a unified measure for effectiveness.

6. EXPERIMENTS

We implemented our simulation framework in NetLog modeling environment [22]. Data aggregation across simulation runs and the calculation of statistical measures was carried out using MATLAB release 2014b [18] and Python 2.7. Our framework provides the capability to instantiate cyber entity models for a variety of cyber-relevant purposes within a simulated environment.

![Figure 4. ExploitDB data mining results for EMET vs Non-EMET-relevant exploits via (a): scripted keyword classification and (b): random samples used to categorize all uncategorized exploits in (a).](image)

6.1 Simulation Parameters

We model a representative network environment that supports the AOC mission (see Section 4). For our representative network, we consider a Class C-sized network with 250 hosts, all with the Windows operating system. 85% of network devices host standard user accounts, 15% host server accounts, and 5% host administrator (admin.) accounts (server and admin accounts are considered as privileged). The AOC mission modeled (comparable to military-style missions) takes three days to complete if unhindered where each of the three mission users takes one day to complete his/her task.

A recent survey of vulnerabilities exploited in the wild leveraged ~ 300M intrusion telemetry reports collected from 6M hosts [21] and provides two key results that we use to parameterize our model: (i) Windows devices are attacked ~ 4 times per year on average over a collection of common Windows software products and (ii) ~ 15% of known vulnerabilities are actually exploited in real-world attacks. The former result is used to set the rate of attacks that arrive at the network (~ 3 per day for a Class C network), while the latter result is used to specify the percentage of host vulnerabilities that can be exploited. We collected vulnerability data from nearly 1M hosts to estimate the number of vulnerabilities per device and their severity levels. Based on our dataset, there are ~ 24 vulnerabilities per device with 66% of these of high severity (based on their CVSS scores) where, on average, 3.6 of these are exploitable by a given attack on a given device. For our experiments the defender scans each network device once a day for compromise and cleanses detected compromises. As discussed in Section 4, compromised devices attack other devices. In our experiments these devices send attacks at a rate of 5 per day. To capture the escalated privileges associated with admin. accounts, compromised admin. devices have probability of success = 1 when they send attacks; other compromised devices have probability of attack success as determined by host model experiments as described in Section 4.

1500 simulation runs were executed for each parameter set with 40 missions (executed in parallel) in each run.

**EMET Specific Parameters**

EMET prevents vulnerabilities in Windows-based applications from being successfully exploited. EMET uses 12 different security technologies, listed in Table 1, to protect software vulnerabilities from exploitation.

We scrape a public catalogue of known exploits, ExploitDB [6], for exploits relevant to EMET. Recall (Section 3) that EMET only protects against memory-based exploits. We used a scripted keyword search to initially classify exploits as memory-based, not memory-based, or undetermined. As shown in Fig. 4 (a), 42% of Windows exploits are EMET-relevant, 2% are not EMET-relevant, and 56% could not be categorized via keyword search. To categorize these unknown exploits, we collected a random sample of the 4080 unclassified exploits for manual inspection. Our final results after keyword search and manual inspection, shown in Fig. 4 (b), give 57% of exploits as not relevant to EMET (43% as EMET-relevant) with a 95% confidence interval. This result is used to set the reduction in probability of compromise when an exploit meets a vulnerability for devices protected by EMET.

The percentages of all windows exploits prevented by individual EMET technology modules are listed in Table 1. While this breakdown is not necessary for our experiments, it is pro-

<table>
<thead>
<tr>
<th>EMET Module</th>
<th>EMET-relevant Exploits</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEP</td>
<td>1.74%</td>
</tr>
<tr>
<td>SEHOP</td>
<td>11.96%</td>
</tr>
<tr>
<td>EAF, NullPage, MemProt</td>
<td>0.00%</td>
</tr>
<tr>
<td>HeapSpray</td>
<td>6.42%</td>
</tr>
<tr>
<td>Mandatory &amp; BottomUp ASLR</td>
<td>87.91%</td>
</tr>
<tr>
<td>LoadLib</td>
<td>0.29%</td>
</tr>
<tr>
<td>Caller, SimExecFlow, StackPivot</td>
<td>2.03%</td>
</tr>
</tbody>
</table>

Table 1. Results in Fig. 4 (b) are categorized by applicable EMET module.

This material is based upon work supported under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Air Force.

© 2016 Massachusetts Institute of Technology.
vided for informational purposes. Most exploits relevant to EMET are addressed by the Mandatory and BottomUp ASLR modules (see Table reftable:emet). Note that the total of all percentages in the table does not sum to 100%; this is because in some cases an exploit may be blocked by more than one EMET module.

REI Specific Parameters

REI protects devices that host privileged accounts from client-side exploits (see Section 3). For our experiments, we wish to characterize the relative proportion of client-side to server-side exploits. We leverage the ExploitDB [6] to categorize exploits, this time as client- or server-side.

ExploitDB already categorizes exploits as local, remote, web-based, or denial of service (DoS). Local exploits require victim to execute malicious software or file (e.g. infected .pdf file). Remote exploits are executed remotely by the attacker. Web-based exploits take advantage of vulnerabilities in web-browsers and their plugins (e.g. Java, Flash Player, Adobe Reader). DoS exploits seek to interrupt services on a host connected to the internet. Based on these categories we initially assume: (i) local exploits are client-side (require action on part of the victim), (ii) remote exploits are server-side (executed directly by attacker), (iii) web-based exploits are client-side (victim must browse to infected site or download malicious content), and (iv) DoS exploits are server-side (attacker makes numerous requests to a service). We collect a random sample from all Windows exploits for manual inspection. Here, manual inspection is more time-consuming than for EMET exploit classification as actual exploit source code must be inspected. Our results yield that 43% of exploits are client-side and thus protected by REI, but due to resource constraints we were unable to inspect a large number of exploits and these results come with a 20% margin of error. For our experiments we wish to capture a best-case scenario for REI’s ability to mitigate attacks because, as discussed in Section 3, we are also considering a best-case scenario for EMET. Thus we use the most optimistic value given by the 20% error margin, 63%, to set the reduction in probability of compromise when an exploit meets a vulnerability for devices that are protected by REI. Note that even when REI is used, not all devices are protected by REI only those that host privileged accounts (i.e. admin. or server accounts).

6.2 Simulation Results and Analysis

As discussed in Section 3, we wish to compare three network defense scenarios: (i) no mitigation, (ii) EMET, and (iii) REI. Figs. 5 and 6 show the mean and the standard deviation of simulation results for these scenarios for security and mission impact, respectively. From Fig. 5, the network security is slightly improved when REI OR EMET is used with EMET having the best security of the three scenarios. However, these differences are not statistically significant.

Fig. 6 gives simulation results for mission impact (i.e. mission delay). Delay is measured in time units (ticks) in which 1000 time ticks is equivalent to 1 day. For example, the no mitigation scenario result shows 234.62 ticks (or 0.235 days) of expected mission delay with a standard deviation of \( \sigma = 466.47 \) time ticks (or 0.466 days). Although the differences between the three scenarios are not statistically significant, EMET and REI decrease the expected mission delay by 20.03% and 10.95%, respectively. The standard deviations of results for both mitigation scenarios are lower than seen for the no mitigation scenario. Overall, EMET performs slightly better than REI in terms of average mission delay and variance of mission performance.

In Section 5.1, we introduced a unified measure for comparing the effectiveness of mitigations. The goal of \( m_g \) is to provide a single measure for effectiveness. The proposed approximation functions \( f_1, f_2, f_3, \) and \( f_4 \) (see Eq. 5) are general functions mapping simulation experiment results into functions that are integrable. \( f_5 \) takes the normalized enhanced security index and mission delay values and merges with the importance factors to generate single evaluation value. In this paper, we incorporate both mean and variance into the unified measure as shown below:

\[
\begin{align*}
    f_1 &: \bigoplus_{i=1}^{\infty} N_i \times (s_{i,M} \vee s_{i,noM}) \rightarrow \bigoplus_{k=1}^{n} s_k = S_k \\
    f_2 &: \frac{1}{n} \times \sum_{i=1}^{n} (S_{k,i}) \times N\left(\frac{1}{n} \times \sum_{i=1}^{n} (S_{k,i})\right) \\
    f_3 &: \bigoplus_{i=1}^{\infty} N_i \times (md_{i,M} \vee md_{i,noM}) \rightarrow \bigoplus_{k=1}^{n} md_k = Md_k \\
    f_4 &: \frac{1}{n} \times \sum_{i=1}^{n} (Md_{k,i}) \times N\left(\frac{1}{n} \times \sum_{i=1}^{n} (Md_{k,i})\right) \\
    f_5 &: \frac{1}{2} \times s_g + \frac{1}{2} \times md_g 
\end{align*}
\]

where \( \bigoplus_{i=1}^{\infty} N_i \) represents all possible histograms and \( f_1 \) and \( f_3 \) take the security index and the mission delay run results for the no mitigation, EMET, and REI mitigation scenarios and create histograms. \( f_2 \) and \( f_4 \) fit the Normal distribution to each histogram and then multiply by the mean of each measurement. After integrating the area under the curve, giving
the probability density function, the unified measure is computed as the average of the $s_g$ and $md_g$.

\begin{equation}
    s_g = \frac{(s_M \times \mathbb{E}(s_{i,M}) - s_{noM} \times \mathbb{E}(s_{i,noM}))}{\max(s_M \times \mathbb{E}(s_{i,k}), s_{noM} \times \mathbb{E}(s_{i,noM}))}
\end{equation}

where the mean calculation shown in $f_2$ in Eq. 6 is carried out into this equation for simplicity.

Based on Eq. 7, the enhanced security index for EMET is $s_{g|E} = 0.0105 \times 0.966 - 0.910 \times 0.960 = 0.0304$ that indicates the security index improvement due to the mitigation is small. REI also provides a small $s_g$ improvement compared to the no mitigation scenario as $s_{g|A} = 0.966 \times 0.962 - 0.910 \times 0.960 = 0.0304$. However, this improvement is two orders of magnitude better than $s_{g|E}$. This difference is due to the relative volatility of $s_g$ for the EMET mitigation scenario compared to the REI mitigation scenario. Fig. 8 gives a box-plot depiction of the variance for REI and EMET results. From the figure, EMET runs have wider result variance due to more extreme outliers. Although the mean and standard deviation for EMET results are slightly better than for REI, our enhanced security measure punishes EMET’s volatility.

Fig. 9 (a), (b), and (c) illustrate the histogrammed mission impact results with the Normal distribution approximation (see Eq. 6). The enhanced mission delay is:

\begin{equation}
    md_g = \frac{(md_{noM} \times \mathbb{E}(md_{noM}) - md_M \times \mathbb{E}(md_M))}{\max(md_{noM} \times \mathbb{E}(md_{noM}), md_M \times \mathbb{E}(md_M))}
\end{equation}

where the mean calculation shown in $f_4$ in Eq. 6 is carried out into $md_g$ for simplicity. Using this equation, the enhanced mission delay for EMET results is $md_{g|E} = \frac{(0.092 \times 234.62 - 0.6895 \times 204.442)}{\max(162.483, 147.614)} = 0.131$. This means that the EMET mitigation scenario gives small improvement to mission impact when compared to the no mitigation scenario. REI results also indicate a small improvement to mission impact relative to the no mitigation case as $md_{g|A} = \frac{(0.092 \times 234.62 - 0.678 \times 208.939)}{\max(162.483, 141.698)} = 0.127$.

We assume that security and mission impact are equally important aspects of mitigation effectiveness so the weight factors in Eq. 5 are set to 0.5 (i.e., $w_1 = w_2 = 0.5$). The unified effectiveness measure for EMET and REI mitigations are $m_{g|E} = \frac{1}{2} \times 0.0062 + \frac{1}{2} \times 0.131 = 0.0686$ and $m_{g|A} = \frac{1}{2} \times 0.0882 + \frac{1}{2} \times 0.127 = 0.1076$, respectively. From a practical standpoint, network administrators can view these results as a measurement of the gain in effectiveness at the network.
scale due to the mitigation. Thus, EMET and REI individually provide \(~7\%\) and \(~11\%\) gain in effectiveness, respectively. While these are seemingly modest gains, it is important to note that defensive mitigations are not meant to be used in isolation, but rather in combination as part of a layered defense. Thus when utilized as part of a greater defensive policy, EMET and REI can provide meaningful improvements to the security posture of a network against attack.

7. CONCLUSION
This paper presents a multiscale agent-based simulation model designed to evaluate two well-known host-level cyber defense mitigations, EMET and REI. We quantify the network-scale effects of these mitigations from the perspectives of security and mission impact. We also introduce a novel, unified measure of mitigation effectiveness that takes into account both of these perspectives and considers mean and variance of results from simulation runs. Experimental results indicate that while both EMET and REI provide modest gains in effectiveness when used in isolation, they could be beneficial when used in combination with other mitigations as part of a greater defensive policy to improve the security posture of a network.

Future work is focused on developing models to evaluate other defensive mitigations both at the host and network levels. We also plan to test the inclusion of new functions to improve our unified effectiveness measure.

REFERENCES
22. Netlogo (version 5.1.0), 2015.