Empirical Game-Theoretic Analysis for Moving Target Defense

Achintya Prakash Michael P. Wellman

Computer Science & Engineering
University of Michigan
{achintya,wellman}@umich.edu

ABSTRACT

The effectiveness of a moving target defense depends on how it is deployed through specific system operations over time, and how attackers may respond to this deployment. We define a generic cyber-defense scenario, and examine the interplay between attack and defense strategies using empirical game-theoretic techniques. In this approach, the scenario is defined procedurally by a simulator, and data derived from systematic simulation is used to induce a game model. We explore a space of 72 game instances, defined by differences in agent objectives, attack cost, and ability of the defender to detect attack actions. We observe a range of qualitative strategic behaviors, which vary in clear patterns across environmental conditions. In particular, we find that the efficacy of deterrent defense is critically sensitive to detection capability, and in the absence of perfect detection the defender is often driven to proactive moving-target actions.

1. INTRODUCTION

The hallmark of a moving-target defense (MTD) is dynamic change of system properties in order to increase uncertainty for attackers. Though techniques that can affect a moving target differ in interesting ways [16], they share the basic nature of the attack-defense interaction: attackers aiming to reveal exploitable properties of a system, while defenders simultaneously strive to change and obscure them.

Our objective is to develop models of MTD scenarios, to evaluate MTD techniques, and develop effective policies for deploying and operating them in specific contexts. To serve such ends, the model must capture the essential dynamic of *progressive* attack interacting with defense operations to impede that progression. MTD impediments do not merely block attacks, but expressly aim to push back any progress the attacker may have made to a given point. This capability provides added value exactly when the attack has an inherently progressive character [6].

MTD'15, October 12, 2015, Denver, Colorado, USA.

ACM 978-1-4503-3823-3/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2808475.2808483 .

In this paper, we examine an abstract cyber-defense scenario designed to capture the strategic interaction between progressive attacks and MTD operations. Though conceptually simple, the scenario is flexible and generic, designed to support qualitative insights applicable for a range of deployment contexts and MTD technologies. Our approach is game-theoretic, in that it seeks the characterization of attack and defense policies in strategic equilibrium. This approach allows us to see how the behaviors of rational attackers and defenders changes as we vary system configurations, agent objectives, technology characteristics, and other features. Unlike most game-theoretic studies, however, we define our model procedurally and explore it through a process of systematic simulation. This empirical approach allows us to accommodate complexity in the form of uncertainty and dynamics that render the game analytically intractable.

We find that the Nash equilibria do indeed vary qualitatively according to environmental conditions. In some environments aggressive defenses can completely deter attackers, and in others (much fewer), an aggressive attack can lead the defense to give up. In a few environments both equilibria are possible. More balanced strategic configurations also emerge as equilibria, including those where attack and defense are in intense conflict, and others where they stay more out of each others' way. Which of these balances obtains depends on the degree of contention for servers, as defined by the agents' utility functions.

Among our findings, we characterize how strategic behavior changes as a function of the defender's ability to detect attacker probes. With perfect detection, maximal defense configurations are prevalent, in which the defender effectively deters attacks with a credible threat of aggressive response. As detection capabilities degrade, the threat weakens, and attackers are more prone to engage or even flip the tables. We also observe a general shift away from defenses triggered exclusively by detected attacks, toward proactive MTDs, which modify systems in a periodic and stochastic manner, even without evidence of attacks.

The remainder of the paper proceeds as follows. Following a general discussion of background and related work (§2), we present a detailed specification of our MTD game (§3). In §4 we describe the heuristic strategies employed by attacker and defender. Our empirical game-theoretic analysis process is the subject of §5.2, and the results of that process are presented in §6. We conclude with a discussion of findings, limitations, and future work in §7.

2. BACKGROUND AND RELATED WORK

Though game-theoretic modeling has seen increasing application to problems in computer security [7, 8, 20, 21, 22], relatively little has been directly applicable to moving target defense. Notable exceptions include the works of Colbaugh and Glass [4] and Carter et al. [2], which focus on the question of how an MTD policy should choose targets for adaptation or platform migration operations.

Moving-target interactions revolve around the manipulation of uncertainty, and scenarios where such defenses are relevant also generally exhibit pervasive uncertainty over system state. One especially salient feature is whether a resource (e.g., a *server*, the canonical label for resource we use in this paper) is under the control of its owner (the *defender*), or whether it is *compromised*—under the control of an *attacker*.

Emphasizing uncertainty in state of control was a primary motivation for the *FlipIt* game [5], where two players vie for control of a single resource. In FlipIt, each player has a single action, which takes control of the resource at some cost (see top diagram of Figure 1). The complicating factor is that neither player can observe when the other has acted, and so is uncertain about the state of control except at the instant it performs its own action. Though the FlipIt model is quite abstract, it captures key elements of system security not well-supported by previous models [1]. Analysis of FlipIt led to interesting insights about the interplay of various strategy classes, and the value of aggressive play and information advances. The game has also served as a useful platform for behavioral studies of security timing policies [15].

Extensions of FlipIt have covered additional relevant scenario features. Pham and Cid [18] introduce sensing actions that reveal the control state of the server, at some cost. Laszka et al. [11] investigate a variation in which defense actions are non-stealthy. In an extension called "FlipThem", Laszka et al. [9] incorporate multiple servers, and model objectives at two extremes where an attacker needs to control one or all the servers to achieve its goal.

Our study adopts versions of the latter two extensions: (1) attackers can tell when control is wrested back by the defender, and (2) multiple servers with objectives over the number controlled. For MTD, as argued above, it is also necessary to incorporate the concept of progressive attack. This concept is illustrated schematically in Figure 1. In the top diagram (basic FlipIt), attack actions flip the binary control state. In the bottom diagram ("Progressive" FlipIt), while the actual control feature (color) is still binary, there is another dimension of state representing an attacker's progress toward compromise.

The model of Laszka et al. [10] captures attack progression in part, by defining the attack action as compromising a server after a stochastic amount of time. Defense actions taken during that time neutralize the attack. In our approach, attack actions may succeed immediately, and if they fail they nevertheless make progress in the sense of increasing the success probability of the next attack action.

The present investigation builds on a preliminary study reported last year [23], substantially extending and superseding that work. The key changes are:

1. The scenario instances include more servers (10 vs. 3).
2. A more flexible utility model (§3.3), allowing tradeoffs between objectives of control and availability.

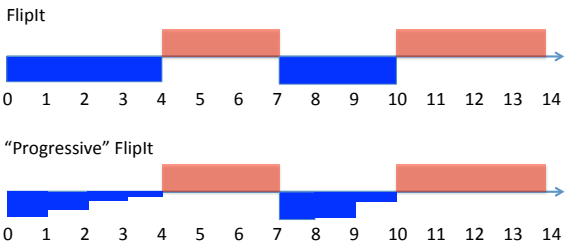

Figure 1: Schematic view of FlipIt (top), and a progressive version (bottom). Blue rectangles (below timeline) indicate server in control of the defender, and red (above line) in control of attacker. In FlipIt, an attacker action at time 4 transfers control to attacker; the defender regains control at time 7, loses it again at 10. Actions taken while one has control (without knowing it) have no effect. In the progressive version, an attack (here taken every time period) may fail to gain control, but leaves the server in a state more ripe for subsequent attack actions.

3. Imperfect probe detection, where the defender only probabilistically observes attacker probe actions.
4. A much richer set of attack and defense strategies, incorporating assessment of overall system state.

Our results here are consistent with those found previously, but go well beyond in scope and strength of evidence.

3. GAME SPECIFICATION

The attacker and defender in our game compete for the control of M servers. In the instances investigated here, we take $M = 10$.[1] Servers start out in control of the defender. The attacker may attempt to wrest control of a server through a probe action, which succeeds with some probability and otherwise increases the success probability of subsequent probes. The defender may at any time reimage a server, which takes the server down for a time, after which it comes up in the control of the defender with any attacker progress erased.

The scenario runs for a finite horizon of T time units. We set $T = 1000$ for the present study.

3.1 States and Actions

At any point of time, the state of a server is described by whether it is up or down, who controls it, and if under the defender's control how much progress the attacker has made toward taking control. Formally, server state is a triple $\langle \chi, \upsilon, \rho \rangle$, where:

- $\chi \in \{att, def\}$ represents the player who controls the server;
- $\upsilon \in \{up\} \cup [0, T]$ represents whether the server is up ($\upsilon = up$), or, is still down from a reimage initiated at time $\upsilon \in [0, T]$; and

[1]The environment simulation scales linearly in M, as do all agent strategies considered with the exception of Control-Target (see §4). That strategy would need to be revised through approximation to accommodate much greater M.

- ρ represents the number of attacker probes since the last defender reimage action.

The state of the overall system is defined by the joint state of the servers, plus the current clock time $t \in [0, T]$.

Each player has one available action, which it can choose to execute at any time on a specified server. The action is atomic and instantaneous, with effect described in terms of an associated state transition.

The attacker action is called *probe*. Probing a server has the effect of compromising it with some probability, depending on the extent of probing to that point. To describe the action precisely, let $\langle \chi_t, \upsilon_t, \rho_t \rangle$ be the state at time t, when a probe action is executed. We denote the state immediately following the probe by $\langle \chi_{t+}, \upsilon_{t+}, \rho_{t+} \rangle$. We specify the probe action's effects by the following rules:

- If $\upsilon_t \neq up$, the probe has no effect: $\langle \chi_{t+}, \upsilon_{t+}, \rho_{t+} \rangle = \langle \chi_t, \upsilon_t, \rho_t \rangle$.
- If $\upsilon_t = up$, the number of probes is incremented: $\rho_{t+} = \rho_t + 1$.
- If $\upsilon_t = up$ and $\chi_t = att$, the attacker maintains control: $\chi_{t+} = att$.
- If $\upsilon_t = up$ and $\chi_t = def$, the attacker takes control with probability $1 - e^{-\alpha(\rho_t+1)}$, where $\alpha > 0$ is an environmental factor representing the information value of probes. That is, with aforementioned probability $\chi_{t+} = att$, and with remaining probability $e^{-\alpha(\rho_t+1)}$, $\chi_{t+} = def$.

The defender action is called *reimage*. The purpose of reimaging a server is to reset its state, so that if compromised it reverts to defender control, and if not compromised the cumulative effect of probes is erased. As for the attacker's action, we define the effect of reimage in terms of state transition rules. Suppose the defender executes a reimage at time t.

- If $\upsilon_t \neq up$, the reimage has no effect: $\langle \chi_{t+}, \upsilon_{t+}, \rho_{t+} \rangle = \langle \chi_t, \upsilon_t, \rho_t \rangle$.
- If $\upsilon_t = up$, the state is reset as follows: $\langle \chi_{t+}, \upsilon_{t+}, \rho_{t+} \rangle = \langle def, t, 0 \rangle$.

We model the reimaging duration by taking the server down for a specified time interval Δ. (In our environment instances, $\Delta = 7$.) If a reimage resets a server's state at time t, then the server comes back up Δ time units later. That is, we have $\upsilon_{t'} = t$ for $t' \in [t+, t+\Delta)$, followed by an update to the state variable $\upsilon_{t+\Delta} = up$. Aside from this one exception, all state changes in our scenario are the immediate effects of player actions.

3.2 Observation Model

The two players have very different observation capabilities. The defender has an (imperfect) ability to detect probes executed on any server, but is unaware of which probes succeed in compromising their targets. The attacker is aware of which probes succeeded, and when the defender retakes a compromised server through reimaging. To state this more precisely, we specify conditions on action-generated state transitions that the players observe.

Following a probe action on a server:

- The attacker perfectly observes the server state at $t+$.
- If $\upsilon_t = up$, the defender detects the probe with probability $1 - \nu$. If $\nu = 0$, we say the defender has *perfect probe detection*, and can therefore infer ρ_{t+}.

Following a reimage action on a server:

- The attacker detects the reimage if and only if (iff) it loses control of that server due to the reimage, that is, iff $\chi_t = att$ and $\chi_{t+} = def$. In that case, it observes the full state of the server at $t+$.
- The defender perfectly observes the server state at $t+$.

Note that the attacker always knows the control state χ, but can only imperfectly track ρ between actions. The reason is that a defender in control of a server may reset the number of probes with a reimage, and the attacker does not find out about this until its next probe on that specific server. The defender has evidence about ρ, but except right after a reimage has no direct information about χ.

3.3 Utility

Each player accrues utility based on the number of servers *up* and in their control and the number of servers that are *down*. We define the rate of utility accrual as the weighted sum of two logistic functions.

Let $f(x; \theta) : [0, 1] \to [0, 1]$ denote the logistic (sigmoid) function with parameters $\theta = (\theta_{sl}, \theta_{th})$:

$$f(x; \theta) = \frac{1}{1 + e^{-\theta_{sl}(x - \theta_{th})}}.$$

Parameter θ_{sl} controls the slope of the sigmoid curve at its steepest point, and θ_{th} the threshold where this steep point occurs. The logistic is monotone, with $f(\theta_{th}; (\theta_{sl}, \theta_{th})) = 1/2$. Intuitively, the threshold θ_{th} represents the fraction of servers the agent needs to have in a desirable state in order to receive the utility in any instant.[2] For this study we fix $\theta_{sl} = 5$ for both players in all environments.

A player's utility is governed by two of these threshold functions: one function for the fraction of servers up and in the player's control, and another for the fraction of servers *not* in the *other* player's control (i.e., either up and in the player's control, or not up). Let n_c^i denote the number of servers controlled by i that are up, and let n_d denote the number of servers that are down (not up). The expression $u^i(n_c^i, n_d)$ denotes the rate of utility accrual for player i, defined by

$$u^i(n_c^i, n_d) = w^i f(n_c^i/M; \theta^{i,1}) + (1 - w^i) f((n_c^i + n_d)/M; \theta^{i,2}), \quad (1)$$

where w^i is the weighting applied by player i, and $\theta^{i,1}$ and $\theta^{i,2}$ its utility parameters for the two threshold functions.

An agent's overall utility is the sum of utility it accrues over time. For example, if player i controls n_c^i servers while n_d servers are *down* for $T/2$ time units, and then a server under its control goes *down* for the remaining $T/2$ time units, its utility would be $(T/2)u^i(n_c^i, n_d) + (T/2)u^i(n_c^i - 1, n_d + 1)$.

This formulation allows us to express a variety of preference patterns, including tradeoffs between natural objectives of an attacker or defender. For example, the *confidentiality*

[2]The extreme settings of $\theta_{th} = 1$ and $\theta_{th} = 1/M$ correspond to smoothed versions of the AND and OR models, respectively, defined for FlipThem by Laszka et al. [9].

objective from the classic "CIA triad" [17] can be interpreted as a defender's strong aversion to allowing the attacker to control servers. Also from the defender's perspective, *availability* is established when an adequate fraction of servers are in its control and not *down*. A weighting of $w^{def} = 0$ expresses exclusive concern with confidentiality, and $w^{def} = 1$ indicates a complete focus on availability.[3] We define the attacker utility in an analogous way. An attacker that accrues utility only by having servers in its control is termed a *control* attacker, whereas an attacker that accrues utility by having servers in its control or *down* is termed a *disrupting* attacker. The four combinations of extreme weightings are listed and labeled in Table 1.

Utility Env	w^{att}	w^{def}
control/avail	1	1
control/confid	1	0
disrupt/avail	0	1
disrupt/confid	0	0

Table 1: Utility environments expressing different player objectives.

The threshold parameters govern the level of contention for servers in the associated environment. For example, by thresholding f at $1/2$ we impose the constraint that significant utility is accrued only if at least a majority are in control, or are in control and *down*. In our exploration, we considered three different settings of these thresholds (as a fraction of M), and focus attention on cases where both attacker and defender have the same thresholds. Our settings with associated labels are shown in Table 2.

Setting	$\theta_{th}^{att,1}$	$\theta_{th}^{att,2}$	$\theta_{th}^{def,1}$	$\theta_{th}^{def,2}$
low	0.2	0.2	0.2	0.2
majority	0.5	0.5	0.5	0.5
high	0.8	0.8	0.8	0.8

Table 2: Combinations of threshold values used for utility functions.

In addition, our model imposes a cost for executing actions. The attacker pays a cost of $c_A > 0$ per probe. In our study we examine $c_A \in \{0.2, 0.5, 1\}$. The cost of the defender action is expressed implicitly in the utility function as the difference in utility accrued by servers being *down* as opposed to in the defender's control.

In the best case, a player accrues one utility unit per time period for keeping servers in their desired state, at no cost. The maximum overall utility for a game run is therefore T. The minimum is not defined, as players may take unlimited costly actions without achieving their objective.

4. HEURISTIC STRATEGIES

A *strategy* for the attacker or defender is a policy by which the player chooses when to execute its actions on what servers, as a function of its observation history and the current time. Even with a single action type, the space of available strategies is vast, owing to the combinatorial explosion of possible histories. Rather than explore the strategy space directly, we therefore focus on parameterized families of heuristic strategies, defined by regular structures and patterns of behavior over time. We define a restricted game over a selected set of such strategies, and systematically refine this set through an iterative process of strategy exploration and empirical game analysis.

Our heuristic strategies generate actions based on the passage of time, or observed events in the system. The former we label *periodic*. Periodic strategies are defined by:

- a period P;
- whether actions are generated deterministically every P time steps, or probabilistically according to a *renewal process* [19], here a Poisson process, with inter-arrival time distributed exponentially with mean P;
- the criteria by which they choose which server to perform the action upon.

Strategies triggered by observed events may apply actions to servers based on observations of that server, or a combination of observations across servers.

Our strategy implementations interact with a discrete-event simulation of the environment. Whenever a player's knowledge state changes (see §3.2), the player strategy is queried for the time of its next action. Depending on the strategy, the player may choose to retain its pending (previously scheduled) time of action, or to replace it on the queue with some other time based on its latest knowledge. Our periodic strategies, however, do not reconsider the specified time of an action once it has been put onto the queue, irrespective of observed changes in the interim time period. At the time of action, the player makes an action decision (which server to probe or reimage) based on its current knowledge state. The environment simulator is driven by the scheduling queue, continually processing the next scheduled player action or environment event (i.e., server transition to *up*), according to time precedence. Among events scheduled for the same time, ties are broken randomly.

4.1 Attacker Strategies

We consider two forms of periodic attacker strategy defined by the following heuristic selection strategies.

- *Uniform-Uncompromised.* Selects uniformly at random among those servers under the defender's control ($\chi_t = def$).
- *MaxProbe-Uncompromised.* Selects the server that has been probed the most since last reimage (that the attacker knows about), among those servers under the defender's control, breaking ties uniformly.

We also include one non-periodic attacker strategy that generates probe actions based on the number of servers that an attacker controls.

- *Control-Threshold.* If the attacker controls less than the threshold τ fraction of the M servers, it chooses to probe the server that has been probed the most since last reimage (as far as it is aware) yet it does not

[3] As we see below, the extreme $w^{def} = 1$ setting leads to trivial solutions, as defenders can keep their machines down to avoid compromise. The CIA *integrity* criterion has no obvious interpretation definable in terms of this utility model.

currently control. Ties are broken uniformly among all eligible servers. A minimum waiting time of one time unit separates any two consecutive actions.

In addition, we consider the *No-Op* strategy, in which the attacker never takes any action.

4.2 Defender Strategies

Among the periodic strategies we explored two different criteria for server selection:

- *Uniform.* Selects uniformly at random among all up servers ($v_t = up$).
- *MaxProbe.* Selects the server that has been probed most since its last reimage, breaking ties uniformly.

The other class of strategies triggers a reimage operation based on probe activity or inactivity. We explored two different types of strategies in this class:

- *ProbeCount-or-Period* (PCP). Reimages a server whenever it detects that it has been probed more than π times since the last reimage, *or* if it has been probed at least once but not within the last P time units. The rationale for reimaging a server that is not being probed is that this could be an indication that the attacker has already compromised it and thus ceased attack.
- *Control-Threshold.* Analogous to the attacker's strategy by the same name, we include a defender strategy that performs a reimage action when the fraction of servers controlled falls below a threshold τ. Unlike the attacker, however, the defender cannot directly observe control state. Instead, the defender estimates the number compromised based on the probes it has observed since reimaging each server. Specifically, given $\hat{\rho}$ observed probes, its estimate of the expected number of servers still controlled assumes that the first $\hat{\rho} - 1$ probes failed and the last succeeded with probability $1 - e^{-\alpha(\hat{\rho}+1)}$. Using this estimate, if
$$\frac{\mathbb{E}\left[n_c^{def}\right]}{M} < \tau,$$
reimage the server with greatest observed $\hat{\rho}$. Ties are broken uniformly. A minimum of one time unit separates any two consecutive actions.
- *Control-Target.* This strategy is like the former, except that parameter τ is interpreted as a target rather than a threshold. Specifically, if[4]
$$\Pr\left(\frac{n_c^{def}}{M} = \tau\right) > 0.001,$$
reimage the server with greatest observed $\hat{\rho}$.

Finally, the defender also has the null strategy *No-Op*.

5. EMPIRICAL GAME-THEORETIC ANALYSIS

Given the scenario setup and a repertoire of heuristic strategies for attacker and defender, we analyze the strategic scenario by an approach called *empirical game-theoretic analysis* (EGTA). The EGTA approach combines simulation with game-theoretic reasoning. For each environment studied, we evaluate the interaction between all pairs of attacker and defender strategies through repeated simulation. Outcomes from these simulations are treated as sample payoffs, and we estimate expected payoffs by the sample average. The estimated payoff function constitutes a game model over the heuristic strategy set, and we analyze this game model to characterize strategy profiles that are in game-theoretic equilibrium.

5.1 Simulation Environments

Our scenario specification includes several configurable parameters, described in §3. As noted above, the scenario instances studied here take $M = 10$, $T = 1000$, $\alpha = 0.05$, and $\Delta = 7$.

As mentioned in §3.3, we employ four different settings of utility weights (Table 1) and three different settings of the utility thresholds (Table 2). This yields twelve different joint utility models. As we see below, these are not equally interesting, as some lead to degenerate strategic equilibria or represent unrealistic objectives.

We explore each combination of these under perfect and imperfect probe detection. With perfect probe detection ($\nu = 0$), we investigate three different settings of the probe cost $c_A \in \{0.2, 0.5, 1\}$. For imperfect probe detection, we consider three different probe miss rates, $\nu \in \{0.3, 0.5, 0.8\}$, all with probe cost fixed at $c_A = 0.5$. In total, we examine 72 parametrically distinct instances of our MTD game.

We implement the scenario using a discrete-event simulator. The simulator maintains state as described in §3.1. It manages a queue of scheduled actions and state transitions, repeatedly processing the next element of the queue. Whenever a state transition includes something observable by an agent, that agent is notified, and based on the strategy may also lead to insertion of further actions on the queue (see §4).

Table 3 lists the strategy instances that we included in our evaluation. For each attacker (Att) or defender (Def) heuristic, we specify the parameter values covered. We instantiated PCP defender strategies for all combinations of parameters π and P listed, except that for $\pi = 1$ the period is irrelevant so only one instance was included. The strategy *Max-Renewal* refers to the stochastic periodic strategy, acting on the server with the maximum probe count. Altogether, we included twelve attacker and twenty defender strategy instances.

Att/Def	**Heuristic**	P	π	τ
Att	Uniform	.1,1	—	—
Att	Max	.1,1	—	—
Att	Control	—	—	.1,.2,.3,.5,.8
Def	Max	1,7	—	—
Def	PCP	1,50	1,2,4	—
Def	Control-Th	—	—	.01,.1,.3,.5,.8
Def	Control-Ta	—	—	.1,.2,.3,.5,.8
Att/Def	Max-Renewal	1,10	—	—
Att/Def	No-Op	—	—	—

Table 3: Strategy instances included in our EGTA study.

[4]This strategy requires that τM be integral.

This strategy set constitutes the base set of strategy employed across environments. The parameters were set heuristically and informed by exploration in our preliminary study [23].

5.2 Game-Theoretic Analysis Process

Our EGTA process followed the steps displayed in Figure 2. For each environment, we ran simulations of all $12 \times 20 = 240$ strategy profiles. Simulations were conducted on a large-scale computational cluster, using the EGTAOnline system [3] to manage the simulation jobs and organizing the resulting data. Each profile was run at least 600 times (often many more), and for each profile we take the sample-average payoff for attacker and defender as the payoff vector in the estimated normal-form game.

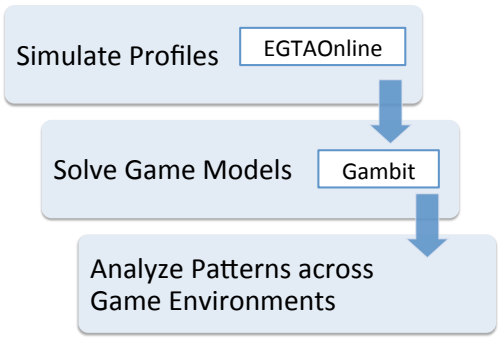

Figure 2: Empirical game-theoretic analysis pipeline.

Once we have a normal-form game model for a specified scenario, we proceed to analyze it using standard game-theoretic algorithms. We calculate Nash equilibria using Gambit [14], a general tool for game-theoretic computation. For all the games that we explored, Gambit was able to identify at least one Nash equilibrium.[5] In some cases there were over one hundred equilibria, including many similar to each other and some qualitatively different.

The final step is to examine the equilibria found, and construct qualitative classes capturing their essential features. Patterns in how qualitative equilibria vary with environment characteristics yield strategic insights about moving target defense in our scenario.

6. EQUILIBRIUM RESULTS

In this section, we report the equilibria found in our collection of game instances, classified into four major qualitatively distinct categories. The main results are summarized in Tables 4 (perfect probe detection) and 5 (imperfect).

6.1 Perfect Probe Detection Environments

We first examine environments where the defender is able to perfectly observe all incoming probes on any of the servers. As described in §5.1, we have a total of 36 different environments with this setting. We identify several qualitatively different equilibria that manifest across these environments. In this section, we describe the qualitative equilibria and explain why they apply in the respective environments. Table 4 lists the environments and their qualitative equilibria.

6.1.1 *Maximal Defense*

In the *maximal defense* equilibria, termed *MaxDef*, the defender responds to probing activity with aggressive reimaging, to the point that the attacker cannot achieve any worthwhile amount of compromise, and in consequence simply gives up and plays No-Op. In this equilibrium the attacker accrues negligible utility, and the defender achieves essentially the maximum possible.

We define two special cases of MaxDef, differing on the nature of the aggressive defense. In *MaxDef-P*, the defender frequently reimages according to a period strategy, for example the periodic strategy MaxProbe with $P = 1$. With the maximally probed server reimaged every time period, the attacker cannot get more than one probe in to any server. Even if the attacker's probe were successful, it would not maintain control for more than a unit time period. For the range of probe success probability, probe costs, and utility thresholds considered, this does not provide expected value worth the cost of the probe. The best response for the attacker is therefore No-Op.

In order for MaxDef-P to be in equilibrium, we also need that frequent reimaging is a defender best response to a No-Op attacker. This is indeed the case for defenders with extreme confid objectives, as such defenders are equally happy for a server to be down or up, as long as they are not compromised. In such settings, it is trivial for a defender to achieve its objective just by continually reimaging. For this reason, the environments with confid objectives are strategically straightforward, hence we focus most of our analysis on the defender with avail objectives. An avail defender gets no utility contribution from servers that are down, so frequent periodic reimaging is not a best response to No-Op attackers.

The second type of maximal defense is more interesting. In *MaxDef-T*, the defender's strategy does not reimage based on a time period, but is specifically *triggered* by detected probe activity. For example, the PCP defense strategy reimages based on a probe threshold (or time without probe, but only if there has been at least one probe). Likewise the Control strategies reimage based on estimates of servers in control, which are in turn derived from the number of probes the defender has detected.

Against a No-Op attacker, trigger defenses never actually have to reimage, and so achieve the greatest possible utility (continual control of all servers, no reimaging cost). Thus, they are clearly best responses to No-Op. If the trigger is sufficiently aggressive (e.g., PCP with a threshold of one probe), then an attacker cannot gain and maintain control of more than a few servers for any significant time. As we see in Table 4, MaxDef-T is indeed an equilibrium of all the control/avail environments.

Perhaps surprisingly, the MaxDef-T equilibrium also appears in some of the disrupt/avail environments. Since the disrupt attacker is happy when servers are down, aggressive reimaging to some extent plays to their objective. However, a reimaged server is down for only $\Delta = 7$ time units, after which it is up and in the defender's control. So if the probe cost is sufficiently great (or high utility threshold at

[5]The Gambit package offers multiple solution algorithms for two-player, non-zero sum games. Most of our results were generated by an algorithm based on extreme point enumeration [13]. For a couple of games this did not converge, and we had to employ an alternative algorithm based on linear complementarity problems [12].

		Utility Environments			
Utility threshold	c_A	disrupt/avail	disrupt/confid	control/avail	control/confid
low	1	MaxDef-T, Share	*MaxDef-P*	MaxDef-T, Share	*MaxDef-P*
low	0.5	Share	*MaxDef-P*	MaxDef-T	*MaxDef-P*
low	0.2	Share	*MaxDef-P*	MaxDef-T, Share	*MaxDef-P*
majority	1	MaxDef-T	MaxDef-P	MaxDef-T	MaxDef-P
majority	0.5	Fight	MaxDef-P	MaxDef-T	*MaxDef-P*
majority	0.2	Fight	MaxDef-P	MaxDef-T, MaxAtt	*MaxDef-P*
high	1	MaxDef-T, MaxAtt	*MaxDef-P*	MaxDef-T, MaxAtt, Fight	*MaxDef-P*
high	0.5	MaxDef-T, MaxAtt, Fight	*MaxDef-P*	MaxDef-T, MaxAtt, Fight	*MaxDef-P*
high	0.2	MaxDef-T, MaxAtt	*MaxDef-P*	MaxDef-T, MaxAtt, Fight	*MaxDef-P*

Table 4: Qualitative Nash equilibria for the thirty-six perfect probe detection environments. Cells in italics indicate games not actually simulated, but with obvious equilibria given the confid defense objective.

any probe cost), then No-Op may still be a best response to aggressive defense.

6.1.2 Maximal Attack

The *maximal attack* (MaxAtt) equilibrium is the dual of MaxDef, where in response to an aggressive attack the defender can do no better than give up and play No-Op. Under a strict definition, MaxAtt is a difficult equilibrium to sustain in our setting, since it does not cost a defender anything to reimage a server that will be out of its control otherwise. We therefore also classify as MaxAtt equilibria where the defender performs occasional reimages, but without obtaining significant utility or denying much utility to the attacker.[6] We do find both the strict and non-strict versions of MaxAtt (though never exclusively) in several environments in Table 4, including all those with high utility threshold. With such strong contention for servers, a defender is generally unable to achieve much utility in the face of an aggressive attack.

6.1.3 Fight

In the *Fight* equilibria, both players perform their actions frequently, in a struggle to achieve their control or disruption objectives. Neither consistently succeeds, and the combined utility is below (often well below) the level of even one player attaining its objective.

For example, in the disrupt/avail environment with majority utility threshold and $c_A = 0.2$, the sole equilibrium has the attacker and defender playing periodic strategies with probability at least 0.8. In that configuration, neither keeps the majority of servers in their preferred state consistently, but both do accrue some utility by these actions.

Fight equilibria also arise in all the control/avail environments with high utility threshold, and other cases where there is significant contention and a possibility that both attack and defense actions are worthwhile.

6.1.4 Share

Our final category of equilibrium is called *Share*, because the strategies chosen allow both players to achieve their objectives, and thus obtain a total utility well above the maximum for one player. For example, in the disrupt/avail environment with low utility threshold, and $c_A \in \{0.5, 0.2\}$, there is an equilibrium with the attacker probing periodically and the defender specifying a control target. In this configuration, both are able to maintain their fraction of servers (only 0.2) in desired states, and are content to refrain from more aggressive attacks and defenses.

Given the low contention when both agents have low utility threshold, each agent's objective is easy to achieve. In particular, if the attacker applies a Control strategy with a threshold fraction just above its utility threshold, the defender will keep control of a satisfactory number of servers without doing anything. Thus, we also see Share equilibria with control attackers and No-Op defenders.

6.2 Imperfect Probe Detection Environments

We also explored environments where the defender is not able to detect all the probes. In these environments, we fixed $c_A = 0.5$ and we varied the miss rate $\nu \in \{0.3, 0.5, 0.8\}$. As for the perfect probe detection environments, we explore all combinations of utility thresholds (Table 2) and weightings (Table 1), yielding a total of thirty-six different settings. We present the qualitative equilibria for these settings, along with the corresponding perfect detection ($\nu = 0$) cases for comparison, in Table 5.

Our first observation is that it is much more difficult to sustain MaxDef-T with imperfect probe detection. The MaxDef-P equilibria in confid environments persist, as in MaxDef-P the defender reimages unconditionally, and so does not depend on probe detection. In MaxDef-T the reimages are triggered by detected probes, and so with a positive miss rate the attacker can expect to get some probes in without a consequent reimage. As Table 5 indicates, we find no MaxDef-T equilibria for the imperfect detection environments.

We do find one control/avail environment where MaxDef is in equilibrium, and interestingly this is neither a periodic (-P) or triggered (-T) MaxDef, but a case where the defender mixes between such strategies. This makes sense, as to achieve sufficient deterrent with imperfect probe detection the defender may need to reimage at least probabilistically without detecting a probe.

Another noteworthy difference with imperfect probe detection is that we now have games where MaxAtt is the only qualitative equilibrium. As one would expect, MaxAtt

[6]We similarly stretch the definition of MaxDef to include weak-but-not-exactly-No-Op attacks, however the vast majority of MaxDef equilibria observed are constituted of actual No-Op attackers.

		Utility Environments			
Utility threshold	ν	disrupt/avail	disrupt/confid	control/avail	control/confid
low	0	Share	*MaxDef-P*	MaxDef-T	*MaxDef-P*
low	0.3	Share	*MaxDef-P*	MaxDef, Share	*MaxDef-P*
low	0.5	Share	*MaxDef-P*	Share	*MaxDef-P*
low	0.8	Share	*MaxDef-P*	Share	*MaxDef-P*
majority	0	Fight	MaxDef-P	MaxDef-T	*MaxDef-P*
majority	0.3	Fight	*MaxDef-P*	Fight	MaxDef-P
majority	0.5	Fight	*MaxDef-P*	Fight	MaxDef-P
majority	0.8	MaxAtt	*MaxDef-P*	MaxAtt, Fight	MaxDef-P
high	0	MaxDef-T, MaxAtt, Fight	*MaxDef-P*	MaxDef-T, MaxAtt, Fight	*MaxDef-P*
high	0.3	MaxAtt	*MaxDef-P*	Fight	*MaxDef-P*
high	0.5	Fight	*MaxDef-P*	MaxAtt	*MaxDef-P*
high	0.8	MaxAtt	*MaxDef-P*	MaxAtt	*MaxDef-P*

Table 5: Qualitative Nash equilibria for forty-eight environments, including thirty-six with imperfect probe detection.

is generally more prevalent with increasing ν. Clearly the ability to reliably detect probes is critical for the defender to maintain an edge over the attacker.

The distribution of Fight and Share equilibria is similar to that seen in perfect probe detection environments. One trend we have noticed is that as ν is increased, the Fight equilibria vary in character, relying more on periodic relative to triggered defender strategies. A natural consequence of degraded probe detection is that the defender Control strategies are less accurate in assessing the system state, and so a less effective guide for when to reimage.

7. DISCUSSION

The game-theoretic solutions produced by our EGTA process (Figure 2) produce qualitative patterns of behavior that accord intuitively with strategic properties of the range of environments studied. In retrospect, it could have been possible to identify some of these without the extensive simulation and game-theoretic reasoning process undertaken. In our experience, however, it often takes some concrete simulation to expose the obvious in complex environments. In any case, the simulations serve to confirm the reasoning given, and the fact that sensible strategy profiles emerged from the search serves as validation of the overall approach. Moreover, having considered a diverse set of alternative strategies provides information about plausible heuristics that turn out not to be part of equilibrium solutions.

In addition to the general patterns of qualitative equilibria, we were particularly struck by several findings from this investigation.

- With perfect probe detection, maximal defense is always an equilibrium when attackers have control objectives, and pervasive as well for disrupt objectives. However it disappears as an option in all but one of the imperfect probe detection environments.

- Maximal attack is occasionally in equilibrium among others with perfect probe detection, but becomes significantly more prevalent once probe detection degrades.

- Fight equilibria are generally pervasive, except when contention for servers is particularly weak (low utility threshold).

- The Control strategies appear widely in equilibrium configurations.

The first two of these reinforce the importance of reliable probe detection for effective deterrent. The third and fourth demonstrate the payoff to sophisticated policies in contentious environments, and suggest that further strategy exploration may produce new equilibria.

As indicated above, the extreme confid environments are of limited interest, as defenders have a trivial strategy. Similarly, environments with low utility threshold allow both players to satisfy their objectives (i.e., play Share equilibria), and so do not push any limits of MTD strategy. Including these environments does provide a sanity check, however. Less extreme versions (i.e., weighted combinations of confid and avail, or environments where only one player has low utility threshold) may prove more interesting environments to analyze.

Of course, any conclusions drawn from an EGTA study must be qualified by limitations inherent in the simulation-based construction of our game models. First, it is not possible to rule out additional equilibria beyond the strategy sets considered here. The strategies we implemented include many obvious candidates (e.g., the periodic and renewal strategies resemble similar strategies analyzed in studies of FlipIt), but omit many others (e.g., forms of non-stationary strategies also considered in FlipIt analyses). It would also be valuable to include strategies that are more sophisticated in their adaptation to experience. Such strategies, for example, could modulate their aggressiveness based on the observed behavior of the other player.

For the environments with multiple qualitatively distinct equilibria, our analysis has nothing to say about selection among these. For example, where both MaxDef and MaxAtt are in equilibrium, which would prevail depends on the relative fortitude of the attacker and defender. More technically, we would ask which player can more credibly threaten its maximalist policy. Such questions could be addressed

through a more extensive-form (dynamic) analysis, for example by explicitly considering multiple stages of decision and adopting equilibrium refinement based on perfection. Alternatively, we could consider Stackelberg models, where one player or the other is presumed to have commitment power based on the scenario setup.

The imperfect probe detection modeled here is limited to false negatives, where actual probes go undetected. It would also be interesting to explore the possibility of false positives, where the defender incorrectly classifies non-probes as probes. This too would be expected to undermine maximal defenses, as the false positives would impose large costs.

There are many other natural directions for further exploration. One is design of defender strategies to deal better with unreliable probe detection. Other avenues for increasing sophistication in attacker and defender policies would be worth exploring, including intent inference, and explicit reasoning about threats and counter-threats. In ongoing work, we also plan to explore environments with a range of probe efficacy (e.g., settings of α), stochastic downtimes, and utility models intermediate between the extreme points considered here.

References

[1] Bowers, K.D., van Dijk, M., Griffin, R., Juels, A., Oprea, A., Rivest, R.L., Triandopoulos, N.: Defending against the unknown enemy: Applying FlipIt to system security. In: 3rd International Conference on Decision and Game Theory for Security. LNCS, vol. 2638, pp. 248–263. Springer (2012)

[2] Carter, K.M., Riordan, J.F., Okhravi, H.: A game theoretic approach to strategy determination for dynamic platform defenses. In: First ACM Workshop on Moving Target Defense. pp. 21–30 (2014)

[3] Cassell, B.A., Wellman, M.P.: EGTAOnline: An experiment manager for simulation-based game studies. In: Multi-Agent Based Simulation XIII, LNAI, vol. 7838. Springer (2013)

[4] Colbaugh, R., Glass, K.: Predictability-oriented defense against adaptive adversaries. In: IEEE International Conference on Systems, Man, and Cybernetics. pp. 2721–2727 (2012)

[5] van Dijk, M., Juels, A., Oprea, A., Rivest, R.L.: FlipIt: The game of "stealthy takeover". Journal of Cryptology 26, 655–713 (2013)

[6] Evans, D., Nguyen-Tuong, A., Knight, J.: Effectiveness of moving target defenses. In: Jajodia, S., Ghosh, A.K., Swarup, V., Wang, C., Wang, X.S. (eds.) Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats. Springer (2011)

[7] Grossklags, J., Christin, N., Chuang, J.: Secure or insure?: A game-theoretic analysis of information security games. In: Seventeenth International Conference on World Wide Web. pp. 209–218 (2008)

[8] Kiekintveld, C., Lisý, V., Píbil, R.: Game-theoretic foundations for the strategic use of honeypots in network security. In: Jajodia, S., Shakarian, P., Subramanian, V.S., Swarup, V., Wang, C. (eds.) Cyber Warfare: Building the Scientific Foundation. Springer (2015)

[9] Laszka, A., Horvath, G., Felegyhazi, M., Buttyán, L.: FlipThem: Modeling targeted attacks with FlipIt for multiple resources. In: 5th International Conference on Decision and Game Theory for Security. LNCS, vol. 8840, pp. 175–194. Springer (2014)

[10] Laszka, A., Johnson, B., Grossklags, J.: Mitigating covert compromises: A game-theoretic model of targeted and non-targeted covert attacks. In: 9th International Conference on Web and Internet Economics, LNCS, vol. 8289, pp. 319–332. Springer (2013)

[11] Laszka, A., Johnson, B., Grossklags, J.: Mitigation of targeted and non-targeted covert attacks as a timing game. In: 4th International Conference on Decision and Game Theory for Security, LNCS, vol. 8252, pp. 175–191. Springer (2013)

[12] Lemke, C.E., Howson, J.T.: Equilibrium points of bimatrix games. Journal of the Society for Industrial and Applied Mathematics 12, 413–423 (1964)

[13] Mangasarian, O.L.: Equilibrium points of bimatrix games. Journal of the Society for Industrial and Applied Mathematics 12, 778–780 (1964)

[14] McKelvey, R.D., McLennan, A.M., Turocy, T.L.: Gambit: Software tools for game theory (2014), version 13.1.2, `www.gambit-project.org`

[15] Nochenson, A., Grossklags, J.: A behavioral investigation of the FlipIt game. In: Twelfth Workshop on the Economics of Information Security (2013)

[16] Okhravi, H., Hobson, T., Bigelow, D., Streilein, W.: Finding focus in the blur of moving-target techniques. IEEE Security and Privacy 12(2), 16–26 (2014)

[17] Pfleeger, C.P., Pfleeger, S.L.: Analyzing Computer Security: A Threat/Vulnerability/Countermeasure Approach. Prentice Hall (2012)

[18] Pham, V., Cid, C.: Are we compromised? Modelling security assessment games. In: 3d International Conference on Decision and Game Theory for Security, LNCS, vol. 7638, pp. 234–247. Springer (2012)

[19] Ross, S.M.: Stochastic Processes. Wiley, 2d edn. (1995)

[20] Roy, S., Ellis, C., Shiva, S.G., Dasgupta, D., Shandilya, V., Wu, Q.: A survey of game theory as applied to network security. In: 43rd Hawaii International Conference on System Sciences (2010)

[21] Vratonjic, N., Hubaux, J.P., Raya, M., Parkes, D.C.: Security games in online advertising: Can ads help secure the web? In: Ninth Workshop on the Economics of Information Security (2010)

[22] Wellman, M.P., Kim, T.H., Duong, Q.: Analyzing incentives for protocol compliance in complex domains: A case study of introduction-based routing. In: 12th Workshop on the Economics of Information Security (2013)

[23] Wellman, M.P., Prakash, A.: Empirical game-theoretic analysis of an adaptive cyber-defense scenario (preliminary report). In: 5th International Conference on Decision and Game Theory for Security. LNCS, vol. 8840, pp. 43–58. Springer (2014)