Check for updates

Special Issue Article

JDMS

Journal of Defense Modeling and Simulation: Applications, Methodology, Technology
2018, Vol. 15(1) 13–29
© The Author(s) 2017
DOI: 10.1177/1548512917725408
journals.sagepub.com/home/dms
SAGE

Cyber threat assessment via attack scenario simulation using an integrated adversary and network modeling approach

Stephen Moskal[1], Shanchieh Jay Yang[1], and Michael E Kuhl[2]

Abstract

Existing research on cyber threat assessment focuses on analyzing the network vulnerabilities and producing possible attack graphs. Cyber attacks in real-world enterprise networks, however, vary significantly due to not only network and system configurations, but also the attacker's strategies. This work proposes a cyber-based attacker behavior model (ABM) in conjunction with the Cyber Attack Scenario and Network Defense Simulator to model the interaction between the network and the attackers. The ABM leverages a knowledge-based design and factors in the capability, opportunity, intent, preference, and Cyber Attack Kill Chain integration to model various types of attackers. By varying the types of attackers and the network configurations, and simulating their interactions, we present a method to measure the overall network security against cyber attackers under different scenarios. Simulation results based on four attacker types on two network configurations are shown to demonstrate how different attacker behaviors may lead to different ways to penetrate a network, and how a single misconfiguration may impact network security.

Keywords

Cyber attack simulation, adversary behavior modeling, fuzzy logic

1 Introduction

As the impact and prevalence of cyber attacks have grown, businesses have taken action toward network security by employing staff, tools, and services to better protect their business, data, and customers. Despite the billions of dollars spent on the prevention of cyber attacks, high-profile data breaches are becoming more common, affecting not only the businesses but also the millions of innocent customers of these businesses. Cyber defense strategies, such as intrusion detection systems (IDSs), strict firewall policies, and access controls, are common defense approaches. However, the implementation of these methods requires expertise and extensive configuration of complex rules.[1] Ginter[2] indicates that firewall configurations contain on average 793 rules for a typical enterprise network and this rule set is often modified multiple times a month. The misconfiguration of the firewall rules of a router was identified as a key cause of the 2015 United Airlines outage that resulted in the grounding of all flights for almost 2 hours and a tremendous amount of negative publicity.[3]

The dependence on the stability and security of computer networks is driving the demand for pre-emptive cyber analytics to aid in the discovery of potential cyber threats or vulnerabilities rather than relying solely on threat detection (or in some cases cyber forensic analysis.) Cyber

[1] Department of Computer Engineering, Rochester Institute of Technology, USA
[2] Department of Industrial and Systems Engineering, Rochester Institute of Technology, USA

Corresponding author:
Stephen Moskal, Department of Computer Engineering, Kate Gleason College of Engineering, Rochester Institute of Technology, 83 Lomb Memorial Drive, Rochester, NY 14623-5603, USA.
Email: sfm5015@rit.edu

analytics, such as firewall policy analysis,[4,5] penetration tests, and physical cyber attack simulations (white-hat hackers actually attack the network to discover vulnerabilities),[6] may be conducted to increase the defense's preparedness for cyber attacks. Currently there is no established method for measuring the effectiveness or risk of defense strategies.[7] Although penetration tests and live simulations may be effective at providing insight into network vulnerabilities, they are impractical to conduct every time there is a defense policy change. Penetration tests typically measure the risk or severity of policies by the ease of exploiting vulnerabilities and potential breaches,[8] but only represent the capability of the tester and what he/she is able to identify, which may not be complete.[9] With the diverse set of attacker types and skill levels that have been identified,[10] as well as attackers and attacker behaviors that are yet to be identified (now or in the future), a methodology is needed that can take into consideration a full range of hacker behaviors when assessing the network's security and risks.

Live simulations and penetration tests typically provide a limited but detailed set of cyber attack scenario data for one network and specific behaviors. Synthesizing this data over alternative network configurations and attacker types/behaviors could provide more accurate and robust security assessments due to the capability of understanding how the vulnerabilities could be realized. However, accurately modeling and representing a full range of cyber attacks and cyber attack behaviors is complex and daunting task due to the sheer number of network possibilities and choices an attacker has to make. This work reduces the complexity of the description of cyber attackers by employing a set of cyber-based contextual models representing the stages of cyber attacks, vulnerability modeling, and a portrayal of the attacker's knowledge of the target network to generate representative cyber attack scenario data in a realistic and efficient manner. We propose a framework to represent cyber attacker behaviors and apply the methodology to an existing cyber attack simulator to measure the effects network configurations and cyber attacker behaviors might have on the overall network security. The contributions of this work are as follows:

- define an attacker behavior model (ABM) using a capability, opportunity, intent (COI) model with additional preferences to differentiate among attacker types;
- incorporate the notion of a developing knowledge base for the attacker to determine the next attack action; and
- investigate the effects network configuration has on attacker behavior by simulating multiple networks and attacker behaviors.

The remainder of this paper is structured as follows: Section 2 contains related work to cyber attack behavior modeling and simulations. Section 3 gives a brief overview of the context models needed to simulate cyber attacks. Section 4 gives a detailed description of the ABM and its process flow. Section 5 describes the simulation setup and experiments that are conducted, where the results of the experiments are shown in Section 6. Lastly, conclusions and future work are discussed in Section 7.

2. Related work

The cyber analytics research field currently has two primary focus areas: attack detection and attack prevention. Attack detection methods range from common tools, such virus scanners or IDSs,[11] to more experimental research, including network behavior profiling and analytics.[12] Attack detection techniques are generally used as a defense method to mitigate damages, which requires observing malicious behaviors indicating potential attacks. Due to the importance and reliance of computer services and certain assets (e.g., customer data), prevention of cyber attacks is equally as critical as detection. This work focuses on pre-emptive analysis strategies. Our goal is to emulate the processes and data a real attack would produce for the purpose of analyzing a network's security from cyber attacks.

Intrusion prevention systems involve detecting vulnerabilities and exploits before an attack has occurred to inhibit or deter future cyber attackers. Core techniques, such as firewalls, access control, and vulnerability scanners,[13] are common methods for preventing basic cyber attacks. For enterprise networks with complex access control schemes and critical assets, more rigorous and comprehensive analysis techniques, such as penetration tests and white-hat live attack simulations, are used. These two practices have the common goal of synthesizing data and offer the following benefits: penetration tests provide detailed insight of the overall resistance to attacks[8] and live simulations demonstrate the behavior of attackers in a controlled environment that can be observed and analyzed.[6]

Overall security is demonstrated in attack graph research where single attack paths are generated using the relationship between the network configuration and the vulnerabilities on the network. Attack graphs allow for various degrees of detail, where the more attack features considered in the attack graph generally yields more realistic attack paths at the cost of computational complexity. Attack graph modeling ranges from considering only network connectivity and vulnerabilites, such as Jha et al.[14] and Sheyner et al.,[15] to dynamic risk assessments using asset and mitigation strategy models by Poolsappasit et al.[16] In 2015, Kotenko and Doynikova[17,18] extended this work by adding the CAPEC database to the attack graph structure to reflect realistic attack paths and scenarios to further increase the quality of the assessment.

To observe how and why an attacker performed certain actions (the main benefit of live simulations), two methods are used: game theory and, less commonly, simulation. Game theory is a useful technique for modeling the interaction between the red and blue teams. Wang et al.[19] develop a method that seeks an optimal defense strategy using game theory. In addition, Chung et al.[20] use Q-Learning to learn past behaviors and defense strategies to play the most effective defensive game. Game theory is a particularly strong application for cyber analytics because it describes detailed behaviors of both the red and blue teams, which enables analysis of the interactions between them.

Similar to game theory, the impacts of attacks and attackers can be realized through the use of configurable cyber attack simulation platforms. NeSSi2, an agent-based simulation platform by Grunewald et al.,[21] models a packet-level description of a network with the primary focus on simulating the effects of distributed denial of service (DDoS) attacks. NeSSi2 models the effects of various worm behaviors and how worms propagate through a network. This technique proves to be useful in other contexts, such as smart grid networks.[22] The predecessor to this work by Moskal et al.[23] introduced a cyber-context model-driven simulator called the Multistage Attack Scenario Simulator (MASS). The MASS develops a Virtual Terrain (VT) and an ABM to simulate the interactions between the network and attackers. Leveraging data from MITRE[24] and the National Vulnerability Database (NVD)[25] to describe the network landscape and the attacker's behavior in terms of configurable parameters, the MASS is able to simulate various cyber attack scenarios. The MASS also models dynamic defense strategies and IDSs to allow the capability of correlating the simulated ground truth data and other data that is typically observed as a result of real network attacks (e.g., IDS logs).

The attack graph literature provides a comprehensive analysis of the potential vulnerabilities and is expandable to account for many different scenarios at the cost of computational complexity as the model becomes more complex (see, for example, Poolsappasit et al.[16]). The game theoretic approaches provide detailed models of the interactions between the red and blue teams, resulting in realistic models of specific cyber attacks. The challenge is to comprehensively analyze the network's security while realistically representing a diverse set of cyber attack scenarios with a reasonable computational complexity. We propose models that capture the essence of what is needed to simulate the interaction between the red and blue teams by leveraging industry accepted models and building on existing research while maintaining an acceptable run time and realistic results.

3 Simulating red team versus blue team

Simulating the interactions between the red team (attackers) versus the blue team (targeted network and network defense) requires the modeling of human versus network interactions (exploitation) and network versus human interactions (defense).[20] The representation of a cyber attacker or a human requires modeling at many levels of detail. High-level cyber attack scenario descriptions can be used to describe an attacker[18] or specific behaviors can be modeled.[19] Networks can also be modeled in various levels of detail: from complex packet-level descriptions[21] to less detailed network terrain descriptions.[23] To simplify these abstract models of humans and networks, an ensemble of cyber-based context models are used to narrow down the potential search space to cyber-specific details. This work uses the concept of previous research.[23,26,27] to describe base context models that encompass the requirements of cyber attacks in general where each context model contains specific components to capture detailed functionality.

Illustrated in Figure 1 are the three basic building blocks to simulate network threats and attacks and how the models interact with one another. The two primary context models are the network and the attacker. The network model that represents the blue team is based on the VT of Moskal et al.[23] and Wheeler,[27] which describes network machines, services, vulnerabilities, and defense mechanisms at a high level. The VT avoids explicitly modeling some ancillary network features, such as packet

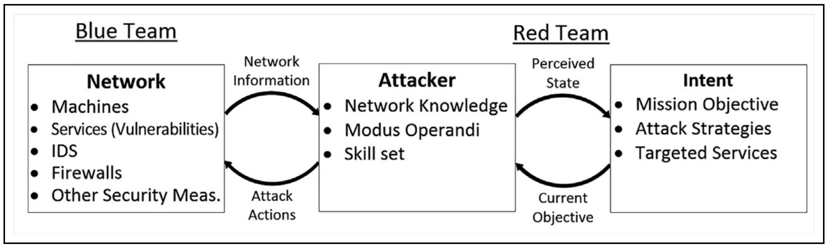


Figure 1. The three cyber-based context models and the knowledge flow between them. IDS: intrusion detection system.

routing and packet flow, while maintaining important cyber security-related features, including firewall configurations and system permissions. In the system architecture, the VT defines the network landscape and features the red team can target and exploit. Figure 1 depicts the dependence the red and blue teams have on one another, although teams are defined separately and function independently. We use the notion of information exchange between the component models to develop the basis of our methodology for representing the behaviors of the red team.

The primary focus of this work is modeling the behaviors and actions of the red team. The red team consists of two parts: the attacker(s) and the intent of the attack. The objective of the attacker model is to represent the processes by which cyber attackers use information when selecting cyber attack actions (see Moskal[28] for a comprehensive analysis of tools and information used by cyber attackers). Intent defines the underlying purpose of the attack, including the attack mission objective, attack strategies, and targeted services. This aspect is important, as the intent of the attack on the network[29] influences the attacker's decision process.[26] Moskal[28] describes the methodology behind the network knowledge, modus operandi, and and the definition of the intent, as described in Figure 1. The following section describes the methodology and process of selecting an attack using the cyber attacker context models given a network description and an intent.

4 Attacker behavior model

The ABM is designed to represent the attacker behaviors, their decision-making processes, the attack actions chosen to interact with the network, and intent models. The purpose of the ABM is to choose an attack action to apply to a network given the accumulated knowledge of the network at a given time in the overall attack scenario being described. The ABM chooses a single attack action to perform on a given network by reducing the set of all possible actions on the network using cyber-based context models leveraging real-world descriptions of cyber attack features. The singular attack action and initial action set in the ABM are defined as follows.

Definition 1. *The* ***network model*** *is defined by a set T of all nodes in the network and a set E of all exposures or vulnerabilities on the network, where a node $t \in T$ can access its children set C_t and $S = \{t|C_t \neq \emptyset, \forall t \in T\}$ is the set of network nodes that could serve as potential attack source nodes in the network.*

Definition 2. *An* ***attack action*** *is defined as a three-tuple $a = (s, t, e)$, where $s \in S$ is a source node (Internet Protocol (IP) address), $t \in T$ is a target node (IP address), and $e \in E$ is an exposure such that s takes action on t by exploiting exposure e expressed as $s \overset{e}{\to} t$. Let $s \mapsto t$ denote the mapping of source node s to target node t, and let $e \mapsto t$ denote the mapping of exposure e that could be exploited on target node t.*

Definition 3. *The* ***attack action set*** *is defined as $A = \{a|s \in S_t \text{ and } e \in E_t, \forall t \in T\}$, where for each $t \in T$, $S_t \subseteq S$ such that $S_t = \{s|t \in C_s\}$ and $E_t = \{e|e \in E\text{ande} \mapsto t\}$.*

The selection process of an attack action for a single iteration of the simulated attack scenario is comprised of three set reduction stages and a selection phase. The reduction stages correspond to the attacker's capability, opportunity, and intent (COI). The attack action set A is filtered using the network, intent, and the ABM context models. This approach builds on the COI model developed by Holsopple and Yang[26] and Gasper,[29] where the COI is used to describe and predict future cyber attacks. The attack action selection phase used in the ABM, represented as the attacker's preferences using the output from the COI model, determines the probabilities for selecting $a \in A$ that result in variable attack paths in the output of the simulation.

This filtering process is illustrated in Figure 2. The input data needed for the filtering process includes the current intent of the attacker and the attacker's knowledge at that given point in time. In particular, the input to the ABM consists of the following.

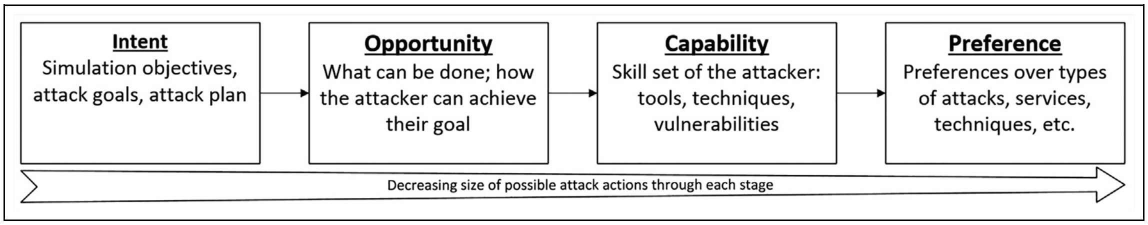

Intent
Simulation objectives, attack goals, attack plan
Opportunity
What can be done; how the attacker can achieve their goal
Capability
Skill set of the attacker: tools, techniques, vulnerabilities
Preference
Preferences over types of attacks, services, techniques, etc.
Decreasing size of possible attack actions through each stage

Figure 2. Internal attacker behavior model states to determine the final attack action from the attacker's intent, opportunity, capability, and preference.

- The attacker's intent, which is represented as a user-defined set of exposures E_{aim} describing the high-level goal of the current step of the attack. The intent model provides the intent to the ABM.
- The attacker knowledge, K, of the network, which is represented by a set of variables describing the VT, including known sources, targets, open ports, services, vulnerabilities, etc.

The attacker's knowledge is a complex data structure representing the various network attributes the attacker has uncovered throughout the attack. A detailed discussion of the attacker knowledge constructs appears in Moskal.[28] The filtering process results in an attack action, a. The sequence of cyber attack actions $\{a_{[1]}, a_{[2]}, \ldots, a_{[n]}\}$ taken by an attacker during an attack scenario represents an attack path.

4.1 Opportunities: Cyber Attack Kill Chain

The opportunities an attacker sees at a given time over an attack scenario depend on the his/her intent and accumulated knowledge of the network. The ABM utilizes the notion of the cyber kill chain, which describes the stages over which an attack scenario could transpire, to assess the opportunities, that is, the possible attack actions based on the attacker intent and accumulated knowledge.

As defined by the Cyber Attack Kill Chain® by Lockheed Martin in 2013,[30] each cyber attack scenario is comprised of a sequence of action types (e.g., reconnaissance, privilege escalation, etc.) The notion of a kill chain is integrated into the ABM as a set of reduction functions describing the types of attack actions that match the objective of a particular kill chain state. We found that depending on the type/context of an attack (e.g., DDoS, data extraction, etc.) or the organization describing the kill chain, the set of kill chain states may vary.[28] Through an analysis of the various kill chain definitions, a set of common kill chain stages have been created as the base set of stages to describe an attack. The minimum viable kill chain (MVKC) is defined as follows.

Definition 4. *The MVKC is defined as the minimum set of kill chain states to describe a realistic attack scenario. The MVKC consists of the kill chain states: –{RECONNAISSANCE, BREACH, EXFILTRATION} where each state describes a reduction function onto A.*

The selection process of the kill chain stages employs fuzzy logic to capture a balance between a rule-based behavior model and a probabilistic model. The parameters used by the fuzzy logic depend on the attacker's accumulated knowledge of the network and the outcomes of past actions. This approach allows the description of the behaviors of an attacker by controlling the membership functions of each kill chain stage as well as the definition of the kill chain stages outside of the MVKC. Figure 3 shows the general process of the kill chain selection phase, where the inputs are the intent and attacker knowledge and the output is a single kill chain stage.

To determine the membership for a particular kill chain stage, a set of ''attack stimuli'' is generated based on the input data. Table 1 shows the set of attack stimuli (used to define the linguistic variables used in the fuzzy rules) currently represented in the ABM. This set can be separated into three categories: (1) the cumulative machines and services discovered (a–d); (2) the newly discovered machines and services (e–h); and (3) the past successes and failures (i–l).

These stimuli influence the definition of fuzzy rules for each of the attack stages enabling representation of an array of attacker types. A membership function is defined for each of the kill chain stages to describe the process by which the attacker chooses attack types. Figure 4 shows an example of a fuzzy rule set for an ''Expert'' attacker type using the MVKC set. Note that the fuzzy rule set itself does not fully represent an attacker behavior. The rule set is combined with the capability and preferences to determine the behavior that influences the attack decision process. The modeling of capability and preferences will be discussed in the next few sections.

For the defuzzification step, linguistic variables (denoted by u) are then accumulated into each of the membership functions (μ) to create a set of membership values for each of the kill chain stages (u'). The kill chain stages are prioritized in order from least importance (*RECONNAISSANCE* for this work) on the ''left'' to most important (*EXFILTRATION*) on the ''right.'' The membership ($\mathcal{S}$) to one of the kill chain stages is determined by the right-most maximum value on the state definition (1):

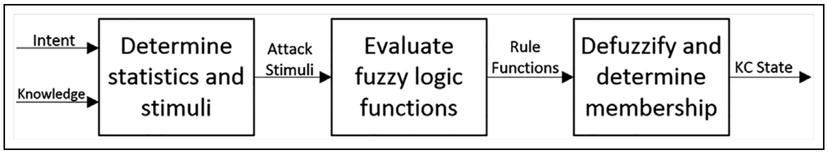

Intent
Knowledge
Determine statistics and stimuli
Attack Stimuli
Evaluate fuzzy logic functions
Rule Functions
Defuzzify and determine membership
KC State

Figure 3. The flow of the kill chain (KC) stage selection using fuzzy logic that processes the attacker's knowledge and intent.

Table 1. The set of the attack stimuli used in the fuzzy logic rules.

	Variables	Description	Values
a	Known machines	Number of machines known by the attacker	$(0, \infty]$
b	Service scanned	Number of machines scanned by the attacker	$(0, \infty]$
c	Scanned-pinged ratio	Service scanned machines/known machines	(0,1)
d	Machines with intent	Number of machines with intent services known on them	$(0, \infty]$
e	Newly discovered	Number of machines "recently" discovered	$(0, \infty]$
f	Newly scanned	Number of machines "recently" service scanned	$(0, \infty]$
g	Newly compromised	Number of machines "recently" compromised	$(0, \infty]$
h	Newly scanned-pinged ratio	Newly scanned machines/newly discovered machines	(0,1)
i	Success rate	Over all types of attacks, successful actions/all actions	(0,1)
j	Total successful actions	Over all types of attacks, the count of successful actions	$(0, \infty]$
k	Total failed actions	The total count of failed actions	$(0, \infty]$
l	No attack count	Count of where the attacker failed to choose an attack	$(0, \infty]$

IF Scanned-Pinged Ratio is *low* **OR** Newly Compromised Machines is *high*
  THEN Kill Chain Stage is *RECON*

IF Newly Scanned-Pinged Ratio is *high* **AND** Newly Compromised Machines is *low*
  THEN Kill Chain Stage is *BREACH*

IF Machines With Intent is *high*
  THEN Kill Chain Stage is *EXFILTRATION*

Figure 4. Example fuzzy rule set for each of the kill chain stages in the minimum viable kill chain.

$$\mathcal{S} = \max(u') = \max(\mu(u)) \tag{1}$$

The reduction function from the kill chain stage returned from the defuzzification process is then applied to A. The set A now represents the available a with the contribution of the attacker's intent and the opportunities (knowledge).

4.2 Capability: skill set of the attacker

Each attacker's capability is represented by a finite set of exposures, E_c, that an attacker is capable of performing on target nodes in the network. The attacker capability is used to reduce the set of exposures on each target node, E_t for $t \in T$, to the set of exploits the attacker is capable of performing on that target. The reduction function corresponding to the attacker's capability is as follows:

$$E_t \Leftarrow E_t \cap E_c \qquad \forall t \in T. \tag{2}$$

The set E_c may potentially be large and difficult to define given the number of potential exposures. To alleviate this burden, the implementation of the capability allows the user to explicitly define a set of exposures the attacker is not capable of performing; functioning as a blacklist of exposures for E_t. As a result of applying the attacker's capability, the reduced set E_t contains only the exposures the attacker can perform on the target and the attacker will not attempt exposures not applicable to the target or outside of his/her capability.

4.3 Preference: selecting an attack action

The attack action selection phase begins with the generation of the probabilities of each attack action contained in A based on the attacker's preference parameters. The preference parameters are specified for the target, source, and exposure within an attack action a. The parameters are defined as follows:

α_s distribution parameter for source preference, range: [0,1];
α_t distribution parameter for target preference, range: [0,1];
α_p exposure preference weight, range: [0,1], default value: 0.5.

In addition to the user-specified parameters, a set of internal variables are utilized to determine the probability of selecting each attack action. These variables are defined as follows:

τ time difference between the current time and the time at which the knowledge was gained;
w_s source preference weight determined by $f(\alpha_s, \tau)$, range: [0,1];
w_t source preference weight determined by $f(\alpha_s, \tau)$, range: [0,1];

w_e exposure preference weight, range: [0,1].

The probability of selecting a particular attack action $p(a)$ is defined as the joint probabilities of choosing a target $p(t)$, a source given the target $p(s|t)$, and an exposure given the target $p(e|t)$ by the following:

$$p(a) = p(t) * p(s|t) * p(e|t) \qquad \forall a \in A \tag{3}$$

Let T_A be the the set of potential targets of the attack actions currently in the reduced attack action set A. The independent probability $p(t)$ is defined as the normalized target selection weight given the target preference parameter α_t and τ and is calculated by the following:

$$p(t) = \frac{w_t}{\sum\limits_{t \in T_A} w_t} \qquad \forall t \in T_A \tag{4}$$

where

$$w_t = f_t(\alpha_t, \tau) = e^{-\alpha_t \tau}. \tag{5}$$

The parameter α_t controls the exponential decay function $f_t(\alpha_t, \tau)$ to determine the relative strength of the preference given to information that has been discovered more recently. Similarly, $p(s|t)$ is determined by the following:

$$p(s|t) = \frac{w_s}{\sum\limits_{s \in S_t} w_s} \qquad \forall s \in S_t \tag{6}$$

where

$$w_s = f_s(\alpha_s, \tau) = e^{-\alpha_s \tau}. \tag{7}$$

The exposure weights w_e depend on the user-specified constants α_p, which are mapped to each exposure such that the set E_p contains $\alpha_p \mapsto e, \forall e \in E$. Since the set E is likely to be large, a value of 0.5 is used as a default weight for the exposures that the user elects not to specify. Thus, a user-specified weight greater than 0.5 means that the corresponding exposure will be more likely to be selected, and less likely to be selected if it is less than 0.5. The probabilities of selecting an exposure given the set E_t for a given target t is as follows:

$$p(e|t) = \frac{w_e}{\sum\limits_{e \in E_t} w_e} \qquad \forall e \in E_t \tag{8}$$

where

$$w_e = \begin{cases} \alpha_p & \text{if user-specified} \\ 0.5 & \text{otherwise} \end{cases} \tag{9}$$

To complete the selection phase and an iteration of the ABM logic, a single attack action $a \in A$ is selected at random based on the set of $p(a)$ generated. This attack action is returned to the simulator where the impacts are applied to the network and the ABM's knowledge is updated depending on the result of the attack action. This ABM process is repeated until one of three conditions is met:

1. the intent module reports the intent has been satisfied;
2. the ABM reports the attacker has chosen to stop the attack based on some user-defined condition; or
3. the simulation core logic times out the attack (fail safe).

In summary, each iteration of the ABM chooses an attack action by reducing the set of all possible actions through the use of cyber-based contextual models and attacker knowledge representation to model the attacker's actions toward achieving a goal. The intent, opportunity, and capability context models described in the ABM aid in the reduction of the possible actions the attacker could perform given their knowledge of the targets at a given time in the attack. The preferences contribute to the attacker action choice by weighing in the attacker's preference of targets, sources, and exposures. These four models combined along with the knowledge representation allows the ABM to generate complex multi-step attack scenarios given a network configuration and provides a rationale on why an action is chosen at a given time.

5 Simulation test configuration

To demonstrate the impacts that specific types of attackers have on the security of a network and how the configuration of the network affects the progress of the attacker, simulations are run for multiple network configurations and multiple attacker types. In this section, simulated threat assessments are performed on two separate networks by analyzing four different attacker behaviors. To conduct these experiments, the Cyber Attack Scenario and Network Defense Simulator (CASCADES) is used to model and simulate a variety of cyber attack scenarios. The network, the attacker, and the intent context models shown in Figure 1 are represented in CASCADES as the VT, ABM, and Adversary Intent Module (AIM), respectively. Figure 5 gives an architectural representation of the CASCADES system, which is written in Java and allows thousands of concurrent simulations of the same or different configurations of the context models.

One thousand simulations were run for each combination of the network configuration and attacker type,

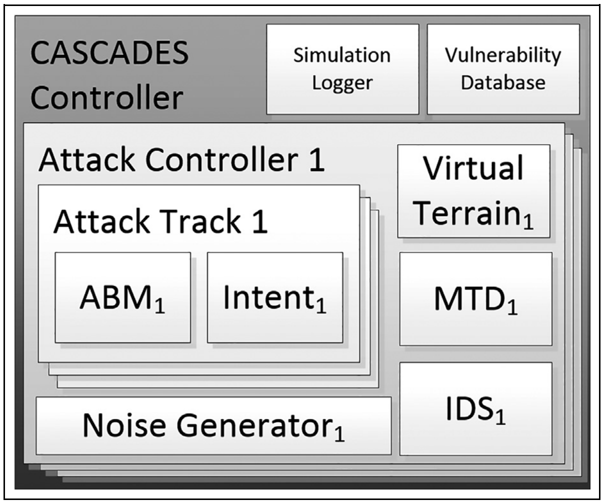


Figure 5. The overall Cyber Attack Scenario and Network Defense Simulator (CASCADES) system architecture. ABM: attacker behavior model; IDS: intrusion detection system; MTD: Moving Target Defense.

generating a total of 8000 attack paths. These attack paths can be analyzed individually or, for the most part of this paper, aggregated to generate metrics, such as the average time-to-intent, success rate, vulnerability usage frequency, and target machine frequency. For reference, all simulations were conducted on a 2013 Macbook Pro 2.4GHz Intel i5 processor with 8 GB of RAM, where a batch of 1000 concurrent simulations takes roughly 400 seconds to complete for the network to be discussed next.

The network is crafted to reflect key attributes that are found in typical enterprise networks, including multiple tiers of permissions, typical enterprise services, and firewall configurations. In total, the network contains four distinct subnets and 15 unique machines. Figure 6 depicts the example network and Table 2 gives the machine types and the access permissions allowed in the network. The permissions column describes the allowed source location a particular machine is allowed to communicate with, as described by the VT's firewall configuration. An ''X'' in the permissions column signifies allowed communication between two machines.

To demonstrate the hypothesis that the network configuration plays a large role in the network's security against an attack,[2] a second, ''misconfigured'' network is created. In this second network, the firewall is misconfigured to give external Internet access to a level 3 machine (ID: 1003). It is hypothesized that this misconfiguration could have significant impacts in the overall success of the attackers and the security of the network due to the machine being in a protected area of the network.

This work addresses four different types of attackers—the expert, the amateur, the comprehensive, and the random attackers—for the purposes of demonstrating the impact of the different attacker behaviors on a network. The expert and the amateur are designed to measure the differences as a result of the capabilities of attackers, while the comprehensive and the random attackers are to measure the network's security against different brute force attacks. Collectively, simulating these attacker types enables a comprehensive threat assessment of the network subject to attacker behavior, which is difficult to do in real life. The attackers differ in three possible ways: capability, kill chain stage membership rule definition, and preferences. Tables 3 and 4 show the different configurations of the capability and membership functions between each of the attackers and the attacker's preference configuration,

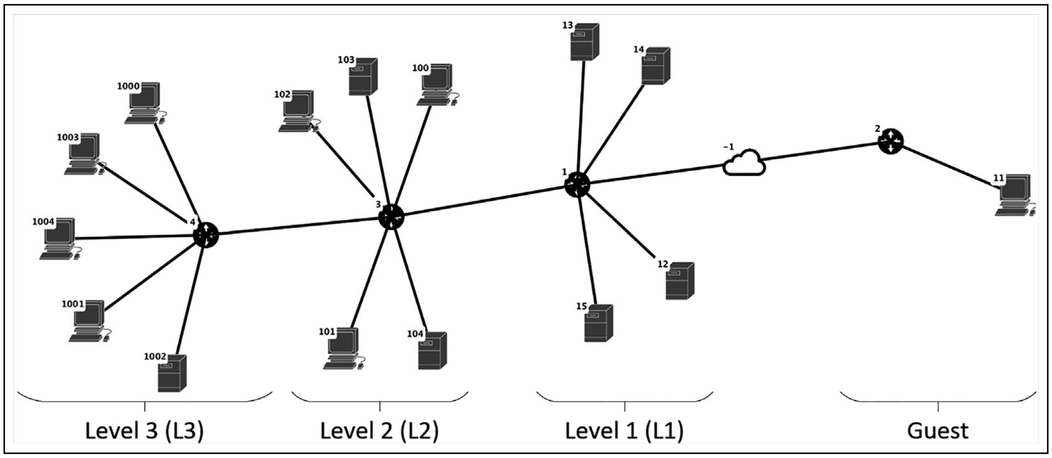


Figure 6. Visual representation showing the connections and machines in the virtual terrain being used.

Table 2. List of the machines on the virtual terrain and their access permissions.

Machine Info			Permissions			
ID	IP Address	Name	I	L1	L2	L3
1	192.168.1.255	External Router	X			
2	192.168.2.255	Guest Wireless Router	X			
3	192.168.100.255	Level 2 Router		X	X	X
4	192.168.200.255	Level 3 Router			X	X
11	192.168.2.1	Guest Windows 8	X			
12	192.168.1.1	External Web Server	X	X		
13	192.168.1.2	External DNS Server	X	X		
14	192.168.1.3	External Mail Server	X	X		
15	192.168.1.4	External Web App Server	X	X		
100	192.168.100.1	Developer Machine (Unix)		X	X	X
101	192.168.100.2	Developer Machine (Mac)		X	X	X
102	192.168.100.3	Administrator (Windows)	X	X	X	X
103	192.168.100.4	OES Print Server		X	X	
104	192.168.100.5	OpenLDAP Server		X	X	X
1000	192.168.200.1	VMWare ESXi			X	X
1001	192.168.200.2	NAS			X	X
1002	192.168.200.3	Backup Server				X
1003	192.168.200.4	Development GitLab			X	X
1004	192.168.200.5	MySQL Customer Info			X	X

IP: Internet Protocol.

Table 3. Attacker behavior model configuration summaries for each attacker.

	Amateur	Expert	Comprehensive	Random
Capability	10%	100%	100%	100%
Recon stage	Comprehensively scan	Selective	Comprehensively scan	Random
Breach stage	Scanned any machine	Selective	Found machines	Random
Exfil. Stage	Intent	Intent	90% Machines compromised	Random
Stop stage	5 Failures	5 Failures	Exhausted known options	5 Failures

Table 4. Attacker behavior model preference summary for each of the attackers.

	Amateur	Expert	Comprehensive	Random
α_s	.5	.9	.1	.1
α_t	.1	.9	.9	.1
α_p	1	1	.5	.5
E_p	Comprehensive recon	Pre-existing knowledge	None	None

respectively. The intent is static across all simulations where the goal is to target the database containing the customer information (ID: 1004).

For each set of simulations, the difference between the attacker's behaviors and the network security are analyzed using the following metrics:

- average, minimum, and maximum number of actions needed to achieve intent;
- percentage of failures;
- kill chain stage selection frequency; and
- frequency of exploited services.

6 Simulation results

The purpose of the first set of experiments conducted in this work is to understand the differences between the attacker behaviors using the ''baseline'' network shown in

Table 5. An example simulated attack sequence for an expert attacker on the baseline network.

Timestamp	Step	Source IP	Dest. IP	Attack type	Service	Attack action
01/11-19:12:17	1	129.21.194.1	192.168.1.1	RECON	nmap -p 1-65535 -T4 -A -v	THOROUGH
01/11-19:12:17	1	129.21.194.1	192.168.1.2	RECON	nmap -p 1-65535 -T4 -A -v	THOROUGH
01/11-19:12:17	1	129.21.194.1	192.168.1.3	RECON	nmap -p 1-65535 -T4 -A -v	THOROUGH
01/11-19:12:17	1	129.21.194.1	192.168.1.4	RECON	nmap -p 1-65535 -T4 -A -v	THOROUGH
01/11-19:12:17	1	129.21.194.1	192.168.2.1	RECON	nmap -p 1-65535 -T4 -A -v	THOROUGH
01/11-20:23:29	2	129.21.194.1	192.168.1.4	COMP.	Ruby on Rails 0.9.4.1	CVE-2013-0156
01/11-23:52:30	3	192.168.1.4	192.168.100.1	RECON	nmap -sn	PING_SCAN
01/11-23:52:30	3	192.168.1.4	192.168.100.2	RECON	nmap -sn	PING_SCAN
01/11-23:52:30	3	192.168.1.4	192.168.100.4	RECON	nmap -sn	PING_SCAN
01/11-23:52:30	3	192.168.1.4	192.168.100.5	RECON	nmap -sn	PING_SCAN
01/12-02:08:06	4	192.168.1.4	192.168.100.2	RECON	nmap -T4 -A -v -Pn	CONNECT
01/12-07:47:19	5	192.168.1.4	192.168.100.5	COMP.	OpenBSD OpenSSH 1.2	CVE-2015-8325
01/12-11:11:45	6	192.168.100.5	192.168.200.1	RECON	nmap -sn	PING_SCAN
01/12-11:11:45	6	192.168.100.5	192.168.200.2	RECON	nmap -sn	PING_SCAN
01/12-11:11:45	6	192.168.100.5	192.168.200.4	RECON	nmap -sn	PING_SCAN
01/12-11:11:45	6	192.168.100.5	192.168.200.5	RECON	nmap -sn	PING_SCAN
01/12-14:55:09	7	192.168.100.5	192.168.200.5	RECON	nmap -T4 -A -v -Pn	CONNECT
01/12-18:10:52	8	192.168.100.5	192.168.200.5	COMP.	MySQL MySQL 6.0.1	CVE-2008-2079

IP: Internet Protocol.

Figure 6 and Table 2. From there, we test the same attacker behaviors on the aforementioned "misconfigured" network. Comparing the simulation results on the baseline network versus those obtained on the misconfigured network, we can assess the effect of a single misconfiguration on the security of the network for each attacker behavior type.

While this paper focuses on the comparative analysis based on a large number of simulated scenarios, a single simulated attack sequence gives insights into how the ABM interacts with the network context model to simulate how an attacker may progress toward achieving the intended goal. Table 5 shows one simulated attack sequence by an expert attacker on the baseline network.

This attack sequence demonstrates how the attacker develops his/her knowledge by first performing recon on the machines accessible from the Internet and then executing a compromising action (Attack Type: "COMP."), targeting the web app server running Ruby on Rails exploiting CVE-2013-0156. The attacker then utilizes a stepping stone 192.168.1.4 and repeats a similar process as before. This time, the attacker tried a different type of scanning followed by exploiting the vulnerability CVE-2015-8325 for OpenSSH. From the new stepping stone, 192.168.100.5, the attacker discovered the mySQL service, which is the intent of the attack sequence, and took advantage of CVE-2006-2079 to compromise it. Note that the above attack sequence simulates an expert attacker who knows a broad range of exploits and happens to choose the right stepping stones to quickly get to the intended mySQL server. In other simulated scenarios, the attacker may not know some of the exploits (e.g., the amateur attackers) and fail to reach the intent, or may not select the right stepping stones (e.g., the comprehensive attackers) and take longer to complete. By simulating a large number of scenarios for each attacker types, this work reveals statistics that can be indicative of the effect of different attacker behaviors on the same network.

6.1 Base network configuration results

Figure 7 gives a box plot of the attack steps from the 1000 simulations of each of the four attacker behaviors on the baseline network. Table 6 shows the average number of attack actions for each of the attacker behavior types, and their failure rates. It is observed that the "amateur" and "expert" attacker types have a similar overall average number of attack steps, which, at first glance, is not expected since these two attacker types have significantly different capabilities—the exposures they know to exploit. A closer examination reveals that the amateur attacker scenarios have a higher failure rate than that achieved by the expert attacker scenarios. In other words, out of the scenarios that successfully reach the intended goal, the amateurs and the experts need similar number of steps statistically. This is not unreasonable, because the baseline network is not very large and complex. When simulating a sufficiently large number of scenarios, the amateur attackers could find alternative paths to reach the goal with a statistically similar number of steps as compared to the paths the expert attackers took. If the network is complex and has a significantly shorter path only the experts can

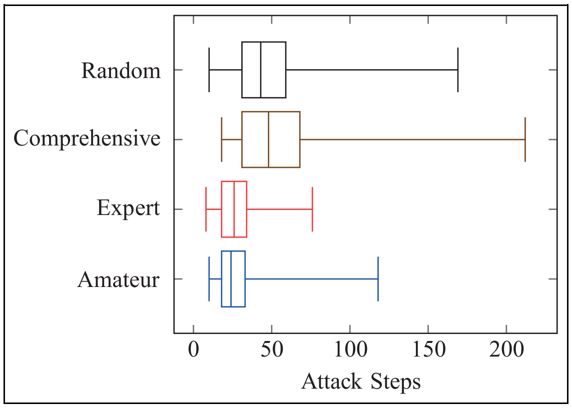


Figure 7. Comparison of the number of attack steps each attacker over 1000 simulation runs for the base network.

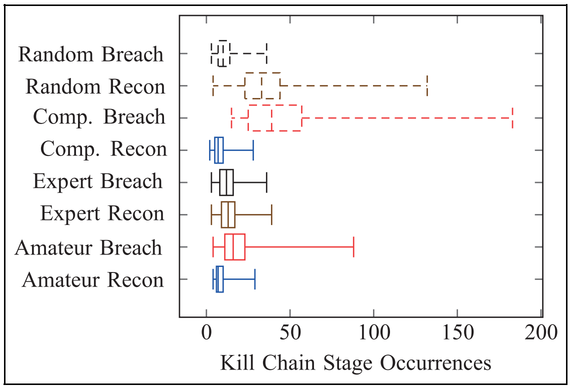


Figure 8. Comparison of the occurrences of the recon and breach stages for each attacker type on the baseline network.

Table 6. Average number of steps taken and failure rate for each attacker.

	Amateur	Expert	Comprehensive	Random
Avg. steps	27.5	27.6	54.9	47.4
% Failures	17.8	1.8	0	0

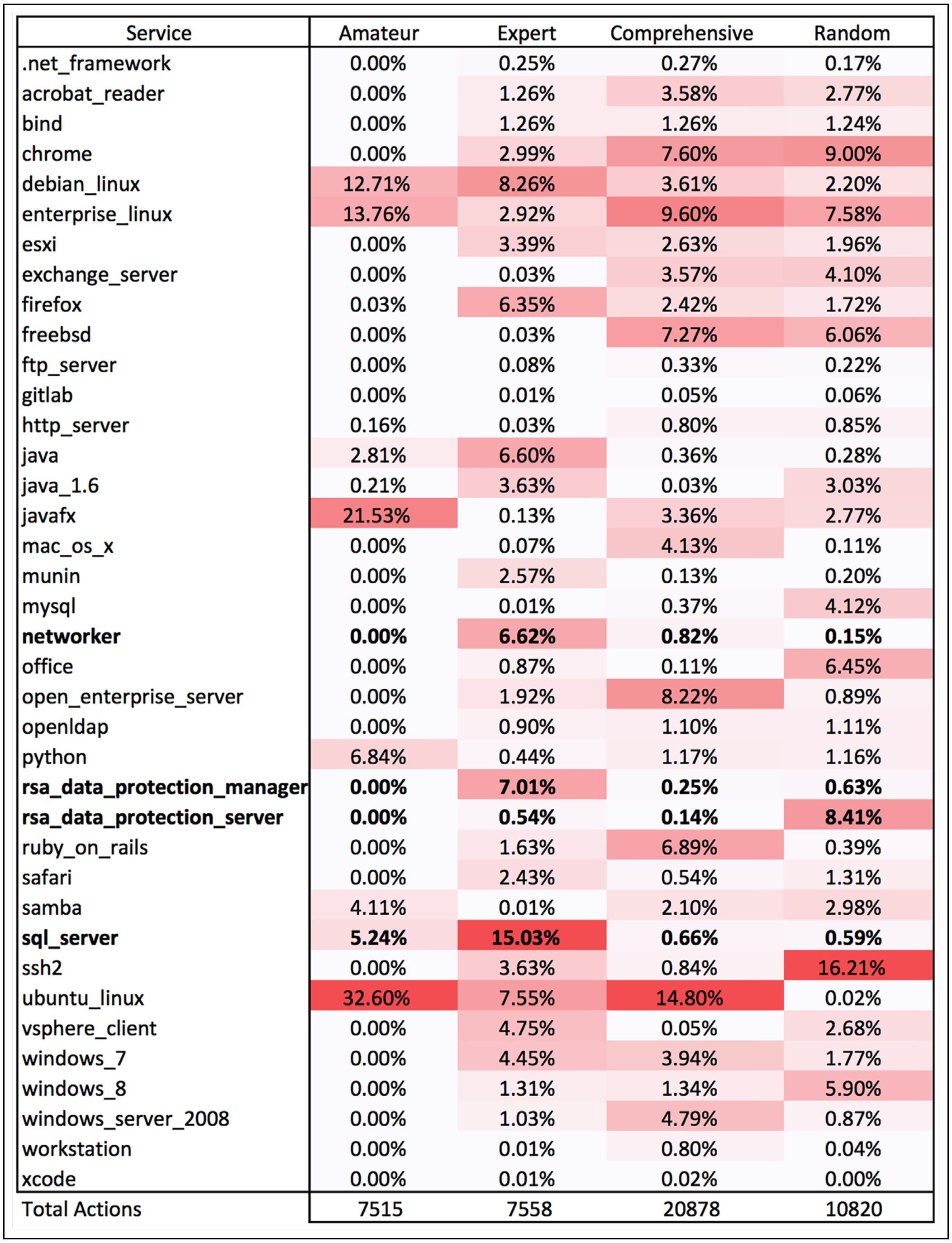
exploit, the Novice scenarios will show a higher number of steps.

Comparing the amateur and expert attackers to the comprehensive and random attackers, the overall number of steps is higher for the latter two and the failure rate is higher for the former two. The comprehensive and random attackers were set to never give up until achieving their intent, while the amateurs and experts had stop conditions. The fact that the comprehensive and random scenarios have a zero failure rate shows that there exists at least one path to reach the intent from the Internet. Furthermore, the number of steps under these two cases reveals what it takes if an attacker takes a brute force approach, even if it may never happen in real life.

Figure 8 gives the box plot of the usage of the recon versus the breach stages in the MVKC for each of the attacker types on the baseline network. This plot shows how each attacker type favors one stage versus the other. The experts use approximately half of their actions for recon and the other half for breach. Note that the experts often choose a recon action that is specific to the target they are interested in, and this is why the simulations show an approximately one-to-one ratio between the recon and breach stages for the expert attacker type. On the contrary, amateurs often comprehensively scan the network and, thus, discover services using fewer recon actions as compared to the experts. This is why the amateurs exhibits less recon usage than the experts. This, however, does not mean that the amateur is better, because the comprehensive network scans are usually easily detectable by IDSs. The random attackers exhibit much more recon behavior because of the mis-match between the random nature of choosing the target, source, and malicious actions and the non-random Fuzzy rule set defined for the breach stage. The comprehensive attackers exhibit significantly more breach actions because they are configured to compromise as many machines as possible.

Figure 9 shows the percentage of times each service is exploited by each attacker type on the baseline network. The bolded services are those associated with the intent/goal machine in our experiments. This representation gives insight into where the commonly exploited vulnerabilities are in the network. For example, the machine with an ‘‘ubuntu_linux’’ service that was commonly exploited by most of the attackers was an externally facing machine (ID: 12) and can be discovered easier than the other machines. Figure 9 also shows how different attacker types favor or are capable of exploiting specific services. For example, the expert attackers were configured to have an exposure preference (E_p) for SQL-based exploits, and the ‘‘sql_server’’ was indeed the most exploited by the experts, even when compared to the rest of the intent services combined. The amateur attackers have zeros for many of the services, because they are limited with the exploits they know how to use.

The comprehensive and the random attacker types give an interesting view into the most vulnerable services on the network. The random attackers have no preference on the source, target, or exposures, meaning that the probability of choosing any given attack action is uniformly distributed. It was found that the service and machine with the most overall exposures would be selected most. This is illustrated in the ‘‘ssh2’’ and ‘‘rsa_data_protection_server’’ cases, where the protection server had 23 total



Service	Amateur	Expert	Comprehensive	Random
.net_framework	0.00%	0.25%	0.27%	0.17%
acrobat_reader	0.00%	1.26%	3.58%	2.77%
bind	0.00%	1.26%	1.26%	1.24%
chrome	0.00%	2.99%	7.60%	9.00%
debian_linux	12.71%	8.26%	3.61%	2.20%
enterprise_linux	13.76%	2.92%	9.60%	7.58%
esxi	0.00%	3.39%	2.63%	1.96%
exchange_server	0.00%	0.03%	3.57%	4.10%
firefox	0.03%	6.35%	2.42%	1.72%
freebsd	0.00%	0.03%	7.27%	6.06%
ftp_server	0.00%	0.08%	0.33%	0.22%
gitlab	0.00%	0.01%	0.05%	0.06%
http_server	0.16%	0.03%	0.80%	0.85%
java	2.81%	6.60%	0.36%	0.28%
java_1.6	0.21%	3.63%	0.03%	3.03%
javafx	21.53%	0.13%	3.36%	2.77%
mac_os_x	0.00%	0.07%	4.13%	0.11%
munin	0.00%	2.57%	0.13%	0.20%
mysql	0.00%	0.01%	0.37%	4.12%
networker	**0.00%**	**6.62%**	**0.82%**	**0.15%**
office	0.00%	0.87%	0.11%	6.45%
open_enterprise_server	0.00%	1.92%	8.22%	0.89%
openldap	0.00%	0.90%	1.10%	1.11%
python	6.84%	0.44%	1.17%	1.16%
rsa_data_protection_manager	**0.00%**	**7.01%**	**0.25%**	**0.63%**
rsa_data_protection_server	**0.00%**	**0.54%**	**0.14%**	**8.41%**
ruby_on_rails	0.00%	1.63%	6.89%	0.39%
safari	0.00%	2.43%	0.54%	1.31%
samba	4.11%	0.01%	2.10%	2.98%
sql_server	**5.24%**	**15.03%**	**0.66%**	**0.59%**
ssh2	0.00%	3.63%	0.84%	16.21%
ubuntu_linux	32.60%	7.55%	14.80%	0.02%
vsphere_client	0.00%	4.75%	0.05%	2.68%
windows_7	0.00%	4.45%	3.94%	1.77%
windows_8	0.00%	1.31%	1.34%	5.90%
windows_server_2008	0.00%	1.03%	4.79%	0.87%
workstation	0.00%	0.01%	0.80%	0.04%
xcode	0.00%	0.01%	0.02%	0.00%
Total Actions	7515	7558	20878	10820

Figure 9. The percentage of times a service was exploited by each attacker type on the baseline network. (Services pertaining to intent are bolded.).

vulnerabilities as opposed to the other intent services, which have five vulnerabilities in total.

Simulating the different attacker types on the baseline network provides many insights into how attacker opportunities, capabilities, and preferences may lead to ways to achieve the same intent. The statistical differences based on the simulated scenarios offer a new perspective on cyber threat assessment, where the interplay between attacker behavior and network configuration signifies the potential threats under different malicious actor conditions. In the next section, we will discuss what happens when a network configuration is slightly changed.

6.2 Misconfigured network configuration results

The same experiments for the baseline network are conducted for the misconfigured network. Recall that the misconfiguration is that one of the machines close to the intent machine is accessible from the Internet. The attack step box plot, the kill chain stage frequency box plot, and the tabulated average number of steps for each attacker type are shown in Figures 10 and 11 and Table 7, respectively.

When introducing the misconfiguration that allows direct access from the Internet to a critical machine (ID: 1003), all but the amateur attacker use fewer steps (greater than 26% change) to achieve the intent. Note that the expert, the comprehensive, and the random attacker types are all configured to have the capability to exploit the vulnerabilities associated with the critical machine. It is expected that the expert attackers will benefit from this misconfiguration. Note that the comprehensive and the random attackers, despite using the brute force approach, exhibit an average number of steps closer to the amateur attackers. This is concerning, because these two attacker types do not intelligently make attack decisions based on the accumulated knowledge of the network. In other words, with a single misconfiguration, the brute force approaches can quickly get to the intent machine as long as the exploits are known.

Despite the significant changes in attack steps between the baseline and misconfigured networks, the general trends in the usage of recon versus breach stages for each of the attacker types are nearly identical, when comparing Figure 8 with Figure 11. This observation signifies that the general behaviors of the attackers are identical regardless of the network configuration. This allows the attackers to be simulated on any network configuration without fine tuning the behavioral models for each network.

Figure 12 and Figure 9 are compared next, which show the percentage of times each service is exploited by the different attacker types. The most significant difference between the two figures is the changes in the percentage that the ‘‘enterprise_linux’’ service is exploited, as shown in Table 8. Note that the misconfigured machine contains this service. The experts demonstrate a 384% increase in the percentage of exploits for this service, which is clear evidence showing that the experts are able to take advantage of this misconfiguration and, consequently, need fewer steps to achieve the intent. There is little change (–0.235%) for amateur attackers because of the insufficient capability. Meanwhile, the comprehensive and the random attackers can exploit this misconfiguration, but not as effectively as the expert attackers since they use brute force approaches.

Another observation from Figure 12 is the high frequency of attacks for the ‘‘ubuntu_linux’’ service for all attacker types. This may be useful for network security analysts as it gives insight into a critical set of vulnerabilities that should be patched. Additional analysis on the specific attack sequences could potentially reveal why many attackers exploited this particular service.

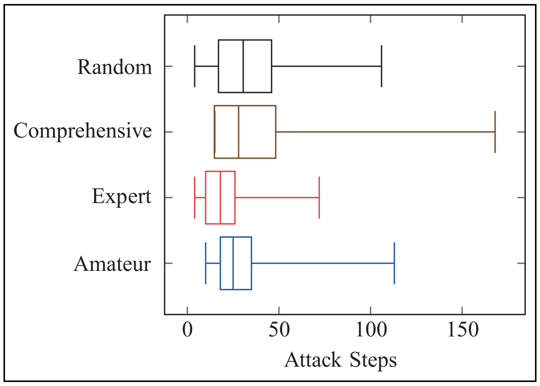


Figure 10. Comparison of the number of attack steps each attacker type took over 1000 simulation runs on the misconfigured network.

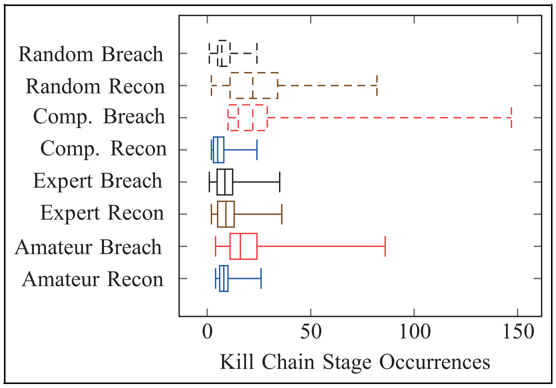


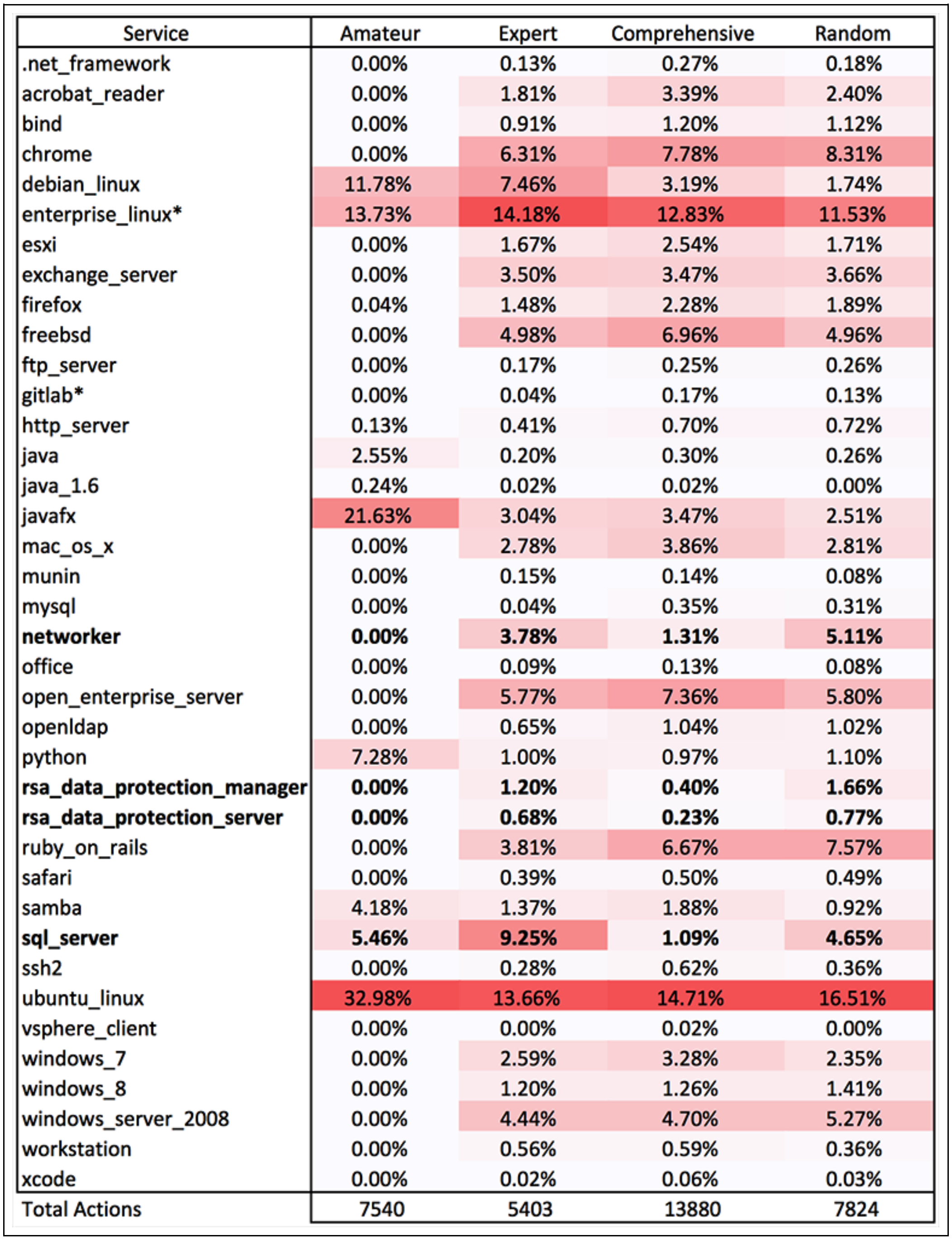
Figure 11. Comparison of the occurrences of the recon and breach stages for each attacker type on the misconfigured network.

Table 7. Average number of steps taken by each of the attackers on the misconfigured network with the percent change compared to the baseline network.

	Amateur	Expert	Comprehensive	Random
Avg. steps	28.8	20.4	36.5	33.5
% Change	4.35	–26.3	–33.4	–29.1

Table 8. Average percent change of attacks against the service “enterprise_linux” between the two networks.

	Amateur	Expert	Comprehensive	Random
Avg. % change	–0.235	384	33.7	52.1



Service	Amateur	Expert	Comprehensive	Random
.net_framework	0.00%	0.13%	0.27%	0.18%
acrobat_reader	0.00%	1.81%	3.39%	2.40%
bind	0.00%	0.91%	1.20%	1.12%
chrome	0.00%	6.31%	7.78%	8.31%
debian_linux	11.78%	7.46%	3.19%	1.74%
enterprise_linux*	13.73%	14.18%	12.83%	11.53%
esxi	0.00%	1.67%	2.54%	1.71%
exchange_server	0.00%	3.50%	3.47%	3.66%
firefox	0.04%	1.48%	2.28%	1.89%
freebsd	0.00%	4.98%	6.96%	4.96%
ftp_server	0.00%	0.17%	0.25%	0.26%
gitlab*	0.00%	0.04%	0.17%	0.13%
http_server	0.13%	0.41%	0.70%	0.72%
java	2.55%	0.20%	0.30%	0.26%
java_1.6	0.24%	0.02%	0.02%	0.00%
javafx	21.63%	3.04%	3.47%	2.51%
mac_os_x	0.00%	2.78%	3.86%	2.81%
munin	0.00%	0.15%	0.14%	0.08%
mysql	0.00%	0.04%	0.35%	0.31%
networker	**0.00%**	**3.78%**	**1.31%**	**5.11%**
office	0.00%	0.09%	0.13%	0.08%
open_enterprise_server	0.00%	5.77%	7.36%	5.80%
openldap	0.00%	0.65%	1.04%	1.02%
python	7.28%	1.00%	0.97%	1.10%
rsa_data_protection_manager	**0.00%**	**1.20%**	**0.40%**	**1.66%**
rsa_data_protection_server	**0.00%**	**0.68%**	**0.23%**	**0.77%**
ruby_on_rails	0.00%	3.81%	6.67%	7.57%
safari	0.00%	0.39%	0.50%	0.49%
samba	4.18%	1.37%	1.88%	0.92%
sql_server	**5.46%**	**9.25%**	**1.09%**	**4.65%**
ssh2	0.00%	0.28%	0.62%	0.36%
ubuntu_linux	32.98%	13.66%	14.71%	16.51%
vsphere_client	0.00%	0.00%	0.02%	0.00%
windows_7	0.00%	2.59%	3.28%	2.35%
windows_8	0.00%	1.20%	1.26%	1.41%
windows_server_2008	0.00%	4.44%	4.70%	5.27%
workstation	0.00%	0.56%	0.59%	0.36%
xcode	0.00%	0.02%	0.06%	0.03%
Total Actions	7540	5403	13880	7824

Figure 12. The percentage of times a service was exploited by each of the attacker types on the misconfigured network. (Services pertaining to intent are bolded.) (Services on the misconfigured machine are labeled with *.).

6.3 ABM discussion

The approach to modeling adversarial behavior of cyber attacks is different from typical data-driven cyber threat assessment methods, where the results are compared against known scenarios or vulnerable paths in the network. Our method provides both values and expected limitations when comparing to previous works. On the one hand, simulating attack scenarios based on the interplay between adversary behaviors and network configurations, reveals cases that may not be readily available in real-world data since not all networks are attacked by all types of adversaries—not to mention that the real-world cyber attack data is typically unavailable for analysis. This work provides a framework to not only model the four cyber attacker types shown but also behaviors observed in real cyber attacks or theoretical behaviors that may have not been observed before.

The results shown in this paper provide statistical summaries that no prior work offers to the best of our knowledge. On the other hand, because of the lack of comprehensive scenarios for all cases in the real world, one would not be able to verify the exact realism of the simulated scenarios against true occurrences. Instead, the purpose of the experimental study shown in this paper is to demonstrate assessment with rational explanation of simulated scenarios and insights obtained through statistical summaries. Two potential use cases for this model have been realized: pre-emptive vulnerability and security assessments for commercial uses and the generation of synthetic cyber attack data for research purposes.

This model allows for security practitioners to assess the impact of various behaviors, vulnerabilities, and network configurations. In the case of the misconfigured network example, the actual impact of making such a change may have not been apparent and may have not been classified as a ''misconfiguration'' as described in this work. Through simulations of different attacker behaviors, this vulnerability was exposed and its impacts were realized by the increase in the rate at which the attacker types were successful compared to the configuration without the vulnerability. In the absence of diverse and labeled cyber attack data, this simulated model generates synthetic data that can be used to fuel next generation prediction techniques. Future work involves extracting salient behavioral features from limited real-world data to generate ABMs. This allows extrapolating from a single example of a cyber attack to a number of simulated scenarios for further analysis.

The simulator also contains built-in logic to check against the validity of attack actions based on system vulnerabilities and exploits chosen. One may question the likelihood of one scenario occurring versus the other, particularly in terms of the exact adversary behavior. This study does not intend to suggest which adversary behavior may be more likely to occur than the other. The configuration parameters are based on the consultation of security practitioners and rational choices; they are not intended to represent any specific attackers. Research is underway to study the human/criminal behavior and translate such into computational models that can be formally evaluated and assessed.[31] The current study focuses on providing the capability where each ABM configuration can lead to randomly sampled/simulated scenarios, and thus offers insights that can be rationally explained and analyzed.

7 Summary and conclusion

The increasing diversity of attack strategies and cyber defense requires pre-emptive and comprehensive threat assessment. This paper presents a simulation framework that exhibits the interplay between the red and blue teams by defining three context models: the network, the attacker, and the intent and their interdependencies. Specifically, this work develops the ABM, which uses a process to examine the attacker's intent, opportunity, capability, and preference (COI + P) to simulate the selection of malicious actions iteratively. The proposed ABM, in conjunction with CASCADES, enables efficient and effective assessment of how different attack scenarios may transpire under different attacker COI + P against network configurations.

Through simulating different attacker behaviors on networks with different configurations, we show the insights one can obtain by comparing the scenarios transpired due to the interplay between the attackers and the network and system vulnerabilities. This paper models, particularly, the behaviors of four attacker types against two configurations of a network. Our results show the differences in behaviors between the attacker types to achieve the same goal on a network, and how they may alter their approaches when the network has a misconfigured firewall setting. Our tests comparing attacker behaviors showed that the ''amateur'' and the ''expert'' attacker types completed the simulation objective on average in a similar amount of time, however, with significantly different strategies. The ''amateurs'' were shown to use simple exploits and potentially detectible actions, whereas the ''experts'' took a methodical approach to the objective and only performed the minimum actions needed. When a backdoor was put into the network, the experts saw a 26% decrease in the time it took to achieve the simulation objective, and the attackers that were randomly picking actions began to be on average just as successful as the amateur attackers. With comprehensive simulation of scenarios for each configuration, we not only analyze specific scenarios but also evaluate summary statistics that are not available using other existing methods.

This work builds a foundation for a variety of possible risk, threat, and impact assessments. While the specific ABM configurations and rules implemented can continue to be improved, the demonstrated system capabilities allow the analysis of a variety of cyber attack scenarios by changing the attacker types and network configurations. This work provides a method to generate labeled cyber attack data leveraging both real-world vulnerability data and simulation of attacker behaviors to potentially strengthen current and future cyber attack analysis techniques.

Funding

This paper is based on research efforts supported by the National Security Agency (Grant #H98230-15-1-0277) and the National Science Foundation (Award #1526383).

References

1. WaterISAC. 10 basic cybersecurity measures best practices to reduce exploitable weaknesses and attacks, https://ics-cert.us-cert.gov/sites/default/files/documents/10_Basic_Cybersecurity_Measures-WaterISAC_June2015_S508C.pdf (2015, accessed 8 March 2017).
2. Ginter A. 13 ways through a firewall: what you don't know can hurt you, https://www.isa.org/standards-publications/isa-publications/intech-magazine/2013/april/special-section-13-ways-through-firewall-what-you-dont-know-can-hurt-you/ (2013, accessed 27 February 2017).
3. DiPietro J. To err is human; to automate, divine, https://www.infosecurity-magazine.com/opinions/to-err-is-human-to-automate-divine/ (2016, accessed 8 March 2017).
4. Golnabi K, Min RK, Khan L, et al. Analysis of firewall policy rules using data mining techniques. In: *proceedings of the 10th IEEE/IFIP network operations and management symposium*, Vancouver, BC, 3–7 April 2006, pp.305–315. Piscataway, NJ: IEEE.
5. Liu AX. Firewall policy change-impact analysis. *ACM Trans Internet Technol* 2012; 11: 15.
6. Mejia R. Red team versus blue team: how to run an effective simulation, http://www.csoonline.com/article/2122440/disaster-recovery/emergency-preparedness-red-team-versus-blue-team-how-to-run-an-effective-simulationhtml (2016, accessed 8 March 2017).
7. Day G. Measuring success in cybersecurity, https://www.fireeye.com/blog/executive-perspective/2015/03/measuring_success.html (2015, accessed 8 March 2017).
8. Group PTGSI. Penetration testing guidance, https://www.pcisecuritystandards.org/documents/Penetration_Testing_Guidance_March_2015.pdf (2015, accessed 8 March 2017).
9. Northcutt S. Penetration testing: assessing your overall security before attackers do, https://www.sans.org/reading-room/whitepapers/analyst/penetration-testing-assessing-security-attackers-34635 (2006, accessed 8 March 2017).
10. SS8. Types of cyber attackers and their motivations, http://blog.ss8.com/types-of-cyber-attackers-and-their-motivations/ (2015, accessed 8 March 2017).
11. Raiyn J. A survey of cyber attack detection strategies. *Int J Secur Appl* 2014; 8: 247–256.
12. Xu K, Zhang ZL and Bhattacharyya S. Profiling internet backbone traffic: behavior models and applications. *ACM SIGCOMM Comput Commun Rev* 2005; 35: 169–180.
13. Thacker BH, Riha DS, Fitch SH, et al. Probabilistic engineering analysis using the NESSUS software. *Struct Saf* 2006; 28: 83–107.
14. Jha S, Sheyner O and Wing J. Two formal analyses of attack graphs. In: *proceedings of 2002 15th IEEE computer security foundations workshop*, Cape Breton, Nova Scotia, 24–26 June 2002, pp.49–63. Piscataway, NJ: IEEE.
15. Sheyner O, Haines J, Jha S, et al. Automated generation and analysis of attack graphs. In: *proceedings of 2002 IEEE symposium on security and privacy*, Oakland, California, 12–15 May 2002, pp.273–284. Piscataway, NJ: IEEE.
16. Poolsappasit N, Dewri R and Ray I. Dynamic security risk management using Bayesian attack graphs. *IEEE Trans Depend Secure Comput* 2012; 9: 61–74.
17. Kotenko I and Doynikova E. Security assessment of computer networks based on attack graphs and security events. In: *proceedings of ICT-EurAsia*, Bali, Indonesia, 14–17 April 2014, pp.462–471. Berlin: Springer.
18. Kotenko I and Doynikova E. The CAPEC based generator of attack scenarios for network security evaluation. In: *proceedings of 2015 IEEE 8th international conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)*, Warsaw, Poland, 24–26 September 2015, vol. 1, pp.436–441. Piscataway: IEEE.
19. Wang B, Cai J, Zhang S, et al. A network security assessment model based on attackdefense game theory. In: *proceedings of the IEEE 2010 international conference on computer application and system modeling (ICCASM)*, Taiyuan, China, 22–24 October 2010, pp.V3–639. Piscataway, NJ: IEEE.
20. Chung K, Kamhoua CA, Kwiat KA, et al. Game theory with learning for cyber security monitoring. In: *proceedings of 2016 IEEE 17th international symposium on high assurance systems engineering (HASE)*, Orlando, Florida, 7–9 January 2016, pp.1–8. Piscataway, NJ: IEEE.
21. Grunewald D, Lützenberger M, Chinnow J, et al. Agent-based network security simulation. In: *proceedings of the 10th international conference on autonomous agents and multiagent systems*. Taipei, Taiwan, 2–6 May 2011, pp.1325–1326. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
22. Chinnow J, Tonn J, Bsufka K, et al. A tool set for the evaluation of security and reliability in smart grids. In: *proceedings of international workshop on smart grid security*, Berlin, Germany, 3 December 2012, pp.45–57. Berlin: Springer.
23. Moskal S, Wheeler B, Kreider D, et al. Context model fusion for multistage network attack simulation. In: *proceedings of IEEE military communications conference (MILCOM '14)*, Baltimore, MD, 6–8 October 2014, pp.158–163. Piscataway, NJ: IEEE.
24. MITRE. Common vulnerabilities and exposures, https://cve.mitre.org (2017, accessed 14 June 2017).
25. National Institute of Standards Technology. Computer security resource center: national vulnerability database, https://nvd.nist.gov (2017, accessed 14 June 2017).
26. Holsopple J and Yang SJ. FuSIA: future situation and impact awareness. In: *proceedings of the 11th ISIF/IEEE international conference on information fusion*, Cologne, Germany, 30 June–3 July 2008, pp.1–8. Piscataway, NJ: IEEE.
27. Wheeler BF. *A computer network model for the evaluation of moving target network defense mechanisms*. Rochester, NY: Rochester Institute of Technology, 2014.
28. Moskal S. *Knowledge-based decision making for simulating cyber attack behaviors*. Rochester, NY: Rochester Institute of Technology, 2016.

29. Gasper PD. Cyber threat to critical infrastructure. In: *proceedings of information & cyberspace symposium*, Fort Leavenworth, Kansas, 22–24 September 2008, pp.22–24. Idaho Falls, ID: Idaho National Laboratories.

30. Cyber Kill Chain. Lockheed Martin Security, http://cyber.lockheedmartin.com/solutions/cyber-kill-chain (2016, accessed 11 April 2016).

31. Rege A, Singer B, Masceri N, et al. Measuring cyber intrusion chains, adaptive adversarial behavior, and group dynamics. In: *proceedings of 5th international conference on management leadership and governance*, Dayton, Ohio, 2–3 March 2017, p.285. Reading: Academic Conferences and Publishing Limited.

Author biographies

Stephen Moskal is currently a PhD of Engineering student at Rochester Institute of Technology (RIT). He has received his BS and MS degrees in Computer Engineering at RIT in 2016. His current research focuses on the simulation and modeling of cyber attack scenarios and behaviors along with the application of deep machine learning techniques to cyber security.

Dr S Jay Yang received his BS degree in Electronics Engineering from National Chiao-Tung University, Hsin-Chu, Taiwan, in 1995, and his MS and PhD degrees in Electrical and Computer Engineering from the University of Texas at Austin in 1998 and 2001, respectively. He is currently a Professor and the Department Head for the Department of Computer Engineering at RIT. He is a Co-Director of the Networking and Information Processing (NetIP) Laboratory and a core member for the Center of Cybersecurity at RIT. His current research focuses on developing machine learning, data analytics, and simulation techniques to enable proactive cyber defense. He has also contributed to the development of two PhD programs at RIT, and received the Norman A. Miles Award for Academic Excellence in Teaching in 2007.

Michael E Kuhl is currently a Professor in the Department of Industrial and Systems Engineering at RIT in Rochester, NY. He holds a PhD in industrial engineering from North Carolina State University. His research focuses on simulation modeling and analysis with applications to cyber security, healthcare, supply chain, and project management.