This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2022-3697J

Defender Policy Evaluation and Resource Allocation With MITRE ATT&CK Evaluations Data

Alexander V. Outkin[‡*], Patricia V. Schulz[§†], Timothy Schulz[¶†], Thomas D. Tarman[‖*], Ali Pinar[***]
[*]Sandia National Laboratories [†]Research performed while at Sandia National Laboratories
[‡]avoutki@sandia.gov, [§]tricia.schulz@outlook.com, [¶]tim@scythe.io, [‖]tdtarma@sandia.gov,
[**]apinar@sandia.gov,

Abstract—Protecting against multi-step attacks of uncertain start times and duration forces the defenders into indefinite, always ongoing, resource-intensive response. To allocate resources effectively, the defender must analyze and respond to an uncertain stream of potentially undetected multiple multi-step attacks and take measures of attack and response intensity over time into account. Such response requires estimation of overall attack success metrics and evaluating effect of defender strategies and actions associated with specific attack steps on overall attack metrics. We present a novel game-theoretic approach GPLADD to attack metrics estimation and demonstrate it on attack data derived from MITRE´ s ATT&CK® Framework and other sources. In GPLADD, the time to complete attack steps is explicit; the attack dynamics emerges from attack graph and attacker-defender capabilities and strategies and therefore reflects "physics" of attacks. The time the attacker takes to complete an attack step is drawn from a probability distribution determined by attacker and defender strategies and capabilities. This makes time a physical constraint on attack success parameters and enables comparing different defender resource allocation strategies across different attacks. We solve for attack success metrics by approximating attacker-defender games as discrete-time Markov chains and show evaluation of return on detection investments associated with different attack steps. We apply GPLADD to MITRE´ s APT3 data from ATT&CK® Framework and show that there are substantial and un-intuitive differences in estimated real-world vendor performance against a simplified APT3 attack. We focus on metrics that reflect attack difficulty vs. attacker ability to remain hidden in the system after gaining control. This enables practical defender optimization and resource allocation against multi-step attacks.

Index Terms—security, cyber security, MITRE ATT&CK, game theory, Markov processes, Markov chains, attacker, defender, attack graphs, attack, physics of attacks, stochastic process, probability theory, optimization, optimal policy, PLADD, GPLADD

1 Introduction

Cyber defenders must monitor and respond to attacks continuously, and their resource allocation decisions must optimize against an uncertain stream of attacks. With no limits on possible start and end times of attacks, the attacker-defender interaction becomes a game of indefinite duration were the attack success probability or other metrics associated with a pre-defined time interval are not sufficient to measure the defender strategy effectiveness. Defenders need a practical method for allocating the intensity of their effort over time, because attacks can occur at any time.

Attacks are also not localized in space. Different parts of an attack can take place on different parts of the defender system, in part because real world attacks generally involve multiple steps. The attacker needs to get into the defender system, move through it, gather additional information, communicate with a command-and-control center, and act on objectives. This multi-step nature of attacks increases the scope of the defender problem, but also presents multiple detection and response opportunities at different steps in the attack. However, without quantitative metrics that reflect the effects of the defender detection and other efforts corresponding to different steps of an attack, the defender has no reliable criteria for allocating resources across such attack detection or disruption opportunities. Should the defender spend time educating system users against phishing attempts, investigating lateral movement events, looking for command-and-control messages, etc.?

The defender problem of securing the system becomes the problem of resource allocation over time and across different parts of the system, where the defender objective is to allocate resources to detection and assessment and to place sensors across the defender system while keeping the expenditures below a certain threshold. The top-level defender problem is then to design a system that provides adequate response and warning about attacks, sufficient automated prevention or response against identified attacks, and uses the available resources in an optimized fashion.

We present a rigorous approach for solving this problem of allocating scarce defender resources to where they are needed most in security settings. It enables quantitative evaluation of the defender policy[1] against multi-step attacks. The primary practical goal of this paper is to help people use their resources as efficiently as possible (buying the right software, hiring the right people, etc.). A key part of our approach is the ability to evaluate the attack success metrics regarding different defender strategies and resource

1. Our approach relates both to dynamic game theory and Markov decision process. Those areas tend to use terms "strategy" and "policy" respectively to describe decision-making algorithms. We use terms "policy" and "strategy" interchangeably here.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

allocations. This method then allows evaluating what the defender can improve in his system. It demonstrates the opportunities and the need for the defender strategy optimization.

We focus on two kinds of metrics: 1) that quantify the attack difficulty and 2) that quantify the attacker ability to remain hidden in the system after attack succeeds. We show that those are somewhat orthogonal, because even if the attack is difficult in the sense of taking a long time to succeed, it may still provide an opportunity for the attacker to remain in the system after attack success.

We primarily use attack time-to-success to quantify attack difficulty in this paper, but other measures, such as attack success probability withing a particular time interval can be readily calculated as well.

While the attack will succeed given enough time, this is insufficient and not necessarily particularly concerning to the defender. The defender needs to know what happens after the attack success. Will the attacker stay in the defender system, or will the attacker be disrupted, and their success obliterated, given the security procedures the defender has in place? The defender needs to know how long it will take for the attack to succeed and what are the chances that the attack will succeed and stay in that "success" state without the defender noticing.

To quantify the attacker ability to remain hidden, we introduce a novel attack metrics - the fraction of time attacker spends in the "Ready" (completed) state. We call this metric "Ready-residence time". The "Ready" state is defined as the state when all attack steps are completed, but the attacker has not yet executed on the attack objectives. The "Ready" state residence time is important because it reflects the ability of the attacker to execute timed attacks or coordinate multiple attacks. Both can happen when the attacker can stay in the "Ready" state with low probability of disruption by defender.

In the Markov chain approach, we use the steady state distribution of the Markov chain to represent the fraction of the time an attacker spends at a particular step. Therefore the "Ready-residence-time" is the weight of the last (terminal) step of the attack in the steady state distribution of the Markov chain. If we assume that the attacker conducts campaigns composed from the individual attacks, then this strengthens this interpretation. The attacker runs campaigns either until he achieves the desired success conditions or abandons the attack. The timing of the attacks in the campaign is determined by the attacker strategy.

Different defenders may have different preferences across the attack difficulty and ready-residence time measures. We show how both kinds of metrics can be impacted by the defender.

We demonstrate this approach by showing how different defender resource allocation strategies or capabilities associated with different attack steps affect the defender performance against an attack. We specifically quantify the effects of a defender allocating more resources for detection at specific attack steps by calculating the sensitivity of the Ready-residence time to such changes.

We also demonstrate the differences in performance of different defenders[2] based on MITRE data.

In the attack metrics analysis, we take the detection probabilities associated with each attack step as parameters and inputs to our analysis. They can be inferred from literature, MITRE data, defender data, etc. They can be associated with the improvements with detection at individual attack steps or they can be a result of exploring correlation between attacker activities at different steps of the attack and therefore improving detection on multiple steps of attack.

Analysis of correlated detection is out of scope of this paper. However, this approach can be used directly and without modification to evaluate the returns on investment on such correlated or "bundled" detection improvements. Furthermore, our attack graph formalism allows visual and intuitive exploration of possibilities for correlated detection.

Many tools in the security industry, particularly Endpoint Detection and Response (EDR) and Security Information and Event Management (SIEM) tools, correlate events together to some degree and even establish a timeline. At present effectiveness of such correlation analysis is limited because the actions that adversaries use to achieve objectives may not be inherently malicious and can also be taken by the legitimate system users. This requires defenders and/or tools (EDRs, SIEMs) to determine the intent behind each of the actions, which is a much more complex and difficult challenge. Existing tools and blogs include [1], [2], [3]. Other existing literature is reflected in the Section 2.

This approach allows incorporating real-world data, such as the results of MITRE ATT&CK® Evaluations[3] (hereafter, ATT&CK Evaluations) into attack success metrics and defender response evaluations. As importantly, it allows aggregating data from empirical observations associated with parts of an attack, such as ATT&CK Evaluations or cyber experiments, into metrics describing an attack as a whole.

We demonstrate this practical data incorporation using the APT3 ATT&CK Evaluations data overlaid on an attack scenario developed based on a simplified APT3 attack on a cyber-controlled power grid. In this scenario, an initial infection occurs in an enterprise network, pivots to the power grid control network and affects grid control devices to disrupt grid operations. As in many real-world scenarios, the grid operator has no control on when an attack can start and end or when the attacker who is already in the system and is ready to attack will execute on his objectives.

This example starts as an attack against an enterprise network with the goal of pivoting and acquiring control of an RTU (Remote Terminal Unit) on a SCADA/ICS system. The attacker attempts achieving his goals by initiating attacks against individual users of the enterprise network, in what we call an attack campaign composed from multiple single attacks.

We conduct the analysis in this paper by representing streams of multi-step attacks as discrete-time Markov chains by extending the GPLADD (Graph-based Probabilistic Learning Attacker and Dynamic Defender) framework introduced in [4]. GPLADD provides for a rich and detailed

2. We assume that both attackers and defenders can be individuals, groups, or organizations. Therefore we generally refer to attacker and defender as "they".

3. https://github.com/mitre-attack/attack-evals

representation of attack states steps and the time they take, and detailed attacker and defender strategies, and attack analysis. GPLADD treats time explicitly as a physical constraint and represents explicitly and probabilistically the time different attack steps take. It allows calculating the attack probability of success, probability of detection, costs, and other parameters quantifying the attack dynamics. We approximate GPLADD games by discrete-time Markov chains with states being the success conditions in the attack graph.

Even approximate Markov chain representation is valuable because it allows for analyzing and visualizing the attacker-defender game progress quickly and explicitly, evaluating tradeoffs between detection expenditures at different parts of the defender system, improving sensor placement and assessment strategies, and developing an optimized defender strategy. Representing attacks as Markov chains also substantially improves the analysis scalability because we use we use the standard Markov chain algorithms to estimate attack metrics and to evaluate defender strategies. This means that state-of-the art algorithms for Markov chains can be used to enable fast and efficient computation. Markov representation allows efficient and straightforward estimation of overall attack time-to-success distributions and steady state distributions. Those are used to evaluate attack difficulty and ready-state residence times.

In application to practical attack analysis, Markov chain representation allows calculating measures of attack intensity and estimating the response intensity as well. These intensity measures are necessary to enable comparison and resource allocation between attacks with different success metrics. This is especially important in security, where unlike trust where the defender may have control over attack begin and end times ([4]), there is no beginning end point for the attacks. Therefore, being able to measure and allocate the intensity of defender efforts per unit of time and attack intensity achieves that purpose. Representing the attack as a Markov process with indefinite end time and with steady state distribution gives us the ability to reason about the defender resource allocation.

Explicit Markov chain representation also enables practical incorporation of available attack-related data into the analysis and investigation of attack metrics sensitivity to parameters associated with different parts of the attack. This allows evaluating quantitative improvements to defender policy, even if the data is sparse, unreliable, or missing for parts of the attack.

There are two other alternative ways to analyze GPLADD games that were explored in [4]: analytic solutions and Monte-Carlo simulations. Solving for attack metrics analytically can lead to efficient computation using closed form expressions, but it is challenging to develop for attacks with complicated structure. Monte-Carlo approaches are also challenging for large attacks for scalability and other reasons.

In general, GPLADD games can be approximated as discrete or continuous time Markov processes. We only treat discrete time Markov representation here, primarily because this reflects the nature of attacker-defender interactions better, given that even automated real-time systems will be operating at discrete time intervals. It also allows incorporating real-world data more easily. We start by showing how a continuous time PLADD game can be converted to a discrete-time game and then we show how different data sources can be incorporated. Discrete-time representation also allows treating more general and perhaps more realistic cases because the space of discrete-time Markov processes is "larger" than continuous-time Markov process (see [5] for more information). This therefore also allows operating with discrete-time Markov processes that cannot be derived from continuous-time Markov processes.

Markov chain approximation also works well with the MITRE data on tactics and techniques and with other empirical data in general. A major advantage of discrete-time Markov representation is that the defender can put their own data on the probability of detection or attack disruption into the model with relative ease. Such a data in general may not be publicly available but may be readily available to a capable defender.

This paper treats a single attacker and a single defender. GPLADD can be generalized to multiple coordinated or uncoordinated attackers but doing so it outside of the scope of this paper.

Our novel contributions include:

- A discrete-time Markov chain model to generate multi-step attack metrics
- An approach to estimate attack metric sensitivity to different defender resource allocation to optimize defender resource allocation over security games of indefinite duration.
- A method for representing attacker-defender contests as games of indefinite duration.
- Introduction of the fraction of the time that attack can achieve in the completed or ready state ("Ready" state residence time) as a metric to evaluate defender strategy in games of indefinite duration.
- Explicit estimation of time-to-success distribution for the entire attack.
- A method for representing ATT&CK Evaluations data as GPLADD games and Markov chains.
- A method for generalizing defender capabilities from ATT&CK Evaluations detection categories.
- A method for estimating the quantitative success parameters for attacks.
- A method for evaluating the effects of defender investments in detection at different parts of the defender system by evaluating the attack success metrics and attack transition diagrams.

The key assumptions underlying our method are

- Time-to-success at a particular attack step can be described by a distribution. This distribution can be discrete or continuous. The time-to-success can also be deterministic parameter, which can be represented as a delta-function.
- Attack follows a specific attack plan. This attack plan is represented as attack graph. This assumption is almost self-evident because the attacker technical and information processing capabilities are necessarily limited. It is also based on an anecdotal knowledge that specific attacker groups tend to use specific and stable tactics and techniques for years.

Assumptions associated with specific parts of this paper, such as representing attacks as Markov chains or inferring transition probabilities from MITRE data are described in the corresponding sections.

In this paper, we describe the following:

1) Attack success conditions representation using GPLADD.
2) Attack dynamics approximation using Markov chains.
3) Attack metrics evaluation.
4) An example phishing-based attack chain represented using the APT3 ATT&CK Evaluations data.
5) The methodology for inferring Markov chain transition rates from the ATT&CK Evaluations data.
6) Sensitivity of attack success metrics to detection at different steps of the attack.
7) Analysis of varying defender capabilities inferred from MITRE ATT&CK Evaluations date and those capabilities impact on attack success metrics and the transition diagrams.
8) Applications to defender policy optimization.

The paper is structured as following. Section 2 describes relevant literature. Section 3 describes our approach based on GPLADD games. It shows how this representation can be transformed into a discrete-time Markov chain representing the attack dynamics.

We introduce discrete-time Markov chain representation of attacker-defender interaction in section 3.4.

We develop a Markov chain representation of an example simplified multi-step attack, similar to APT3 as it is described in MITRE ATT&CK framework[4] in section 4 and develop a method to estimate the transition probabilities based on attacker and defender capabilities and available data.

We apply our methodology to analyze the sensitivity of attack success metrics to defender resource allocation in Section 5.1. It shows the sensitivity of Ready-residence time to detection improvements at different parts of attack graph. Section 5.2 shows the analysis of the same attack using the MITRE ATT&CK Evaluations data. It demonstrates non-intuitive differences in attack success metrics corresponding to defenders with different capabilities.

The Conclusions section summarizes and outlines possible extensions to this work.

2 Prior Work

Our theoretical approach builds on and extends our work in building attack models as GPLADD games ([4]). GPLADD is a game-theoretic approach to represent the attack success conditions as attack graphs and to quantify the attack success metrics as a function of the attacker and defender strategies in continuous time. GPLADD can embody a specific attack, represent attack and system parameters and agent strategies, and therefore enable evaluation of the defender strategy and attack success metrics estimation. In this paper, we extend and approximate GPLADD as discrete-time Markov chains as described in Section 3. A more detailed description of GPLADD is provided in 3 Section.

The literature on Markov chains is vast. Markov chains applications to represent agent interactions include [6] and [7] as well as literature on Partially Observed Markov Decision Process (POMDP) ([8]). The [8] provides a model of an attacker-defender interaction represented as a POMDP from the point of view of a defender. The [8] has a somewhat similar formalism as in [4] to represent the attack success conditions, which the authors call a "condition dependency graph". It models the dependency between the attacker progress and the resulting system state and the set of the available exploits available from the attacker. A fundamental difference between the approaches in [8] and [4] is that the latter treats the time required to complete an attack step explicitly. The POMDP-based approaches explicitly treat the uncertainty in the defender knowledge in the system state and the resulting cyber awareness. Markov processes are often used to approximate more general stochastic processes. For example, [9] provides a method for generating approximations of intractable probability distributions using Markov chains. The ability to aggregate the agent-based, and by extension, game theoretic models into a Markov chain approximation, is discussed thoroughly in [6].

A full review of the corresponding literature is outside the scope for this paper. The literature on cyber awareness includes [10] that describes the implementation of their approach on a high-performance computer. The [11] describes a defender strategy based on network segmentation. Game theory and advanced persistent threats (APTs) are discussed in [12].

The MITRE ATT&CK Framework is described in [13]. It contains information on observed adversary tactics, techniques, and procedures (TTPs). MITRE leveraged the information in this framework to organize their ATT&CK Evaluations, which tested the capabilities of EDR (Extended Detection and Response) vendors to detect TTPs associated with specific adversary campaigns. The results of these evaluation provide information that can be used to estimate attacker success and detection probabilities on different steps in an attack. APT datasets are also discussed in [14]. Concrete steps and attacker options at different stages of the attack can be informed from MITRE ATT&CK (see [13], [15], [16] for more information). Additional empirical data on attackers and defenders might be generated using emulation ([17], see also [18], [19], [20], [21]).

The ATT&CK Framework explicitly represents the attacks as multi-step. However, generally there is limited amount of work on multi-step attacks analysis. Existing work includes [22] which applies Bayesian techniques for detection of coordinated attacks and [23], [24], [25], [26], [27] which apply different AI and machine learning techniques. Similar techniques can be applied to inferences of the state of a multi-step attack, given partial and noisy detection on individual attack steps. Attack identification based on multiple detections along the attack progression is outside the scope of this paper, however the section 5.1 specifically provides for evaluation of tradeoff of defender resource allocation associated with detection corresponding to different steps of the attack. This capability to explore correlated detection is especially valuable because the practice

4. We will call this a "notional attack" in what follows to indicate that we represent a simplified version of an APT3 attack.

of correlated detection is still nascent.

The kill chains described in [28] lay out the logical phases in attack progress, as well as provide a high-level framework for reasoning about attacker and defender actions, attack progress, and detection. However, kill chain representation itself does not provide a quantitative framework for attack analysis or for specifying concrete steps in execution of a particular attack. Kill chains do not map directly into GPLADD; they represent a different, higher level of abstraction that may not represent the execution steps, strategies, timings, or dependencies.

Other papers that address detection and consider the dynamic nature of the attacker-defender interaction include [29], [30], [31] and [32]. They deal with the problem of detection related tradeoffs and sensor data aggregation in a dynamic game setting, somewhat similar to GPLADD. Unlike GPLADD, they do not represent the steps in the attack.

The approaches in [33] use detection and false alarm benchmarks to quantify confidence levels in system verification procedures. Hardware Trojan detection methods include [34], [35], and [36], and others. The [37] represent trust in the system as the product of individual subcomponents trust levels. Additional work on the attacker-defender interaction includes [38], [39] and [40] which optimize the Trojan detection technologies.

The security metrics literature includes [41] and work attacks and attack trees modeling that include [42], [43], [44], [45], [46], [47], [48].

3 Approach

We assume that the defender constantly monitors for the attacker activity associated with some or all steps and execution methods of the attack using the defender IDS system and by collecting sensor information to infer and observe the attacker activity. This detection is fine-grained in the sense that defender can associate detection events with possible attack steps and disrupt the attack. Disrupting the attack sends the attacker to the beginning of attack (Step 1) in this paper[5].

We use GPLADD ([4]) to formally represent attacks, attacker-defender interaction, and attack success metrics. In GPLADD, attack is a game between attacker and defender. This game is specified by the attack graph, and the attacker and defender strategies.

We first start with a description of notation and parameters.

Notation and Parameters Summary

The list below summarizes the parameters in the general formulation of the GPLADD game:

- $f(x)$ - the attacker's time-to-success distribution that reflects attacker and defender capabilities. This distribution can be continuous or discrete.
- ψ - an attack graph.
- ζ^H - attacker strategy.
- ζ^D - defender strategy.
- Ψ - an attack represented as a game between attacker and defender as following: $\Psi = \{\psi, \zeta^H, \zeta^D\}$.
- V - a set of all nodes that can be involved in an attack.
- v - a subset of V that resides on defender system.
- v' - a subset of V that resides outsider of the defender system.
- $V_\psi = v_\psi \cup v'_\psi$ by definition.
- V_ψ - a set of all nodes on which attack ψ takes place.
- v_ψ - nodes in defender system on which attack ψ takes place.
- v'_ψ - nodes in defender system on which attack ψ takes place.
- l a chain in the attack graph ψ represented as $l = \{c^l_1, \ldots, c^l_{n_l}\}$.
- c^l_i - logical condition i in chain l.

3.1 GPLADD Games

GPLADD games were introduced in [4]. We describe them briefly here, following [4]. GPLADD is a dynamic multi-step game between attacker and defender. GPLADD represents the multi-step attacks explicitly by constructing multi-step attacks from chains of PLADD games.

GPLADD distinguishes explicitly between an attack and attack graph. An attack is a game between attacker and defender that takes place on attack graph. Formally, a GPLADD game is specified by the attack graph and the attacker and defender strategies and actions at different stages of the attack. The game dynamics is affected by the attacker and defender strategies.

Attack graph defines a partially ordered set of conditions the attacker must achieve for the attack to succeed. Those conditions can be defined over the nodes in the defender system (e.g., attacker gaining access to a version control server) or nodes outside of the defender system (e.g., the attacker making modifications to a stolen file using the attacker own system). Those individual conditions a generally represented as PLADD games defined in [49]. A PLADD game is a game for control between attacker and defender over a single asset. A GPLADD game therefore consists of multiple PLADD games connected by a graph structure and is a partially ordered collection of PLADD games. These PLADD games can take place on certain parts of the defender system or outside of it.

Informally, in GPLADD, the attacker must complete a set of challenges, that constitute attack graph. Each challenge is represented by a PLADD game. The attacker activity while completing these challenges provides the defender with an opportunity to detect and disrupt that attack.

The time dynamics of GPLADD game is a stochastic process that is determined jointly by the attack graph and the attacker and defender capabilities and strategies.

3.2 PLADD - Building Blocks for GPLADD

PLADD (Probabilistic Learning Attacker and Dynamic Defender) games are single-step games that represent attacker-

5. In practice the defender may not have the capability to disrupt the attack completely, and so may disrupt the attacker progress only partially, for example by identifying and cleaning the malware used for a particular lateral movement. Partial attack disruption sends the attacker to one of the previous steps of the game that is not Step 1. Because of data limitations and for simplicity, we only consider full attack disruption here. When this approach is used by a defender team, their knowledge of their systems and capabilities can be used to refine the attack disruption representation.

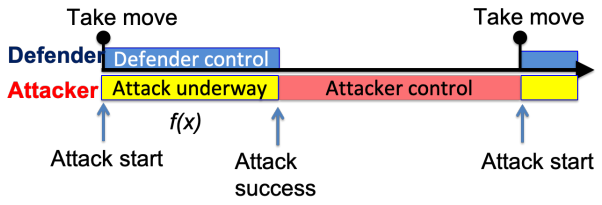

Figure 1: PLADD Game: blue indicates defender control, red shows attacker control, and yellow shows attack in progress.

defender contests for a resource. PLADD games were introduced in [49]. In PLADD, the defender has one or more different moves ("take" and "morph" in the original PLADD) to assert control over the resource. The attacker has only one move that represents the initiation of the attack. We also call it a "take" move. The attacker take move does not succeed immediately, but rather after a period of time termed time-to-success that is represented as a random variable with the time-to-success distribution denoted as $f(t)$. Once started, the attack on a node v_i succeeds after a certain delay, which we call the time-to-success. The time-to-success can be either deterministic or probabilistic. The time-to-success is a node-specific random variable distributed according to $f_i(\cdot)$ for all $v_i \in V_\psi$.

The time-to-success distribution reflects features of the node, of the defender system, as well as the defender and attacker capabilities. It can also be interpreted as a reflection of the difficulty of the corresponding stage in the attack. It reflects both the attacker capabilities and the defender system with security measures in place. An example distribution for a specific example is described in [50]. A study by [51] evaluates possible distributions and their parameters for time to compromise of a computer system based on empirical data.

The defender "take move" sends the attacker to one of the previous steps of the attack or to the beginning of the attack. We assume the defender can take immediate control of their system, incurring appropriate cost for that action (such a cost may include rendering the system inoperable for a period of time).

This attacker-defender contest can therefore be represented as a sequence of attacker and defender moves. In an example of a single asset, the defender may make the take move on a system that is under attacker control. This interaction is illustrated in Figure 1. It represents alternating periods of the defender control (blue bars above the x-axis) and attacker control (red bars below the x-axis), .

3.3 Transitions on Attack Graph

The same back-and-take between attacker and defender applies to the entire attack graph. The defender can make take moves on a pre-determined schedule, at random, or because of detecting the attacker activity at different parts of the attack graph, each of which is an individual PLADD game. This gives rise to a set of transitions on attack graph, that we represent as a Markov process in this paper. This means that attack graph is a game board on which the attack "game" is played.

We assume that attacker activity in defender system gives the defender an opportunity to detect that activity and disrupt it with take moves. Depending on the defender capability level, they may be able to associate the detection events it with an attack step and a part of the defender system. Both empirically and anecdotally such capabilities vary substantially across different defenders. Detection is probabilistic and is affected by the defender capabilities and allocation of resources to sensing and assessment of specific attack indicators. Allocating more resources to detecting specific events may increase the probability of correctly detecting the attacker activity, but it also increases the defender costs and potentially reduces resources available to other parts of the system.

An attack Ψ is formally defined as the following: $\Psi = \{\psi, \zeta^H, \zeta^D\}$, where ψ is an attack graph, and $\zeta^H(\psi)$ and $\zeta^D(\psi)$ are the attacker and defender strategies.

The attack graph ψ represents both the success conditions as the nodes in ψ and the logical dependency between different steps in the attack as the links on the graph. These steps are represented as a set of directed acyclic chains. For attack to succeed, the attacker must complete all the steps.

The set of terminal nodes of all those chains in the attack graph constitute the necessary and sufficient conditions for the attack success.

A single chain l in the attack graph ψ is $\{c^l_1, \ldots, c^l_{n_l}\}$, where the superscript l signifies that this is a l-th chain in the attack graph ψ. Each element in l is a logical condition on V_ψ. The term $c^l_{n_l}$ denotes the terminal node in chain l. The numbering of the chains does not imply any ordering or temporal precedence, but the numbering of the elements in each chain does imply a temporal precedence.

The attack graph ψ is represented as follows:

$$\psi = \{\{c^1_1, \ldots, c^1_{n_1}\}, \ldots, \{c^m_1, \ldots, c^m_{n_m}\}\}, \tag{1}$$

A condition c^j_i in ψ may represent a specific system state or a change in the system state. For example, the logical condition for attacker control on node v_i is represented as $s(v_i) = 1$ where s is the state of the node which can take two values: $\{0, 1\}$, where 0 denotes defender control, 1 stands for attacker control [6].

Having all terminal conditions in ψ being true at the same time is necessary for the attack success [7]. This is expressed as:

$$S(\psi) = \{c^1_{n_1} \wedge \ldots \wedge c^m_{n_m}\}, \tag{2}$$

where S represents "attack success" and $\wedge$ represents logical "AND".

6. In the original continuous-time GPLADD in [4], there is a third state that indicates that a node is under attack, but is still under defender control because the attack has not succeeded yet. This state is not needed for Markov chain representation in this paper and is omitted for simplicity.

7. This does not mean that success on all those conditions must be achieved simultaneously. Success can be achieved at different times. However, a condition achieved earlier must stay in the "success" state. For example, a file that was successfully modified by the attacker prior to being injected into the defender system must be current and relevant for such injection. For example, it may no longer be current if defender changed their system so much that the modified file will no longer work.

An example attack and a more extensive discussion on the attack graph specification can be found in [4].

3.4 Discrete-Time Markov Chain Approximation

The satisfaction of different attack success conditions means that the attack progress to the next step, as defined by attack graph. This therefore gives rise to a transition structure and a stochastic process on attack graph with probabilities that can be determined as in [4]. The attack analysis in continuous time is shown in the [4], which also incorporates a relatively rich state space that includes attacks in progress on individual nodes.

There are situations when attacks analyzed in [4] are Markov processes in continuous time[8]. In this paper, we simplify this further to conduct the analysis in discrete time and in a relatively simple state space that only includes information on whether an attack step has been completed or not. Transitioning to discrete time and a simpler state space would generally be an approximation of a richer model, however it allows incorporating and reflecting certain important features of the real attacks:

- Real-world attackers and defenders have limited information processing capabilities and tend to operate at discrete time intervals using simplified representations of the system
- Information to ascertain the full system state is generally not fully available to either attacker or defender. Both the attacker and the defender may try to obfuscate it to deny information to the other party.
- When defender design their system, they are unlikely have information about specific attacker strategies and have to design a system against a set of possible attackers

Taken together, these features mean that the defender needs a "coarser" representation to enable practical evaluation of attacks even when full information is lacking or unobtainable. Representing attacks as discrete-time Markov chains, provides significant practical benefits and a relatively quick way to estimate attack success metrics. An additional benefit of such approximation is that it allows to represent games of indefinite duration and to estimate both the time-to-success and attacker ability to remain hidden metrics using the standard tools for Markov chains.

The representation of GPLADD games as continuous time Markov chains can be exact in certain situations, as can be derived from analysis in [4]. Depending on the assumptions about the time-to-success distributions and the attacker and defender strategies, the Markov representation can be exact or approximate. This can happen when the attack graph has a complicated structure or when agent strategies have long dependencies on attack history. However, for practical reasons described above, we focus on discrete-time Markov chains here. Discrete time Markov chains will generally be an approximation of continuous-time GPLADD games. This approximate representation of game dynamics allows a simpler and often practically useful and sufficient treatment of certain aspects of the defender strategy optimization It provides several benefits: tractability, scalability, ease of interpreting the attacks, effects of attacker and defender strategies, and understanding the results.

In practice, non-Markovian dynamics can be converted to Markovian by expanding the state space. While the definition of a stochastic process state can often be expanded to make the dynamics Markovian, it is likely that not all attacker-defender games can be represented in a practically useful way as Markov chains. For example, a defender may keep a running tally of detections across all attacks by all attackers and drastically change the defender system when the tally reaches a certain point in a course of a particular attack. By construction, this process will look non-Markovian for the attacker who is executing the attack when the tally reaches the threshold because this attacker will not have access to the defender state space representation and the running tally. The possibility of non-Markovian transitions is raised in the context of POMDPs by [52]. As shown in [52] this is resolved in POMDPs by introducing the agent belief state, whose updating is Markovian. A similar approach may be used in the previous example with the defender running a tally of detected attackers, where the attacker may form a belief distribution over possible system states that includes different tally counts.

Here, we approximate GPLADD games with discrete-time Markov chains. A somewhat different approach to represent attacker-defender interaction as Markov Games is shown in [53].

We develop the Markov analysis and approximation for "linear" attack graphs of the form:

$$\psi = \{c_1, \ldots, c_n\}, \tag{3}$$

where the attacker must achieve the attack success conditions sequentially and where the only terminal attack success condition is c_n. The attack steps correspond to the success condition in this example[9]. We therefore can treat a success condition c_i as achieved or not achieved. The only pre-requisite for attacker being able to go from attack step i to step $i+1$ is achieving success condition c_i. This allows us then to represent the state of the attack as the currently achieved success condition i. The Markov representation can be extended to attacks with "parallel" chains and conditions but is out of scope for this paper.

If the dynamics of state transitions on 3 is Markovian, then in can be represented by an n by n matrix $tr_{i,j}$ that represents both attacker successes and failures in moving along the attack graph. The matrix $tr_{i,j}$ defines all possible transition probabilities between the n attack steps in Equation 3:

$$tr_{i,j} = P(\Sigma(t+1) = j \mid \Sigma(t) = i), \tag{4}$$

where $1 \leq i, j \leq n$, $\Sigma(t)$ represents the attack state at time t, and $P(i,j)$ is the transition probability from state i to state j at consecutive times.

The matrix $tr_{i,j}$ has the following properties, for $\forall i$, and $j \leq n$, given the attack graph definition:

8. Although, this analysis is unpublished.

9. In general, the attack steps can be a combination of attack states corresponding to different conditions if those attack steps can be executed simultaneously as shown for a more general attack in [4].

$$tr_{i,i} \geq 0, \tag{5}$$
$$tr_{i,j} > 0, j = i + 1 \tag{6}$$
$$tr_{i,j} = 0, j > i + 1 \tag{7}$$
$$tr_{i,j} \geq 0, j < i \tag{8}$$
$$\tag{9}$$

The condition 5 means that the attacker may spend more than one time step at a particular attack step, the condition 6 is necessary for attack to be feasible. The condition 7 signifies that all attack steps are necessary and cannot be "jumped" over by the attacker. The condition 8 means that the defender may have an ability to disrupt the attacker progress and send the attacker to the beginning of one of the previous steps of the attack. The matrix $tr_{i,j}$ reflects both the attacker and defender capabilities for a given system. Its values may change if either attacker or defender capabilities or the system change.

If the underlying GPLADD game is fully specified, then the transition probabilities can be calculated using the time-to-success and detection probability distributions. If those distributions are only known for certain PLADD games, then that data can be used to populate a part of the matrix in 4. If data is available from other sources, such as experiments or IDS systems evaluations, it can be used to populate transitions for other PLADD games or to refine the existing transition probabilities. If the data combined from different sources is used consistently and carefully, the results preserve the attack metrics ranking and can be used to improve the defender posture even if the results are not fully calibrated or if there is missing data for parts of the attack.

This is a summary of how the transition matrix tr is created:

- We introduce a set of attack success and detection probabilities associated with each step in the attack graph.
- We interpret detections as giving the defender the ability to disrupt the attack and to send the attacker to the very first step or to one of the preceding steps in the attack.
- Attacker success for each step means moving to the next step and avoiding detection.
- The attacker may also stay at the same step of the attack in multiple time steps, if they did not succeed in executing the methods required to getting to the next attack step and at the same time has not been detected.
- Therefore, each step has success (moving to the next step and not getting detected), failure (detection), and stay-in-place probabilities.

3.5 Markov Chain Transition Probabilities Inference

The Markov chain representation is then can be inferred from the time-to-success distributions and attacker and defender strategies, and from available data. The key task is to populate the tr matrix. Even if some data is missing or imprecise, attacks and defender strategies can still be compared and ranked for the same defender and the same system.

To infer transition matrix tr PLADD games, we introduce a fixed time step duration for all attack steps, δt[10].

The transition probabilities are dependent on the time step duration, because longer time-step duration weekly monotonically increases the probabilities of both success and detection. When the time-to-success distributions for certain PLADD games are known, the raw probability of success for each attack step is calculated by integrating the corresponding time-to-success distribution $f(t)$ for each step starting at zero and with the corresponding time step duration as the upper bound. The effective probability of success for an attack step is calculated as the raw probability of success times the probability that detection did not occur, under assumption that success and detection are independent random variables. The raw time-to-success distribution as it is defined in this paper is independent from the detection distribution. This calculation is approximate because we are multiplying the success probability by the no-detection probability for the entire time interval to estimate the effective success probability. If both time-to-success and detection probabilities are known, the effective probability of success in a time interval can be calculated more precisely by integration (if distributions are continuous) or summation (if discrete) of the conditional probability of success at a particular point in time, given no detection (represented as product of probability of success and probability of no detection). Given that the probabilities we use are estimated, we believe this level of precision is not needed in this paper.

When time-to-success distributions are not known, but there is data on detection or success probabilities associated with attack step, these probabilities can be used to estimate the transition probabilities of both the attacker going to the next step and attacker being disrupted.

We demonstrate our approach using a simplified APT3 ("notional" in what follows) attack represented as a GPLADD game and treated as a Markov chain.

We create the Markov chain representation of the game between the attacker and the defender for a notional attack example and estimate the Markov chain transition probabilities using two different kinds of data sources.

The first type of data sources is described in the Section 5.1. It is based on estimating the probabilities of success events on public data and SME knowledge.

The second data source we use is the MITRE ATT&CK Evaluations data for the APT3 attack emulation.

Those two data sets can be combined in a single analysis, but this is outside of the scope of this paper.

We show creation of the Markov chain representation for notional APT3 attack in the Section 4.1. The first method and the notional attack are described in the section 5.1.1 .

The methodology for estimating the detection probabilities and Markov chain parameters from the ATT&CK Evaluations is described in the Section 5.2. It then applied to comparison of different defender performance against the same attack and the same attacker.

10. This duration does not need to be the same for different attack steps. It is assumed to be the same for all attack steps for simplicity.

Figure 2: APT3 attack steps.

4 Simplified APT3 Attack Definition

4.1 Attack Definition

We start with a GPLADD that consists of multiple PLADD games and translate it into a Markov chain representation. We analyze a simplified version of the APT3 techniques as described in the MITRE ATT&CK Framework. The attack chain starts with a phishing attack, and then pivots through the network with the goal of gaining access to an ICS (Industrial Control System) and taking control of an RTU or a set of RTUs.

The defender network consists of an enterprise network, an ICS network, and a DMZ (demilitarized zone subnetwork). The ICS controls an electric power system via RTUs.

This attack goal is to take control of one or more RTUs in an energy distribution system. The attack consists of an "initial foothold" phase aimed at the enterprise network, a pivot phase from the Enterprise network to the ICS, and the RTU control phase.

In the initial foothold phase, the attacker aims to establish presence on a computer inside of the enterprise network with the goal of discovering an opportunity to pivot to a workstation on the ICS network. The attack steps are visually represented in Figure 2.

The attack begins with the attacker sending a phishing email to induce potential victim to follow a link. The attack graph is represented as a set of success conditions:

$$\psi = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}, \tag{10}$$

where the intermediate attack success conditions, also represented in Figure 2, are as follows, with a short name in () indicating the attack step abbreviated description:

- c_1 - Attack Start (Start).
- c_2 - Email delivered to user (Email).
- c_3 - User follows a link to a malicious website, designated as badguy.com (Link).
- c_4 - The malicious code (trojan) from the website executes on the user computer (Exec).
- c_5 - The trojan obtains "address" of an engineering workstation. This is a waiting stage, the user of the infected machine either has access or doesn't have access to the engineering workstation (IPEW).
- c_6 - After engineering workstation connection is established, the trojan sends a message back to the command-and-control center (Msg).
- c_7 - Once the user connects to the engineering workstation, the trojan moves to the workstation (MvEW).
- c_8 - The trojan finds RTUs and targets them (RTU).
- c_9 - When RTUs are found, the trojan is in attack "Ready" state. The "Ready" state is achieved if at least one RTU is found (Ready).

This readiness in the "Ready" state denotes the fact that the attacker has successfully completed the infiltration to their desired network location, but before the attacker has executed their final objective (e.g., exfiltration of data or changes to control systems). The fraction of the time when the attacker is "ready" to execute on their objectives is represented by the steady state fraction of time the attack spends in the "Ready" state.

An attack succeeds if all the terminal conditions are satisfied. For this attack, the attack succeeds when the condition c_9 is satisfied.

As explained before, this attack graph does not uniquely determine the attack pathways. The attack can follow many pathways even for this simple single-chain attack graph shown here, depending on the attacker and defender actions. These actions are determined by the attacker and defender strategies, described below.

4.2 Attacker strategy

The attacker strategy is composed of active components that are relevant to the interaction with the defender systems and in preparatory components that are relevant as a reflection of the attacker background and other capabilities available for the attack. We treat these background and prepared capabilities as static during the attack and use only information on active components of the strategy in the game analysis.

The purpose of introducing an attack campaign is twofold: to represent attackers who may attempt multiple attempts at getting into a particular system; and 2) to enable representation of the attack difficulty via the analysis of the corresponding stochastic process and to allow certain parameters of such process, as steady state distribution, expected time to success, etc. to be interpreted as attack metrics. As a part of the attack campaign, the attacker sends emails at certain intervals, determined by their strategy. The emails spread over time to decrease the chance of being noticed by the IDS system of the target organization. We assume that the attacker sends just one email at any given time to avoid detection.

We consider here an "eager" attacker strategy, defined in [4] as the attacker starting at the next attack step as soon as all the preconditions are satisfied.

4.3 Defender Strategy

The defender runs an IDS (Intrusion Detection System) that attempts detecting malicious activity. The specific tools and techniques used by the defender affect the probability of detection at different stages of the attack. These probabilities are also affected by the structure of defender networks and the attacker capabilities. Changes to the structure and operation of the Enterprise and the ICS networks, such as segmenting the network, can be deployed as mitigations to decrease the attack effectiveness or impacts.

If detection occurs, the defender can roll back some or all the attacker success. In this paper, we assume that detection of the attack at any stage terminates and completely disrupts that attack, and as the result the attacker moves back to the "Start" step of the attack. Relaxing this assumption and allowing the attacker to persist at previous steps of the attack will not change the analysis methodology but requires a more extensive data gathering effort.

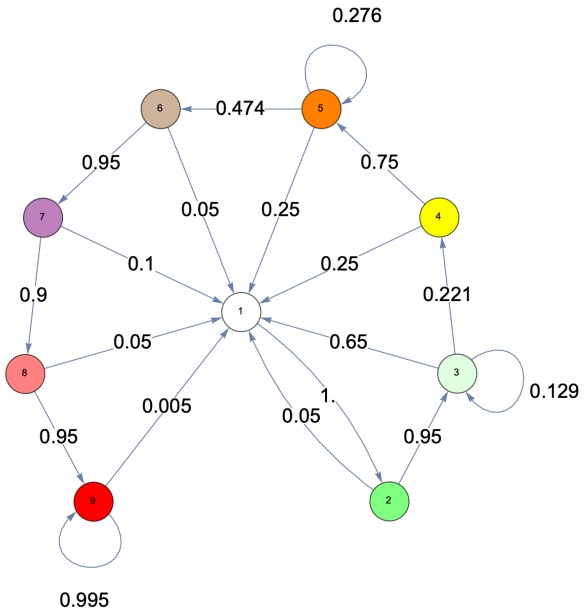

Figure 3: A Markov chain representation of notional attack.

If the detection does not occur, there are two other possible outcomes: the attacker may succeed or may stay at the same step. Specifying all such probabilities under the Markov assumption that only the current state affects the probability distribution of the future states gives rise to a Markov transitions matrix. This matrix can be represented as a graph showing the transition probabilities between the states. We will call this graph a (Markov) Transition Diagram. An example Markov Transition Diagram, along with the transition probabilities, is displayed in Figure 3. The same transition probabilities are displayed in Table 1, which constitutes an alternative and equivalent way to represent the same information.

The structure of the Markov chain is the result of the event probabilities in the underlying GPLADD model and of the attacker and defender strategies and capabilities. Its calculation is described in the Section 5.1.1.

5 Results

Our analysis was done using the following tools: Mathematica for graph representation, Markov chain analysis and simulation; Python for MITRE data analysis and detection probability estimation.

5.1 Attack Metrics Sensitivity to Detection

Integrating the time-to-success distributions at each time step to estimate the success probability, estimating the detection probability based on available data or using the available subject-matter expert knowledge, and calculating the probability of the attacker remaining at the same time step by treating the success and detection for each time step as independent events, so that the resulting probability of all three possible events at each time step is 1.

Our primary goal on this stage of the analysis is to illustrate the ability to use the Markov chain representation of GPLADD games to calculate the attack success metrics and the defender ability to allocate their resources to degrade the attack success metrics.

We show that the defender effort at different parts of his system has substantially different effect on the attack success metrics, on the transitions and the structure of the Markov graph representing the attack, and on the overall attack difficulty. We show that the defender can change the attack transition graph by changing his strategy.

This analysis can assist in success metrics quantification, representing the attack dynamics, attack ranking, identification of improvements to defender system, ranking the improvements, improving defender responses and strategies, system hardening, and mitigations.

5.1.1 Transition Matrix from Literature and SME Data

The time-to-success distribution parameters are based on the SME Evaluations and published research. For example, according to [54], about 30% of phishing emails are opened by their targets. The authors in [54] also describe certain attack success parameters dependence on the attack technique used. This and other information have been incorporated into the attack representation.

In this method, the probability of detection is based on SME information and in some cases on the parameters inferred from published research.

There is also a probability that the attacker will not be detected and will not be able to complete an attack step. This is calculated as one minus the detection and effective success probabilities for the corresponding time step.

This and the approach described above based on the ATT&CK Evaluations data allow creating a Markov chain representation with the states of the Markov chain based on nodes in ψ.

A somewhat different method for creating a Markov chain representation for an attacker-defender interaction can be found in [55].

We have created a Markov transition matrix based on this method and populated it with transitions probabilities with the $\delta t = 1hr$. This transition matrix is shown in the Table 1.

Table 1: The Markov process transition matrix showing the transition probability for each attack step. The transition probabilities were inferred from the available literature and/or based on subject matter expert opinions. The time step for which the probabilities were evaluated is 1 hour.

Step									
	1	2	3	4	5	6	7	8	9
1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.05	0.0	0.95	0.0	0.0	0.0	0.0	0.0	0.0
3	0.65	0.0	0.13	0.22	0.0	0.0	0.0	0.0	0.0
4	0.25	0.0	0.0	0.0	0.75	0.0	0.0	0.0	0.0
5	0.25	0.0	0.0	0.0	0.28	0.47	0.0	0.0	0.0
6	0.05	0.0	0.0	0.0	0.0	0.0	0.95	0.0	0.0
7	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.0
8	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.18	0.32
9	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.95

At this point, they should be treated as rough approximations for three reasons: 1) they are based on the available

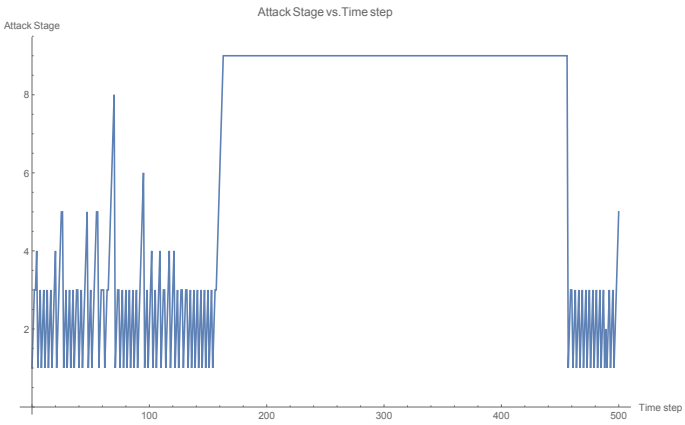

Figure 4: A simulation of attack state over time.

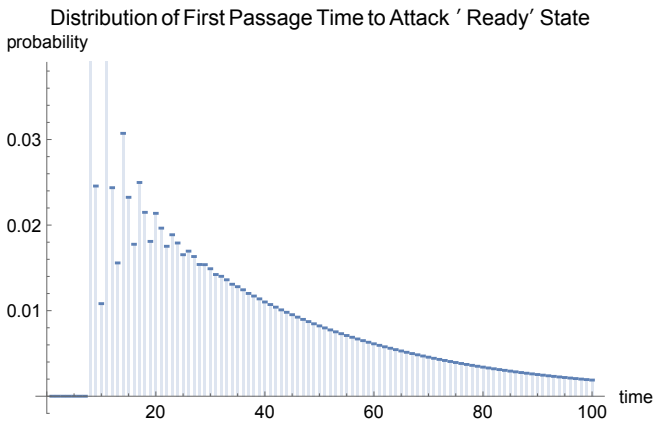

Figure 5: The distribution of the time of the first passage to the "Ready" state.

data and SME opinion about such probabilities in general and as such can only reflect the averages describing the current and recent state of practice in cyber security; 2) they vary significantly between different defender capabilities as shown in Section 5.2.1; 3) they can change with improvements in defensive or offensive capabilities. Certain parameters can also be informed in the future by cyber and emulation-based experimentation and refined with uncertainty quantification.

5.1.2 Sensitivity Analysis

We start with representation of the attack dynamics over time. Figure 4 shows an example of simulated time dynamics of an attack, where the attack succeeded sometime before the time step 200, and then was subsequently detected sometime after time step 400, thus allowing the attacker at least 200 time steps to execute either a single, or a set of coordinated attacks. Unlike the fraction of the time the attack spends in the "Ready" state, which is based on the steady state of the Markov process, this plot illustrates a single realization of the attack, that may differ between different realizations. In the realization shown here, the attacker spends a lot of time at the steps 1 - 3 of the attack, and only occasionally reaches the steps 4, 5, 6, and 8, only to be quickly dislodged from there. When the attacker finally reaches the step 9 "Ready", he stays there for a while, because of the low detection probability in that state.

Figure 5 shows the distribution of the attack first passage time to the "Ready" state. The distribution of the first passage time to the "Ready" or other states reflects the attack expected progression speed and improve the defender policies.

The time shown on the X-axis is effectively the number of steps in all the attacks done as a part of the attack campaign until success. If a successful attack campaign for a linear attack graph, such as our notional attack, has more steps than the attack graph, it means that one or more of the attacks have failed, and the attacker had to restart the attacks.

Given enough time and an opportunity to conduct the attack campaign even when a set of attacks in the campaign have been disrupted, the attacker eventually succeeds with high probability in this example.

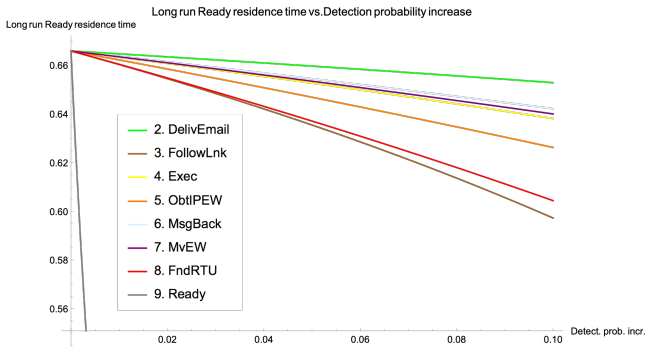

Figure 6: Declines in the attack time spent in the "Ready" state vs. investments corresponding to detection at specific stages of an attack.

On the other hand, the probability of the attack succeeding undetected (in the case of the notional attack this means succeeding in 9 steps) is small, perhaps about 4%.

Figure 6 shows how the fraction of time the attack spends in the "Ready" state is affected by the improved detection at specific stages of an attack. One clear implication is that from the cost-benefit point of view, it is best to detect the attack at the last, "Ready", state. The intuition for this result is very simple: given that the attack spends most time in the "Ready" state and little time in any other state, the chances of detecting attack in other states are low, given that most of the time there is nothing to detect there. However, this may not be optimal from the risk-minimization point of view, because if detection at the "Ready" state fails, then there are no other options available for the defender to stop the attack. The ability to quantify the effects of improvements in detection corresponding to different steps in the attack, allows the defender to improve their resource allocation and can be used as an input for the defender strategy optimization.

We also observe here that making a single step in the attack difficult generally does not have a strong effect on the fraction of the time the Markov chain steady state distribution assigns to the "Ready" state, because once that difficult step is passed, there is nothing else that stops the attack from succeeding and we conjecture that changing the

defender system so that there are more ways in which the attack may fail or be pushed back to one of the earlier stages may be an effective heuristic for defender optimization.

5.2 MITRE ATT&CK Evaluations Analysis

MITRE ATT&CK® is an industry-standard framework that captures adversary behaviors as they have been observed, and publicly reported, in real world attacks and intrusion campaigns. This framework, and the underlying cyber threat intelligence, informed the MITRE ATT&CK® Evaluations (Evaluations). The Evaluations team built a test plan that emulated the cyber intrusion steps demonstrated by a known threat actor, APT3 or Gothic Panda. The emulated scenario was then tested against Endpoint Detection and Response (EDR) products of companies interested in demonstrating their product's detection capabilities. The released results represent a set of data that can be leveraged to estimate attacker success in similar scenarios or in scenarios that use techniques tested during the Evaluations.

In this section we estimate the Markov chain parameters directly from the MITRE ATT&CK Evaluations data and skip integrating the time-to-success distributions.

5.2.1 Detection Probabilities

Many steps in the previous GPLADD scenario could be directly mapped to techniques tested in the Evaluations. In this new work, we leveraged Evaluations data to approximate probabilities of detection to inform the Markov chain simulations of GPLADD models. We effectively associate an attack step with a technique (e.g., gain persistence). This allows hot-swapping techniques for a given tactic to understand the uncertainty associated with options available to the attacker.

To calculate these probabilities, each step of the scenario in our prior work was mapped to a step in the Evaluations test plan, as outlined in Table 2. In the Evaluations, each step of the test plan was made up of sub-steps that tested detection for one ATT&CK Technique. Because our scenario steps more closely aligned with the test plan steps, rather than the sub-steps, we calculated probability of detection for a step as the ceiling of the probabilities of its sub-steps. Probabilities of detection for a sub-step were the number of detections (detailed below) by unique vendors divided by the total number of vendors (twelve) who participated in the Evaluations.

Therefore the stay-in-place probability is not present in the MITRE ATT&CK Evaluations data and is not treated in this section. This can be interpreted as the attack time step being so large that either attack success or detection will certainly happen.

In the real world, detection depends on the tools and capabilities of the defenders. We tried to capture that dependency by using the "detection categories" defined in the Evaluations and mapping them to a level of defender. The mapping was based on the level of ambiguity that human defenders would be required to deal with. The three ATT&CK Evaluations categories used in defender capabilities were Indicator of Compromise (IOC), Specific Alert, and General Alert. IOCs provide the highest level of confidence that a system artifact is malicious, as it has been previously identified and labeled as such. Accordingly, the capability to recognize a threat using IOCs requires the lowest investment in defender skills and tools. Specific Alerts signal high confidence that activity or artifacts are malicious, but they may still require investigation by a more skilled defensive team. General Alerts are the most ambiguous of the alert categories, and thus require the most defender expertise when determining malicious activity or artifacts. These defender parameters are summarized in Table 2. This ATT&CK Evaluations data was empirically generated for a given attacker capability. If the data for different attacker capabilities is available, it can be incorporated into this analysis seamlessly, in the same way as the data for different defender capabilities.

Table 2: Detection and Defender Capabilities

Defender Name	Level of Ambiguity	Detection Categories
Blue 0	None	IOC
Blue 1	Medium	IOC, Specific Alerts
Blue 2	Most	IOC, Specific Alerts, General Alerts

Each level of defender capability increases the ability to deal with ambiguity, as outlined by the table. Probabilities of detection were calculated, as detailed above, for each of the three detection categories. For a particular level of defender, their overall probability of detection was determined by the ceiling of the probabilities of detection for their relevant detection categories.

We have further considered the two versions of the APT3 attack: 1 and 2 as outlined in the table. Taking some steps from each variation of APT 3 as emulated by the Evaluations team allowed us to create two unique scenarios that mapped to the GPLADD (notional attack) index. Additional information on the MITRE ATT&CK chains specification and representation can be found in [16].

Table 3: Attack Chains used in the analysis.

Step	GPLADD Index	Att. Chain 1	Att. Chain 2
1	Start		
2	Email		
3	Link		
4	Exec	1.A.1	11.A.1
5	IPEW	15.B.1	5.A.1
6	Msg	1.C.1	11.B.1
7	MvEW	6.C.1	16.E.1
8	RTU	4.A.1	13.A.1
8	Ready	7.C.1	17.C.1

Applying the methodology described above gives us the probabilities of detection associated with each step for each defender type and each attack variant, as represented in the Table 4. We will refer to the combination of defender type and attack variant as Evaluations.

For the analysis, each raw in the table 4 is converted to the corresponding transition matrix, introduced in Equation 3. For example, the B10 set of capabilities will result in a very simple transition matrix where almost all elements of the form $tr_{i,i+1}, i+1 \leq 1$ will be 1, except $tr_{6,1}$ that will be 0.92 (1 - 0.08), etc. The full transition matrix for B10 is shown in Table table:B10TransMatrix.

Table 4: Detection probability by attack step for each evaluation (E). First number after B is the attack variant (1, 2) and second number is the defender type (0, 1, 2).

E	Step								
	1	2	3	4	5	6	7	8	9
B10	0.0	0.0	0.0	0.0	0.0	0.08	0.0	0.0	0.0
B11	0.0	0.0	0.0	0.58	0.08	0.17	0.0	0.0	0.67
B12	0.0	0.0	0.0	0.58	0.08	0.17	0.17	0.25	0.67
B20	0.0	0.0	0.0	0.17	0.0	0.08	0.0	0.0	0.0
B21	0.0	0.0	0.0	0.75	0.5	0.17	0.0	0.0	0.42
B22	0.0	0.0	0.0	0.75	0.5	0.17	0.08	0.42	0.42

Table 5: The Markov process transition matrix for defender B0 and Evaluation B10.

Step									
	1	2	3	4	5	6	7	8	9
1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
6	0.08	0.0	0.0	0.0	0.0	0.0	0.92	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

5.2.2 *Defender Performance*

We compare and contrast here the attack results analysis for the three kinds of defender and two kinds of attacks described in section 5.2.1 and specified in the Table 4, designated as Evaluations B10 . . . B22.

Our primary inferences from this analysis are threefold: 1) we conclude that even state of the art capabilities available on the market at the time of the original Evaluations, may not be able to prevent certain APT threats from succeeding; 2) there is significant variation in outcomes between defenders of different capabilities and outcomes; 3) the ability to increase the difficulty of the attack at a single step is insufficient to make the overall attack difficult. Rather, resource allocation must be optimized across the enterprise based on data corresponding to different steps in possible attacks.

We have analyzed the Markov chains corresponding to detection probabilities in the Table 4. Figures 7 - 9 show how the ability to detect the attacker at different steps in the attack changes the structure of the corresponding Transition Diagram. Qualitatively, the results for the attack variants 1 and 2 are very similar, so we describe here the analysis for the attack variant 2.

Figure 7 shows the transition diagram for evaluation B20 from Table 2.

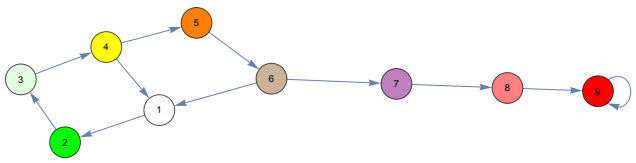

Figure 7: The transition diagram for Defender B0 and Evaluation B20 from Table 2.

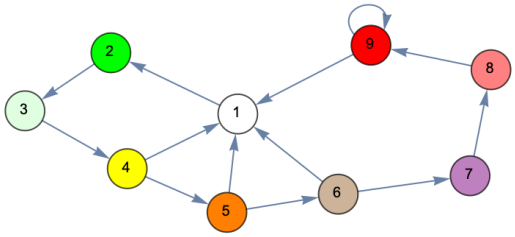

Figure 8: The transition diagram for evaluation B21 for defender B1.

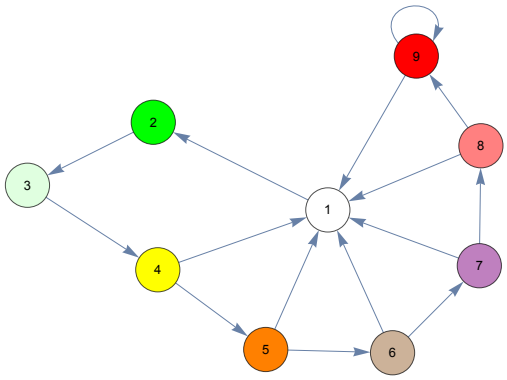

Figure 9: The transition diagram for evaluation B22 for defender B2.

In evaluation B20 the defender, B0, is not very capable, because they can detect and disrupt the attack only at the earlier attack stages; once the attacker succeeds at the step 6, there is no chance of detecting and arresting the attacker progress anymore.

Figures 8 and 9 show defenders with improving capabilities. In Evaluation B21 (8), the defender B1 is better able to detect and respond to threats than defender B0. This is reflected in the fact that the transitions diagram for B21 has more ways that B1 can disrupt the attack than B20.

Defender B2 (Figure 9) has the best capability of all three. They are able to roll back the attacker progress at most steps in the attack.

Understanding of the attack progress can be improved by augmenting it with the quantitative attack success parameters. Given the corresponding Markov chain transition probabilities, we can calculate the steady state of the Markov chain and a measure of the time-to-success distribution. We define the time-to-success distribution as distribution of the first time the attacker reaches the "Ready" node.

We first analyze the steady state distributions corresponding to the Evaluations B20 - B22 and see that they paint even more drastic difference between the defenders B0, B1, and B2.

B0 effectively has no defense against the attack - the steady state distribution as shown in Figure 10 has all its weight on the "Ready" state. This can be interpreted as a reflection of the fact that if the attacker is able to repeat their campaign until step 6 succeeds, the defender has no ability to revert that outcome, unless the defender rebuilds his system from scratch (this option is not considered in this

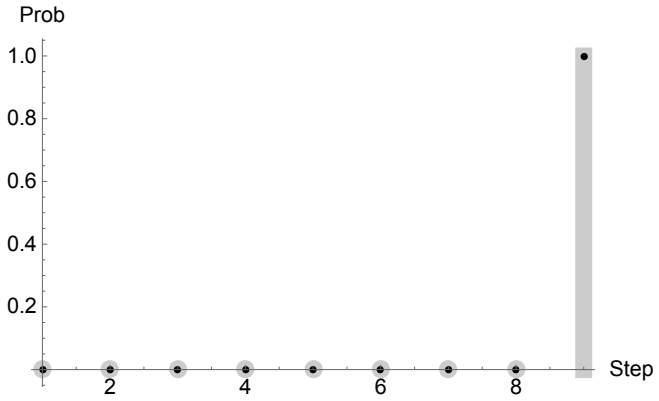

Figure 10: The Markov chain steady state distribution for evaluation B20 and defender B0.

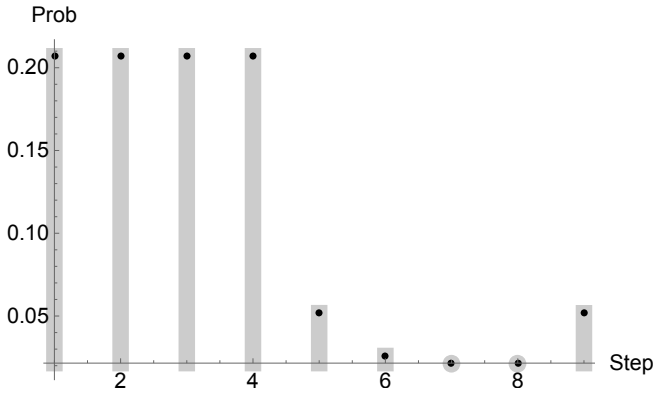

Figure 11: The Markov chain steady state distribution for the evaluation B21 and defender B1.

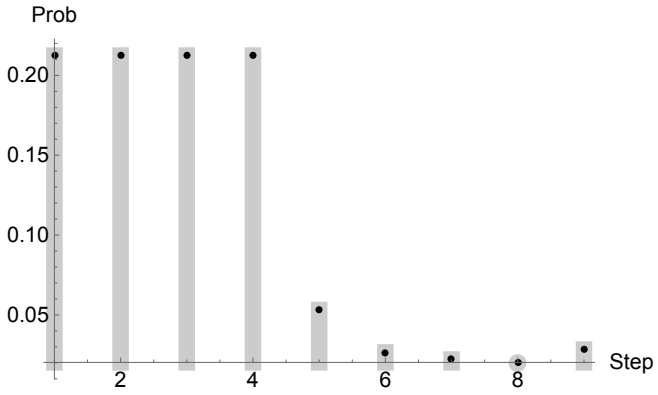

Figure 12: The Markov chain steady state distribution for evaluation B22 and defender B2.

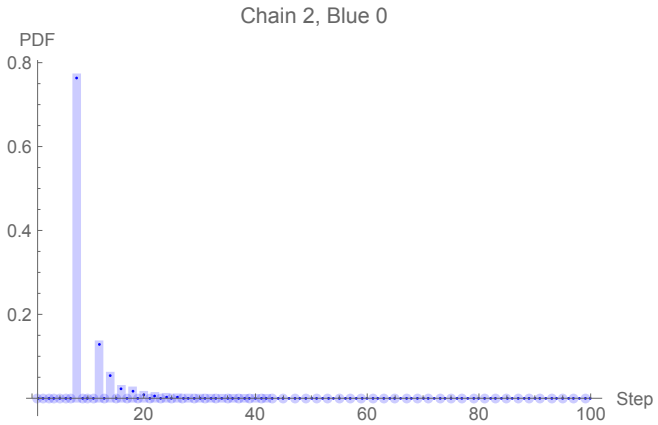

Figure 13: The time to success for evaluation B20 and defender B0.

analysis).

This result is consistent with the transition graph shown in the Figure 8.

When we look at the improved defender capabilities for defenders B1 (11) and B2 (12), we see vastly different steady state distributions. In Figure 12, it is the attacker who appears to be at a significant disadvantage.

It is also notable that while Figures 11 and 12 look similar, the attacker performance at the "Ready" state is noticeably worse in Figure 12 with about 2% of the time allocated to the "Ready" state as compared to about 5% of time at the "Ready" state in Figure 11 . This difference is because the defender B1 cannot detect the attacker at stages 7 and 8. It is also a result of weak detection capabilities at stages 7 and 8 (8% and 42% respectively) for defender B2. Whether the difference between 5% vs. 2% in "Ready" state is material would be determined by the defender preferences, cost difference between B1 and B2 capabilities, and the nature of the system the defender is protecting.

When looking at the steady state distributions, it is also important to understand whether they are operationally relevant, because the convergence time to steady state may be long. To investigate that aspect of the notional APT3 attack, we calculated the "time-to-success" distribution defined as the time of the first passage to the "Ready" state. These distributions are shown in Figures 13 - 15. The times to success shown here are somewhat notional. A key component of the time is to allow artifacts generated by the adversary behaviors to register in the necessary logs that seed defensive information analysis capabilities. The time required for information to percolate from logging to analysis can also vary widely between defenders, but we did not have data to incorporate this variability into our analysis. Despite this limitation of the analysis, we believe these distributions are still informative for defender decision making, because of substantial differences in the shape of the distributions.

In evaluation B20 (Figure 13), defender B0 is largely unable to stop the attacker progress: about 80% of the time the attacker succeeds completely unimpeded.

The improved defender B1 (14) and B2 (15) capabilities result in substantially better outcomes and time-to-success distributions that are more favorable to the defender.

We see that in the evaluation B21, the attacker succeeds unimpeded with a probability of only about 10% and in the evaluation B22, this probability further drops to about 6%.

Given appropriate timing information, the same technique for quantifying the time-to-success distribution for the attack that starts at step 1 can be applied to estimating the attacker progress as a conditional distribution given a delayed detection at a different step of the attack. The likely progress the attacker has made between an attacker's action

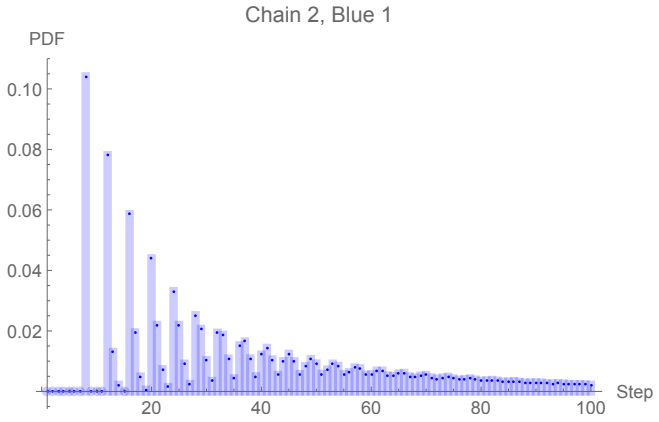

Figure 14: The time to success for the evaluation B21 and defender B1.

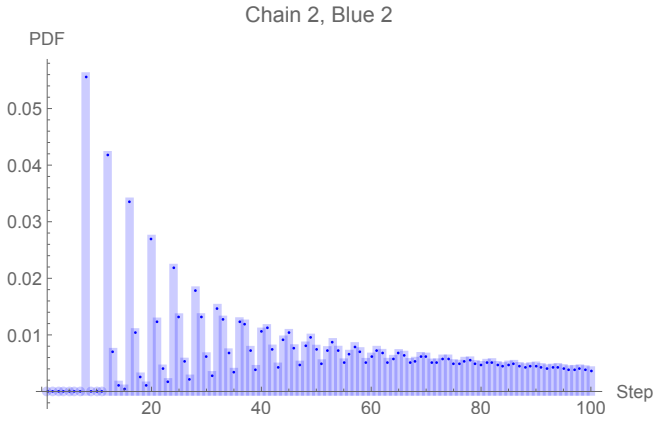

Figure 15: The time to success for the evaluation B22 and defender B2.

(reflected in logs) and detection by defender (an actionable alert) can then be estimated by applying the attack state at the action/logging time to the corresponding Markov process transition diagram. This generates the distribution of possible states at the detection time and can help direct either automated or manual response.

We also would like to highlight another, less immediately obvious implication from our analysis that emphasizes the need for defenders to optimize resource allocation between different attack steps (in real world, this would roughly correlate to investing resources in detection of a variety of TTPs from the ATT&CK framework). If we compare Evaluations B12 and B22 by looking at the Table 4, it is not immediately clear which one would give better outcomes to the defender. The set of steps with positive detection probabilities is the same for both Evaluations, but the detection probabilities themselves are different between B12 and B22. In some cases, B12 has a higher probability of detection vs. B22: in step 7 and especially in step 9 with 67% detection probability vs. 42% for evaluation B22.In other steps, e.g., 4 and 8, the detection probability for B12 is lower than for B22. This can be interpreted as B2 allocating more resources to detection on steps 7 and 9 of attack chain 1 and the same defender B2 allocating more resources to steps 4 and 8 for attack chain 2. We cannot

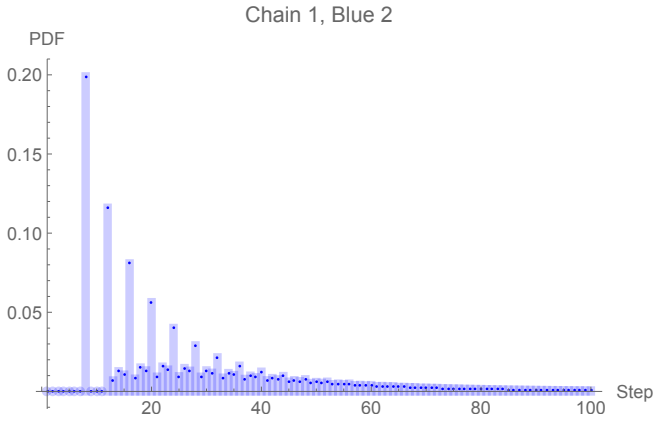

Figure 16: The time to success for the evaluation B12 and defender B2.

say a priori which allocation is better, but we can use the time-to-success distribution to investigate the differences between the outcomes defender B2 achieves against attack chains 1 and 2 represented respectively in Evaluations B12 (Figure 16) and B22 (Figure 15). We see that even though the evaluation B12 provides the defender with a much better opportunity to detect the attacker at "Ready" (step 9), this alone is insufficient to degrade the attacker success.

We see that the attacker succeeds very quickly with probability of about 20% in evaluation B12 and with probability of only about 6% in evaluation B22. This means that optimized effort allocation by defenders (B2 in this case) across different parts of their system may be more important than making a single step substantially more difficult. This result also supports the need to look at the whole defender system in the context of possible attacks and allocate the resources across multiple opportunities for attack disruption, rather than rely on a single robust defense.

We also note that more advanced defenders, B1 and B2, exhibit non-monotonic time-to-success distributions where relatively high values alternate with relatively low values, giving an appearance of a mixture of distributions. We believe this is because the probability of all possible pathways to attack success that have the same length does not need to be a monotonically decreasing function of the pathway length but leave the exploration of this phenomenon to the future work.

6 Conclusions

We developed a model and a method for explicit quantification of attack success parameters and costs based on individual attack steps difficulty and detection probabilities. Our approach started with explicit characterization of the attack success conditions and attacker and defender strategies, and it used Markov chain representation to generate attack success metrics including the "Ready" residence time, time-to-success distribution, and steady state distribution.

This allowed evaluating tradeoffs between defender actions, such as detection efforts at different parts of the system and at different steps of the attack. For a real-world defender, these differences might look like the tradeoff between investing in tools to alert on a larger variety of TTPs

or investing in training for the team analyzing the alerts to recognize more subtle signs of adversary actions. These metrics can be used to allocate resources between detection on different parts of the system and optimize the defender policy against an attack or a set of attacks. These metrics can also inform cyber or emulation-based experiments.

We used the MITRE ATT&CK® vendor Evaluations APT3 data to quantify attacks success parameters and difficulty. We showed that performance of different defender models against APT3 notional attack varied greatly and could mean the difference in the attacker always achieving their objectives vs. the attacker spending the most of their time trying to regain a foothold in the system and rarely accomplishing their objectives.

We showed that changes to the defender resource allocation and detection capabilities on different steps of the attack affected the attack success metrics, potentially in a non-intuitive fashion, and compared such metrics for defenders with different capabilities.

We showed that the defender detection expenditures corresponding to different stages of the attack or attacker TTPs have substantially different effects on the attack success parameters and different returns on defender investments. This strengthens the need to look at the whole defender system in the context of possible attacks and allocate the resources across multiple opportunities for attack disruption, rather than rely on a single robust defense.

This approach can be applied to a large set of attacks, including other attacks and attacker capabilities documented in the MITRE ATT&CK Framework. The resulting attack metrics assessment can be used to develop or improve the defender policy for a given system, or to change the system if necessary to achieve cybersecurity risk reduction or other defender objectives. In addition to MITRE ATT&CK Evaluations data, this approach can use similar data derived from cyber experimentation in emulated environments and other empirical methods. The analysis in this paper can also allow identifying the parameters whose uncertainty has the greatest effect on attack success metrics and therefore guide the emulation experiments or other data collection efforts.

REFERENCES

[1] McAfee Enterrise, *What Is Endpoint Detection and Response (EDR)?*, 2022 (Accessed January 20, 2022). [Online]. Available: "https://www.mcafee.com/enterprise/en-us/security-awareness/endpoint/what-is-endpoint-detection-and-response.html"

[2] Palo Alto Networks, *What is endpoint detection and response (EDR)?*, 2022 (Accessed January 20, 2022). [Online]. Available: "https://www.paloaltonetworks.com/cyberpedia/what-is-endpoint-detection-and-response-edr"

[3] Splunk, *What Is IT Event Correlation?*, "2022 (Accessed January 20, 2022)". [Online]. Available: "https://www.splunk.com/en_us/data-insider/it-event-correlation.html"

[4] A. V. Outkin, B. K. Eames, M. A. Galiardi, S. Walsh, E. D. Vugrin, B. Heersink, J. Hobbs, and G. D. Wyss, "GPLADD: Quantifying trust in government and commercial systems a game-theoretic approach," *ACM Trans. Priv. Secur.*, vol. 22, no. 3, pp. 18:1–18:27, Jun. 2019. [Online]. Available: http://doi.acm.org/10.1145/3326283

[5] C. R. Shelton and G. Ciardo, "Tutorial on structured continuous-time markov processes," *Journal of Artificial Intelligence Research*, vol. 51, pp. 725–778, 2014.

[6] S. Banisch, *Markov Chain Aggregation for Agent-Based Models*. Springer, 2015.

[7] X. Feng, Z. Zheng, P. Mohapatra, and D. Cansever, "A stackelberg game and markov modeling of moving target defense," in *Decision and Game Theory for Security*, S. Rass, B. An, C. Kiekintveld, F. Fang, and S. Schauer, Eds. Cham: Springer International Publishing, 2017, pp. 315–335.

[8] E. Miehling, M. Rasouli, and D. Teneketzis, "A pomdp approach to the dynamic defense of large-scale cyber networks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2490–2505, 2018.

[9] J. P. Hobert, G. L. Jones, and C. P. Robert, "Using a markov chain to construct a tractable approximation of an intractable probability distribution," *Scandinavian Journal of Statistics*, vol. 33, no. 1, pp. 37–51, 2006.

[10] A. Wollaber, J. Peñna, B. Blease, L. Shing, K. Alperin, S. Vilvovsky, P. Trepagnier, N. Wagner, and L. Leonard, "Proactive cyber situation awareness via high performance computing," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2019, pp. 1–7.

[11] N. Wagner, C. Ş. Şahin, J. Pena, and W. W. Streilein, "Automatic generation of cyber architectures optimized for security, cost, and mission performance: A nature-inspired approach," in *Advances in Nature-Inspired Computing and Applications*. Springer, 2019, pp. 1–25.

[12] L. Huang and Q. Zhu, "A dynamic games approach to proactive defense strategies against advanced persistent threats in cyber-physical systems," *Computers & Security*, vol. 89, p. 101660, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404819302020

[13] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," *MITRE Product MP*, pp. 18–0944, 2018.

[14] B. Stojanović, K. Hofer-Schmitz, and U. Kleb, "Apt datasets and attack modeling for automated detection methods: A review," *Computers & Security*, vol. 92, p. 101734, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404820300213

[15] B. E. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, D. P. Miller, C. Wampler, S. M. Whitley, and R. D. Wolf, "Finding cyber threats with att&ck-based analytics," *The MITRE Corporation, Bedford, MA, Technical Report No. MTR170202*, 2017.

[16] MITRE ATT&CK Framework, *Attack Chain Information*, 2020 (accessed July 27, 2020). [Online]. Available: https: attackevals.mitre.org/APT3/operational-flow.html

[17] B. Van Leeuwen, V. Urias, J. Eldridge, C. Villamarin, and R. Olsberg, "Cyber security analysis testbed: Combining real, emulation, and simulation," in *44th Annual 2010 IEEE International Carnahan Conference on Security Technology*. IEEE, 2010, pp. 121–126.

[18] B. P. Van Leeuwen, W. M. Stout, and V. E. Urias, "Empirical assessment of network-based moving target defense approaches," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 764–769.

[19] S. T. Jones, K. G. Gabert, and T. D. Tarman, "Evaluating emulation-based models of distributed computing systems," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2017.

[20] E. DeWaard, C. Deccio, D. J. Fritz, and T. D. Tarman, "Research directions for cyber experimentation: Workshop discussion analysis." Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2017.

[21] V. Urias and B. Van Leeuwen, "Experimental methods for control system security research," in *Cyber-security of SCADA and Other Industrial Control Systems*. Springer, 2016, pp. 253–277.

[22] T. Kenaza, K. Tabia, and S. Benferhat, "On the use of naive bayesian classifiers for detecting elementary and coordinated attacks," *Fundamenta Informaticae*, vol. 105, no. 4, pp. 435–466, 2010.

[23] K. Tabia, S. Benferhat, P. Leray, and L. Mé, "Alert correlation in intrusion detection: Combining ai-based approaches for exploiting security operators' knowledge and preferences," in *Working Notes for the 2011 IJCAI Workshop on Intelligent Security (SecArt)*, 2011, p. 42.

[24] S. Benferhat, A. Boudjelida, K. Tabia, and H. Drias, "An intrusion detection and alert correlation approach based on revising probabilistic classifiers using expert knowledge," *Applied intelligence*, vol. 38, no. 4, pp. 520–540, 2013.

[25] S. Elhag, A. Fernández, A. Bawakid, S. Alshomrani, and F. Herrera, "On the combination of genetic fuzzy systems and pairwise learn-

ing for improving detection rates on intrusion detection systems," *Expert Systems with Applications*, vol. 42, no. 1, pp. 193–202, 2015.

[26] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686–728, 2018.

[27] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.

[28] E. M Hutchins, M. J Cloppert, and R. M Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, 01 2011.

[29] K. C. Nguyen, T. Alpcan, and T. Başar, "A decentralized bayesian attack detection algorithm for network security," in *IFIP International Information Security Conference*. Springer, 2008, pp. 413–428.

[30] Q. Zhu, H. Tembine, and T. Başar, "Distributed strategic learning with application to network security," in *2011 American Control Conference on O'Farrell Street, San Francisco, CA, USA June 29 - July 01, 2011*, 2008.

[31] T. Alpcan and T. Başar, "A game theoretic approach to decision and analysis in network and intrusion detection," in *Proceedings of the 42nd IEEE Conference on Decision and Control, Hawaii USA, December, 2003*. IEEE, 2003.

[32] ——, "A game theoretic analysis of intrusion detection in access control systems," in *43rd IEEE Conference on Decision and Control, December 14-17, 2004 Atlantis, Paradise Island, Bahamas*. IEEE, 2004.

[33] D. Wilt, R. Maitzler, and J. DeVale, "Metrics for trust in integrated circuits," in *Proceedings of GOMACTech-08 Government Microcircuit Applications and Critical Technology Conference*, 2008, pp. 233–236.

[34] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using ic fingerprinting,," in *Proc. IEEE Symp. Security Privacy*, 2007, pp. 296–310.

[35] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," in *Proc. IEEE Int. Workshop Hardw.-Orient. Security Trust*, 2008, pp. 51–57.

[36] J. Zhang, F. Yuan, L. Wei, Y. Liu, and Q. Xu, "Veritrust: Verification for hardware trust," *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, vol. 34, no. 7, pp. 1148–1161, 2015.

[37] D. Pentrack, N. Levine, J. Lloyd, and A. Gahoonia, "Quantifying system trust and microeletronics integrity," in *Proceedings of GOMACTech-15 Government Microcircuit Applications and Critical Technology Conference*, 2015, pp. 211–214.

[38] J. Graf, "Trust games: How game theory can guide the development of hardware trojan detection methods," in *Proceedings of the 2016 IEEE International Symposium on Hardware Oreinted Security and Trust (HOST)*. IEEE, 2016, pp. 91–96.

[39] ——, "Toward optimal hardware trojan detection through security economics and game theory," in *Proceedings of the 41st Annual GOMACTech Conference*, 2016, pp. 187–190.

[40] C. A. Kamhoua, H. Zhao, M. Rodriguez, and K. A. Kwiat, "A game-theoretic approach for testing for hardware trojans," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 3, pp. 199–210, 2016.

[41] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Comput. Surv.*, vol. 49, no. 4, pp. 62:1–62:35, Dec. 2016. [Online]. Available: http://doi.acm.org/10.1145/3005714

[42] B. Schneier, "Attack trees," *Dr. Dobbs Journal*, December, 1999.

[43] J. P. McDermott, "Attack net penetration testing," in *Proceedings of the 2000 Workshop on New Security Paradigms*. New York: ACM Press, 2001, pp. pp.15–21.

[44] R. E. Moore and R. Linger, "Attack modeling for information security and survivability," CMU Software Engineering Institute, Tech. Rep., 2001.

[45] L. P. Swiler, C. Phillips, D. Ellis, and S. Chakerian, "Computer-attack graph generation tool," in *Proceedings of the Second DARPA Information Survivability Conference & Exposition (DISCEX II)*, vol. vol. II. Los Alamitos, California: IEEE Computer Society, 2001, pp. pp. 307–321.

[46] K. Ingols, M. Chu, R. Lippmann, S. Webster, and S. Boyer, "Modeling modern network attacks and countermeasures using attack graphs," in *Computer Security Applications Conference. ACSAC '09*, 2009.

[47] S. NOEL, S. JAJODIA, L. WANG, and A. SINGHAL, "Measuring security risk of networks using attack graphs," *International Journal of Next-Generation Computing*, vol. 1, no. 1, 2010.

[48] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using bayesian attack graphs," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, pp. 61 – 74, 03 2012.

[49] S. Jones, A. Outkin, J. Gearhart, J. Hobbs, J. Siirola, C. Phillips, S. Verzi, D. Tauritz, S. Mulder, and A. Naugle, "Evaluating moving target defense with pladd," Technical report, Sandia National Laboratories (SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2015.

[50] Y.-C. Chen, D. Campbell, V. Mooney, S. Grijalva, A. O. Brandon Eames, E. Vugrin, R. Helinski, and B. Anthony, "Power grid bad data injection attack modeling in prestige," in *GOMACTech 2019*, 2019.

[51] H. Holm, "A large-scale study of the time required to compromise a computer system," *IEEE Trans. Dependable Secur. Comput.*, vol. 11, no. 1, pp. 2–15, Jan. 2014. [Online]. Available: http://dx.doi.org/10.1109/TDSC.2013.21

[52] Z. Hu, M. Zhu, and P. Liu, "Online algorithms for adaptive cyber defense on bayesian attack graphs," in *Proceedings of the 2017 Workshop on moving target defense*, 2017, pp. 99–109.

[53] R. R. Bishop, A. V. Outkin, B. K. Eames, and C. C. White, "Increasing trust in development processes using robust, data-driven markov games: An application to prestige," *IEEE Transactions on Computational Social Systems*, 2020.

[54] E. J. Williams, J. Hinds, and A. N. Joinson, "Exploring susceptibility to phishing in the workplace," *International Journal of Human-Computer Studies*, vol. 120, pp. 1–13, 2018.

[55] R. R. Bishop, "Developing trust and managing uncertainty in partially observable sequential decision-making environments," Ph.D. dissertation, Georgia Institute of Technology, 2019.

ACKNOWLEDGMENTS

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

The authors would like to thank Laura Swiler, Eric Vugrin, Jerry Cruz, Anya Castillo, Zach Benz, Jared Gearhart, and the rest of the SECURE Grand Challenge team for valuable suggestions and comments. We would like to thank Vince Urias for valuable discussions and advice without which this paper would not have been possible. Alexander Outkin would like to thank Chip White, Reid Bishop, and Bob Cole for insightful discussions on the Partially Observed Markov Decision Processes and Games, and Rossitza Homan, Mike Livesay, and Dean Jones for valuable discussions on cybersecurity modeling, representation, and visualization, and Jacquilyn Weeks for invaluable writing advice.