A Decision Engine for Configuration of Proactive Defenses—Challenges and Concepts

Michael Atighetchi, Brett Benyo
Raytheon BBN Technologies
Cambridge, MA
{matighet, bbenyo}@bbn.com

Thomas C Eskridge
Florida Institute of Technology
Melbourne, FL
teskridge@fit.edu

David Last
Air Force Research Laboratory
Rome, NY
david.last.1@us.af.mil

***Abstract* — Selecting appropriate cyber defense mechanisms for an enterprise network and correctly configuring them is a challenging problem. Identifying the set of defenses and their configurations in a way that maximizes security without exhausting system resources or causing unintended interference (a situation known as cyber friendly-fire) is a multi-criteria decision problem, which is difficult for humans to solve effectively and efficiently. Proactive defenses are especially difficult to configure due to their temporal nature. This paper describes the challenges and solution concepts for a decision engine that (1) intelligently searches for optimal cyber defense configurations in a way that leads to continuously improving solutions; (2) uses compute clusters to scale computation to realistic enterprise-level networks; and (3) presents meaningful choices to operators and incorporates their feedback to improve the suggested solutions.**

Keywords: cyber security analysis, modeling, threat assessment

I. Introduction

In current cyber warfare, the odds are inherently stacked against the defender. According to the 2015 Verizon Data Breach Investigation Report [1], attackers were able to compromise an organization within minutes in 60% of cases and many of these attack can go undetected for months. Cyber attackers frequently automate much of their work through management platforms, such as Metasploit, that enable rapid sharing and reuse of code. Furthermore, malware has evolved to the point where botnets and viruses make autonomous decisions, e.g., to remain dormant if they detect monitoring in an environment or to intertwine attacks with regular user activities to stay within the variance of observable parameters. This level of sophistication and the time pressure introduced by automated execution makes targeted attacks difficult to detect and mitigate.

One way the system owners and cyber defenders have responded to counter this threat is to use proactive defenses to make targets less predictable, giving rise to what is known as Moving Target Defenses (MTDs) [2]. State-of-the-art MTDs continuously change attack surfaces of applications, hosts, and networks to increase adversarial work load and uncertainty. While there is great value in proactive defenses in general and in MTDs specifically, it is also quite easy to add defenses that provide little added value, introduce unacceptable cost or overhead, inadvertently increase the attack surface, or exhibit unintended side effects when combined with other defenses. A Command and Control of Proactive Defense (C2PD) solution is needed to prevent such cyber friendly fire. We envision a decision engine as an integral component of C2PD to help cyber defenders choose from among available proactive defenses, configure deployed defenses, and achieve the best protection for the target system with the least impact on the system's mission effectiveness.

Distribution Statement "A" (Approved for Public Release, Distribution Unlimited). Case #88ABW-2016-1829. This effort is sponsored by the Air Force Research Laboratory (AFRL).

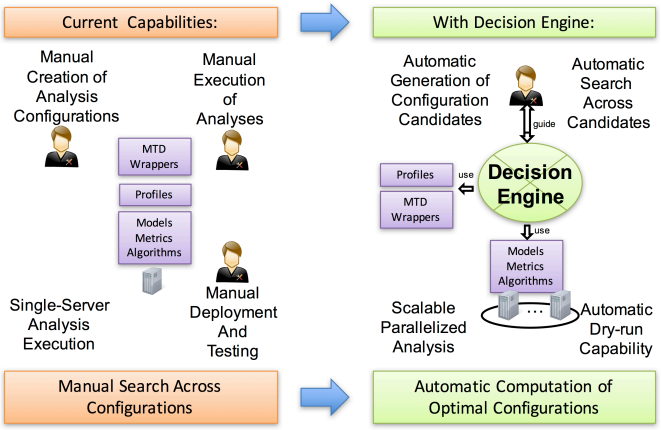

Fig. 1. High-Level Approach of Attack Surface Reasoning

As shown in Fig. 1, such a decision engine will enable defenders to select and configure the most appropriate cyber defenses for a given target environment supporting multiple concurrent mission operations more effectively and efficiently. By automating activities at multiple levels, the decision engine transforms a cyber defense management process that is currently dominated by manual operations into a streamlined computer-assisted workflow, which delegates heavyweight computation to a compute cluster and leverages human insight to guide the search for optimal configurations.

Using the decision engine, cyber defenders will be able to explore a large space of possible configuration settings in a short amount of time, enabling an agile defense posture that continuously incorporates and adapts defenses based on new proactive

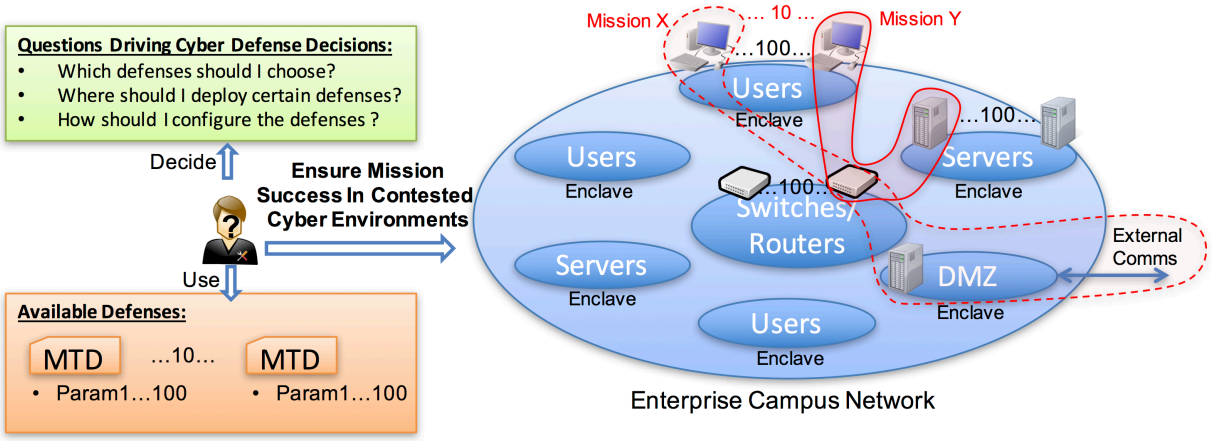

Fig. 2. Protection of mission operations in the context of an enterprise campus network

defenses that become available, new information about adversarial capabilities, new mission or changing requirements, and/or changes in protected systems. The benefits of the decision engine will apply to a variety of cyber defenders in different environments, including system administrators and other personnel who are responsible for the continued operation of computer systems and networks that might come under attack.

The rest of the paper is organized as follows. Section II provides a motivating example, Section III describes the high-level architecture of the decision engine, Section IV discusses key technical challenges and solutions, Section V reviews related work and Section VI concludes the paper.

II. Motivating Example

To illustrate the decision problems that cyber defenders face, consider a simplified example of a network environment typically found in enterprise environments, such as the networks present at a company campus shown on the right of Fig. 2. A campus network consists of multiple network enclaves each containing hundreds of computing resources, including servers, laptops, and network equipment. Resources in the enclave can be shared between multiple concurrent missions with different requirements on security guarantees (e.g., expressed in terms of availability, confidentiality, and integrity of services and data) and cost (e.g., measured by throughput and latency of information exchanges).

Cyber defenders, shown on the left, are responsible for ensuring mission success by ensuring that the compute and network resources provide the required functionality, even in contested cyber environments involving sustained adversarial activities. To protect the target network and systems, they have access to a number of defenses, including different MTD implementations, each with a different set of parameters and associated security benefit/cost tradeoffs. The main decision problem faced by cyber defenders is composed of the following three parts:

Defense Selection (Which defenses should I choose?): The defender needs to choose the most appropriate combination of defenses for the given set of resources, missions, and expected attack types. One key concern for deployment is the desire to create defense-in-depth postures using a strategic combination of defenses that complement each other and require adversaries to overcome multiple hurdles. However, a common problem is concentration on a single type and instance of defense, which can have wide-ranging consequences if adversaries find a bypass or compromise vulnerability for the specific defense.

Defense Deployment (Where should I deploy certain defenses?): The defender needs to identify the places in the network or platforms to place the selected defense instances. Liberally sprinkling defenses throughout the network without regard for their resource requirements and interactions can easily become a management nightmare and introduce unacceptable cost, causing missions to fail. One driving concern is to ensure defense coverage over the attack surface. For instance, a frequent mistake is to concentrate defenses on the network layer and fail to provide defense coverage on endpoints.

Defense Parameterization (What parameters should I choose for the defenses?): Once the defense and the deployment targets are identified, the cyber defender needs to ensure that the defenses are configured properly for the systems on which they have been installed. Modern cyber defenses can offer a large set of tunable parameters to adjust. A driving concern is to find parameter settings that maximize security while controlling for associated costs. A secondary concern is the desire to create configuration diversity across defense instances to increase adversarial workload.

Given these decision points, cyber defenders in charge of deploying and monitoring defenses face a multi-criteria decision problem well beyond the scale at which a single person can be expected to find optimal solutions by hand. This is particularly true because the criteria involved are not independent of each other, requiring search across a large space of possible candidate configurations. Manual approaches generally turn into frustrating tasks of continually tweaking candidate configurations, may devolve to random walk searches, and are not the best use of human time and expertise. In addition, it is also hard for humans to notice when a radically different candidate configuration change is warranted, e.g., due to a few changes in mission requirements. Finally, human-only approaches suffer from a knowledge transfer problem, as new cyber defenders require significant training and knowledge transfer associated with staff rotation.

The decision engine automates the tedious manual activities associated with exploration of defense performance while at the

same time leveraging human intuition and experience to help guide the search for optimal defense selection, deployment, and parameterization. The combination of these data along with the semantic representation of network, system, attack, and defense models form a candidate configuration to be evaluated by the decision engine. The decision engine provides a feasible solution as fast as possible, using further time and resources to refine the candidate configuration or explore a wider set of options by finding alternate candidate configurations that either favor different cost/security tradeoffs or lead to structurally different deployments.

III. Decision Engine Architecture

To achieve scalability and minimize decision latencies, the design of the decision engine strategically combines anytime search with big-data processing. Fig. 3 shows the overall architecture of the decision engine as a collection of three frameworks.

The ***GA framework***, shown in the middle of the figure, implements the anytime search across defense configurations. The Candidate Generator constructs new defense configurations to consider using multiple methods, including (1) a knowledge base of previous operator-selected standard configurations, (2) genetic crossover and random mutations of high-scoring candidates from the previous iterations of the search algorithm, and (3) mixed-initiative guidance provided by human operators. The GA framework uses the reasoning framework discussed below to compute fitness scores over security and cost tradeoffs. Upon receiving results, the Selectors choose a subset of the higher-scoring configurations as input for the next round of candidate generation.

The ***parallelized reasoning framework***, shown at the bottom of Fig. 3, computes the security and cost tradeoffs of the attack surface associated with each candidate configuration using algorithms developed under the previously developed Attack Surface Reasoning (ASR) effort. Multiple candidate configurations can be computed independently of each other, allowing for effective parallelization using cloud-computing substrates.

The ***UI framework***, shown at the top of Fig.3, enables operators to provide feedback on the direction of evolution used in the GA search, allowing human input to better guide the search. Human operators can influence the search tradeoff between *exploration*, where the candidate generator can produce largely varying configurations to explore different areas of the search space, and *exploitation*, where smaller changes are made to a promising high-scoring candidate, to more thoroughly explore a small region of the configuration space. For ultimate control, operators can request specific changes, e.g., the use of a specific defense or a restriction on modifying a network resource, to be included in the next generation. In addition, operators can access quantitative results about the currently explored defense configurations, e.g., to identify the configuration with the highest security given a certain upper limit for cost. At any time, the operator can access the best configurations found so far and determine whether the search is explorative (better results may take many generations to be found, if at all) or exploitative (better results can be found in a few more iterations).

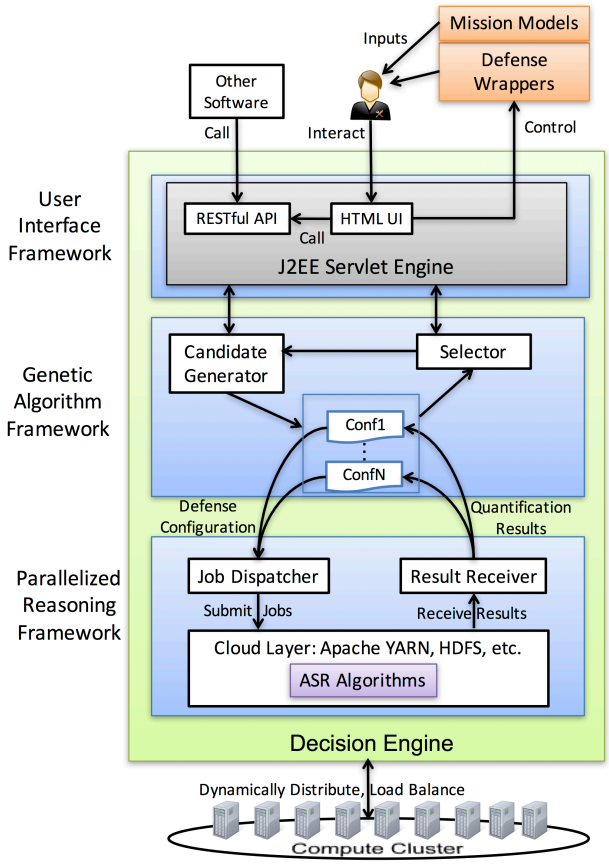

Fig. 3. The decision engine features a modular design that enables integration of the GA search with cloud frameworks a collaborative User Interface.

IV. Solutions to Key Technical Challenges

A. Optimize multi-dimensional utility functions

The decision engine needs to take into account constraints from IT infrastructures, adversary capabilities, and mission operations to identify the best security possible at an acceptable cost. Solving utility functions for more than one constraint is very difficult for humans to manage.

Our approach for solving the defender's multi-criteria decision problem involves *anytime search* [3] over the set of possible candidate configurations (i.e., what, where, and how to deploy defenses) in a practical way that hides much of the complexity from the defender and presents results in easy to understand and quantitative way. Leveraging capabilities developed under ASR, the decision engine reuses previous work on quantifying the attack surface of a candidate configuration through a *fitness function* F over the three aggregate level indexes provided by ASR [4][5][6], namely the Aggregate Security Index (ASI), Aggregate Cost Index (ACI), and Aggregate Mission Index (AMI). In addition, the decision engine provides additional metrics for inclusion into the fitness function, namely a Defense Conflict Index (DCI), User Preference Conformity (UPC), and Solution Uniqueness Value (SUV). Note that metric values are turned into ratios for metrics where lower means better. These metrics not only cover criteria related to security, cost, and mission-impact, but also capture the risk of functional incompatibilities between multiple defenses (DCI) and enable operators to provide guidance in the search for optimal configurations (through the

UPC and SUV). Furthermore, the decision engine enables operators to define multiple *selectors*, each with a specific set of weights used for the calculation of a fitness function. The overall search process can use a combination of selectors, e.g., to focus on finding the most secure configurations initially (through a selector with a proportional high weighting factor for the ASI) while switching over the search to minimizing cost later on (through a selector with a proportional high weighting factor for the ACI).

B. Engage the operator via a targeted what-if capability

As new defenses become available and situations change in environments where defenses are already deployed, it is desirable to do quick reevaluations. Furthermore, drastic changes to already deployed components are untenable in operational environments during live mission execution. For these reasons, the decision engine needs to support targeted exploration through a what-if capability.

The decision engine provides a directed model-based UI that enables operators to inject their knowledge and constraints into the search and decision-making process, e.g., by removing a specific defense instance or all instances of a specific type, making specific changes to defense parameters, or changing the importance of features in the evaluation function. This enables cyber defenders to start with a candidate configuration and study the impact of specific changes prior to deployment.

C. Identify unintended interaction effects across defenses

Deploying multiple cyber defenses into a network can easily lead to cyber friendly fire. A decision engine needs to deconflict multiple defenses by reasoning about unintended side effects and competing requirements on security and cost. The decision engine extends prior work on ASR to identify interaction effects introduced by software dependencies and information that is required to be static by some defenses but dynamically varied by others, and introduce the new Defense Conflict Index to quantify these effects.

D. Operate at realistic scale, tempo, and fidelity

To assist cyber defenders in operational environments, a decision engine needs to analyze candidate configurations within hours across base-level networks (hundreds of hosts) covering relevant and available cyber defenses (both proactive/reactive and across hosts/networks) in support of multiple concurrent missions. The models used by the decision engine also need to accurately reflect real-world attacks and defense aspects in order to avoid making decisions using information that is stale, incomplete, or inappropriate.

The decision engine addresses these challenges via three design considerations. First, the decision engine leverages cloud technologies to scale to large problems by using an appropriate level of compute resources. Second, the anytime properties of GA search enable the decision engine to quickly arrive at a good-enough answer that users can work with immediately, while the system continues to look for a globally optimal candidate configuration. Third, the decision engine leverages human intuition and experience through a UI that guides search convergence to higher quality solutions faster.

V. Related Work

The concept of a decision engine that assists cyber defenders in finding the best placement and configuration of defenses relates to a number of different efforts. Some formal methods approaches, e.g., [7][8], employ I/O automata to formalize attack surfaces and provides a metric for measuring such surfaces. Other formal approaches such as [9] use discrete event simulation to explore the attack space. These approaches differ from our work in that they (1) are based on less user-accessible formalisms than the ASR semantics-based approach and (2) require high-fidelity functional models of systems. Other attack surface measures and frameworks such as [10] focus on a single application and operate at lower levels concentrating on source code or software modules. The work described in [11] focuses on a specific class of web applications rather than entire distributed systems. Numerous manual threat modeling and analysis frameworks [12][13] provide common terminology, diagrammatic notation, and process descriptions for threat modeling. However, these frameworks do not perform any automated analysis nor necessarily provide specific metrics and measures. Beyond attack surface measurement, quantifying security in general is a long-running and difficult line of research. Both [14] and [15] outline some of the complexities and undertake a survey of potential directions forward.

VI. Conclusion and Next Steps

This paper describes emerging work on a decision engine that assists cyber defenders in selecting and configuring defenses to maximize security and minimize cost. The main contributions of this paper are a discussion of the critical needs, together with a description of high-level design and innovative solutions to the key technical challenges. In future work, we plan to create prototype implementations of the decision engine and evaluate performance in the context of multiple realistic use cases.

References

[1] Verizon, "2015 Data Breach Investigations Report."

[2] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving target defense: creating asymmetric uncertainty for cyber threats*, vol. 54. Springer Science & Business Media, 2011.

[3] S. Zilberstein, "Using anytime algorithms in intelligent systems," *AI Mag.*, vol. 17, no. 3, p. 73, 1996.

[4] M. Atighetchi, B. Simidchieva, N. Soule, F. Yaman, J. Loyall, D. Last, D. Myers, and C. B. Flatley, "Automatic Quantification and Minimization of Attack Surfaces," presented at the The 27th Annual IEEE Software Technology Conference (STC 2015), Long Beach, CA, 2015.

[5] N. Soule, B. Simidchieva, F. Yaman, R. Watro, J. Loyall, M. Atighetchi, M. Carvalho, D. Last, D. Myers, and C. B. Flatley, "Quantifying & Minimizing Attack Surfaces Containing Moving Target Defenses," presented at the 3rd International Symposium on Resilient Cyber Systems (ISRCS), Philadelphia, PA, 2015.

[6] M. Atighetchi, N. Soule, R. Watro, and J. Loyall, "The Concept of Attack Surface Reasoning," in *The Third International Conference on Intelligent Systems and Applications*, Sevilla, Spain, 2014.

[7] P. K. Manadhata and J. M. Wing, "A Formal Model for a System's Attack Surface," in *Moving Target Defense*, Springer, 2011, pp. 1–28.

[8] P. Manadhata and J. Wing, "An Attack Surface Metric," *IEEE Trans. Softw. Eng.*, vol. 37, no. 3, pp. 371–386, 2011.

[9] M. D. Ford, K. Keefe, E. LeMay, W. H. Sanders, and C. Muehrcke, "Implementing the ADVISE security modeling formalism in Moebius," in *Dependable Systems and Networks (DSN), 2013 43rd Annual IEEE/IFIP International Conference on*, 2013, pp. 1–8.

[10] M. Howard, "Attack surface: Mitigate security risks by minimizing the code you expose to untrusted users," *MSDN Mag.*, Nov. 2004.

[11] T. Heumann, S. Turpe, and J. Keller, “Quantifying the Attack Surface of a Web Application,” *Sicherheit*, pp. 305–316, 2010.
[12] A. Shostack, *Threat Modeling: Designing for Security*. John Wiley & Sons, 2014.
[13] P. Saitta, B. Larcom, and M. Eddington, *Trike v. 1 methodology document*. 2005.
[14] M. Torgerson, “Security metrics for communication systems,” presented at the 12th International Command and Control Research Technology Symposium, Newport, Rhode Island, 2007.
[15] W. Jansen, *Directions in Security Metrics Research*. Diane Publishing, 2010.