Empirical Game-Theoretic Analysis of an Adaptive Cyber-Defense Scenario (Preliminary Report)

Michael P. Wellman and Achintya Prakash

University of Michigan, Ann Arbor, MI, USA

Abstract. We investigate an adaptive cyber-defense scenario, where an attacker's ability to compromise a targeted server increases progressively with probing, and the defender can erase attacker progress through a moving-target technique. The environment includes multiple resources, interdependent preferences, and asymmetric stealth. By combining systematic simulation over a strategy space with game-theoretic analysis, we identify equilibria for six versions of this environment. The results show how strategic outcomes vary qualitatively with environment conditions, and demonstrate the value of reliable probe detection in setting up an effective deterrent to attack.

1 Introduction

Game-theoretic analyses of cyber-security domains typically start with stylized models of environments and agent strategies, and seek analytical characterization of solutions (e.g., equilibria) in terms of qualitative strategy properties. Such an approach often yields valuable insight, which may apply generally for broad classes of scenarios. An alternative, less frequently employed, is to start with a detailed environment model and specific dynamic strategies, and solve games based on these. We take the latter approach because it complements the former, and allows us to explore a rich set of questions without premature simplification, such as isolating all the key strategic variables in advance. This flexibility is particularly valuable for the study of *adaptive* cyber-defense, due to the complexity of analyzing strategies that interact over time.

The defining characteristic of adaptive cyber-defense, for our purposes, is that the defender policy takes into account the attack state of the system, in consideration of how successful attacks require a succession of actions to gain knowledge about and eventually compromise targeted resources. The present study incorporates only simple forms of adaptive behavior, but these are sufficient to exhibit strategically interesting decisions by both attacker and defender.

The approach we adopt is *empirical game-theoretic analysis* (EGTA) (Wellman, 2006), in which game-theoretic models are estimated from simulation data. The advantage of simulation is its ability to handle complex, stochastic, and temporally extended scenarios. Its main disadvantage is that conclusions may be difficult to generalize beyond the specific environments and strategy instances

R. Poovendran and W. Saad (Eds.): GameSec 2014, LNCS 8840, pp. 43–58, 2014.

studied. This complements traditional game-theoretic treatments, which sacrifice complexity for generality (within the simplified model). We have employed EGTA for strategic reasoning in a variety of domains, including for example collusion in privacy attacks (Duong et al., 2010) and incentives for compliance with a network security protocol (Wellman et al., 2013).

The EGTA exercise presented here demonstrates some of the interesting strategic behavior emerging from a simple adaptive cyber-defense scenario, and shows how solutions vary qualitatively depending on environmental conditions. Our results shed light on important ingredients of attacker and defender strategies, and pivotal features of environment models. We label the report "preliminary" at this stage, however, as the investigation has as yet not sufficiently explored the space of strategies and variations in environment settings that would lead us to consider the findings conclusive about strategic behavior in this domain.

2 Scenario: General Description and Related Work

Our adaptive cyber-defense scenario comprises two players: a *defender* who operates an array of networked computational assets, and an *attacker* who seeks to control or compromise these assets. For concreteness, we refer to the assets as *servers*. Servers are initially under the control of the defender, but the attacker may gain control through targeted attacks. A key feature of our scenario is that attack effort is cumulative, in that the more time an attacker has spent probing a server, the greater its prospect for successfully taking control. The main defense action is a *moving-target* technique (Jajodia et al., 2011), where the defender effectively resets the state of the server such that the attacker must restart its effort from scratch. For example, the defender could reimage a server, dynamically randomizing the layout of its address space. Our scenario model abstracts from the implementation details of this defense operation, but it falls in the category of *dynamic runtime environment* techniques, within the taxonomy of moving-target defenses presented by Okhravi et al. (2014). The point of dynamically modifying the runtime environment is that specific knowledge that the attacker has accumulated from probes to that point (e.g., based on specific memory locations of attack surfaces) is rendered obsolete by the runtime modification.

Figure 1 illustrates how a sequence of attack and defense actions can play out over time in our scenario. Attacker probes are indicated by demon heads, and defense reimage operations by reset icons. Each row represents a server, which may be under control of the defender (light blue) or attacker (dark red). Attacker probes succeed in changing control probabilistically, whereas defender reimages always work. Although the figure presents a sequence of discrete time periods, in our actual model time is continuous and actions are asynchronous.

The setup we investigate shares several features with the FlipIt abstract cybersecurity game (van Dijk et al., 2013). As in FlipIt, a server is at any time under the control of one of the players, and gaining this control is the players' main interest. Also like FlipIt, our scenario exhibits *stealth*, in that defenders do not know when the attacker has taken control. Our scenario also incorporates some

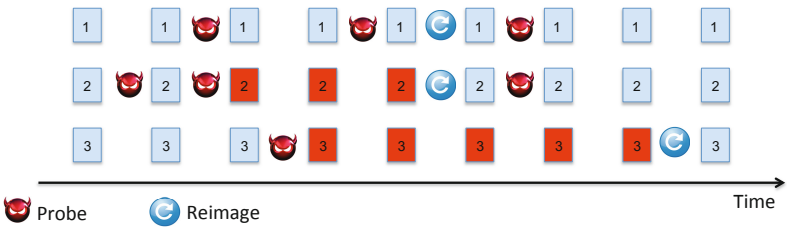

Fig. 1. Illustrative timeline of our adaptive cyber-defense scenario

major extensions of the FlipIt model, along with other important differences. First, we consider multiple servers, which is not simply the same as playing multiple FlipIts simultaneously because the utility of controlling or compromising one server depends on the state of others. The second major extension is a finer-grained model of attack actions, which compromise a server through cumulative acquisition of knowledge rather than in a one-shot takeover. Each probe action succeeds in taking over a server with some probability, which is increasing in the number of probes since the last defender reset. This is an essential feature, since as pointed out by Evans et al. (2011), moving target defenses are effective only when the attack process is incremental or progressive in some way. Finally, the stealth in our scenario is asymmetric. Attackers know when they have compromised a server, and when the defender has retaken control. And though defenders cannot tell whether an attack has succeeded, they can detect the attacker probe actions.

Extensions related to these were also studied in a series of recent papers, most written by Laszka and colleagues. In one extension, the authors incorporate multiple servers, and model objectives at two extremes where attacker control of one or all is required to control the system (Laszka et al., 2013a). Pham and Cid (2012) introduce sensing actions that reveal the compromise state of the server, at some cost. In the FlipIt version studied by Laszka et al. (2013b), the effect of attack actions is not immediate, but rather the compromise takes a stochastic amount of time. These same authors also investigate a variation in which defense actions are non-stealthy (Laszka et al., 2013c); that is, as in our scenario, attackers are aware of the state of server control.

Each of these extensions is well motivated by practical realism, but seriously complicates analysis of the FlipIt game, which to date has eluded complete analytic solution, even in its basic version. The works cited provide partial analytic solutions, contributing significant strategic insights on individual issues. In order to combine multiple issues and enable extension to yet richer environments, we adopt a simulation-based approach.

3 Detailed Game Specification

The two players in our game vie for the control of m servers, with $m = 3$ in the environment instances investigated here. The scenario runs for a finite horizon of T time units. We set $T = 10,000$ for the present study.

3.1 States and Actions

At any point of time, the state of each server can be described by 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
- 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 $e^{-\alpha(\rho_t+1)}$, $\chi_{t+} = def$. In our focal environment, we set $\alpha = 0.05$.

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

As noted above, our observation model is asymmetric with respect to the two players. The defender is aware of every probe that is 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:

- The attacker perfectly observes the state at $t+$.
- If $\upsilon_t = up$, the defender detects the probe, and can therefore infer ρ_{t+}.

Following a reimage action:

- 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 at $t+$.
- The defender perfectly observes the 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. The defender always knows ρ, but except right after a reimage does not know χ.

3.3 Utility

Each player accrues utility depending on the number of servers in their control per unit time. Let u_k^i denote the rate of utility accrual for player $i \in \{att, def\}$ when i controls $k \in [0, m]$ servers. For example, if i controls m servers for $T/2$ time units, then loses control of one server for the remaining $T/2$ time units, its utility accrued would be $(T/2)u_m^i + (T/2)u_{m-1}^i$. We normalize by setting $u_0^i = 0$ and $u_m^i = 1$. Utility for control is monotonic: $k' > k \Rightarrow u_{k'}^i \geq u_k^i$. In one example instance (with $m = 3$), we set $u_1^{def} = 0.1$, $u_2^{def} = 0.7$, $u_1^{att} = 0.3$, and $u_2^{att} = 0.7$. With these settings, the attacker's utility per server compromised is close to linear, whereas the defender takes a particularly large penalty for losing control of its second server.

We can interpret these utility values in terms of the so-called "CIA triad": Confidentiality, Integrity, Availability (Pfleeger and Pfleeger, 2012). From the defender's perspective, confidentiality is maintained when all servers are under its control, availability when any of them are, and integrity (in a rough sense) when the predominance of servers are controlled by the defender. A low value for u_2^{def} would indicate that confidentiality is paramount, as most utility is lost if even one server goes out of control. Conversely, a high value of u_1^{def} would represent that the defender is concerned primarily with availability.

In addition, our model imposes a cost for executing actions. Invoking a reimage costs the defender $c_D > 0$ per unit of downtime, or a total of $c_D \Delta$ per reimage.

The attacker pays a cost of $c_A > 0$ per probe. In our study we set $c_A = 0.2$, and consider downtime costs $c_D \in \{0.3, 0.6\}$.

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 strategy implementations interact with a discrete-event simulation of the environment. Any time that a player's knowledge state changes (see §3.2), the player strategy is queried for its next action—time and target server—assuming that it gets no further observations in the meantime. Depending on the strategy, the player may choose to retain its pending (previously scheduled) action, or to replace it on the queue with the action selected based on its latest knowledge. 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

The heuristic attacker strategies we consider are basically periodic, differing on the period P and the criteria by which they choose the server to target. We have thus far defined three different selection strategies.

- *Uniform-Uncompromised.* The attacker selects uniformly at random among those servers under the defender's control ($\chi_t = def$).
- *MaxProbe-Uncompromised.* The attacker 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.
- *Uniform-Uncompromised-or-Threshold.* The attacker considers servers eligible for probe if they are under the defender's control, *or* if they have not been probed within the last τ time units. The rationale for attacking servers already compromised is to prevent the defender from inferring from lack of probes that it has lost control of a server. The attacker selects uniformly among the eligible servers.

We implemented two different policies for employing these selections in a periodic manner. In *Periodic-A* strategies, the attacker schedules a probe at time $t + P$ on the designated server (or null, if no servers meet the eligibility criteria).

If the attacker observes a state change at time $t < t' < t + P$, it withdraws its pending probe, reconsiders according to the specified criteria, and schedules a new probe for $t' + P$. In *Periodic-B* strategies, the attacker selects a server based on its criteria at time t, and executes the probe immediately. It then schedules a dummy action for $t + P$ so that it can evaluate its choice at that time.

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

4.2 Defender Strategies

For the defender, we define two selection criteria for periodic strategies, and one heuristic based on probe activity. The periodic strategies are defined by their period P, criteria for selecting which server to reimage, and the periodic management policy. A defender using *Periodic-A* strategies schedules a reimage at time $t + P$ on the designated server. If all servers are down, the defender schedules a dummy action for $t+P$ and checks whether any servers are up at that time. In case a server comes up at a time $t' \in [t, t + P]$, the defender schedules a reimage action at $t' + P$. In contrast to *Periodic-A* attackers, the *Periodic-A* defenders do not reconsider their scheduled reimage based on observations within the period. *Periodic-B* defenders select a server to reimage based on their knowledge state at t, and initiate the reimage immediately.

- *Uniform.* The defender selects uniformly at random among all active servers ($\upsilon_t = up$)
- *MaxProbe.* The defender selects the active server that has been probed most since the last reimage, breaking ties uniformly.

The third strategy triggers a reimage operation based on probe activity or inactivity.

- *ProbeCount-or-Period.* The defender 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.

Finally, defenders may also adopt the null strategy *No-Op*.

5 Empirical Game Analysis

We analyze our adaptive cyber-defense scenario using *empirical game-theoretic analysis* (EGTA), an approach combining simulation with game-theoretic reasoning. Starting with a representative set of heuristic strategies for attackers and defenders, we evaluate their interactions by repeated simulation. Outcomes from these scenarios are used to define payoffs for the respective strategies in the game, and the resulting game model is analyzed to determine strategy profiles that are in game-theoretic equilibrium.

5.1 Simulation Setup and Game Model Generation

Our scenario specification includes several configurable parameters, described in §3. As noted above, the scenario instances studied here take $m = 3$, $T = 10,000$, $\alpha = 0.05$, $c_A = 0.2$, $c_D \in \{0.3, 0.6\}$, and $\Delta = 7$. We analyze six environment instances, differing only in defender utility for server control, and downtime cost for reimaging. Specifically, our environments employ the following utility settings, each for both Low ($c_D = 0.3$) and High ($c_D = 0.6$) downtime costs:[1]

Int: $u_1^{def} = 0.1$, $u_2^{def} = 0.7$
Con: $u_1^{def} = 0.1$, $u_2^{def} = 0.2$
Ava: $u_1^{def} = 0.8$, $u_2^{def} = 0.9$

We label these utility settings by the "CIA" features they emphasize, respectively: integrity, confidentiality, and availability. For all environments we take $u_1^{att} = 0.3$ and $u_2^{att} = 0.7$, which means that the attackers value server control in an approximately linear manner.

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. Actions may be directly executable (fully specified probes or reimages), or may be dummy action objects that require a call to the agent strategy to flesh out in detail. 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 (or dummy actions) on the queue.

Table 1 lists the strategy instances that we included in our evaluation. For each attacker (Att) or defender (Def) heuristic, we specify the parameter values covered. For example, we included Uniform-Uncompromised attacker strategies, in both *Periodic-A* and *Periodic-B* policy versions, for each of the nine periods (P) listed. For the Uniform-or-Threshold attackers we considered all combinations of the indicated P and τ values (25 total). For the defender, one P value was inadvertently omitted in the *Periodic-A* case. We instantiated ProbeCount-or-Period 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. Altogether, we included 87 attacker and 58 defender strategy instances.

For each environment, we ran simulations of all $87 \times 58 = 5046$ strategy profiles. 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.

5.2 Game-Theoretic Analysis Process

Once we have a normal-form game model for a specified scenario, we proceed to analyze it using standard game-theoretic algorithms. Our analysis followed the

[1] For the Ava/Low environment, we also ran a version with $T = 1000$, which produced identical game analysis results.

Table 1. Strategy instances included in our EGTA study

Att/Def	**Heuristic**	A/B	P	π or τ
Att	Uniform	A,B	0.1, 0.5, 0.7, 4, 7, 11, 15, 17, 20	—
Att	MaxProbe	A,B	0.1, 0.5, 0.7, 4, 7, 11, 15, 17, 20	—
Att	Uniform-or-Threshold	A,B	0.1, 5, 11, 17, 20	1, 5, 13, 27, 35
Def	Uniform	A	3, 15, 23, 31, 46, 57, 67, 75	—
Def	Uniform	B	3, 7, 15, 23, 31, 46, 57, 67, 75	—
Def	MaxProbe	A,B	3, 7, 15, 23, 31, 46, 57, 67, 75	—
Def	ProbeCount-or-Period	—	10, 50, 100, 150, 200, 250, 300	1, 2, 3, 4
Att,Def	No-Op	—	—	—

process displayed in Figure 2. As described above, simulating all combinations of attacker and defender strategies yields estimated payoffs for a normal-form game. We then simplify the game by removing dominated strategies. This produces a game model we can solve with standard algorithms, employing Nash equilibrium as a solution concept. Further analysis yields insight on the qualitative performance of heuristic strategies. The remainder of this section and the next elaborate on the last three steps, and their application to the six environment instances studied here.

Fig. 2. Empirical game-theoretic analysis pipeline

We start by eliminating strategies that are strictly dominated. Such strategies cannot be part of any Nash equilibrium, and removing such strategies simplifies the game. Moreover, eliminating a dominated strategy may render other strategies dominated, hence we iterate the pruning process. One pass of *iterated elimination of strictly dominated strategies* (IESDS) removes a strategy for one player such that there exists another strategy that performs strictly better regardless of the other player's choice among its remaining strategies. We implemented IESDS using the algorithm of Knuth et al. (1988). Starting from games of size 87×58 for each of the six environments, IESDS is able to prune 2–8 attacker strategies, and 6–41 defender strategies. Using linear programming to compute domination by mixtures eliminated just a few additional strategies. The residual subgames are still too large for our available game solver, so we require a more aggressive pruning regimen.

Toward this end, we eliminate some strategies that are not strictly dominated, but would be if the dominating strategy (or mixture of strategies) were given a boost by some small payoff increment δ. This concept, called δ-*dominance*

(Cheng and Wellman, 2007), has been found to achieve significant simplification with modest loss of accuracy. Although such aggressive pruning can eliminate some equilibrium strategies, all equilibria of the game after iterated "weaker-than-weak" elimination are guaranteed to be approximate equilibria of the original game. Specifically, they are ϵ-Nash equilibria for any ϵ less then or equal to cumulative δ (i.e., sum over iterations) employed for elimination.

In our study, we employed δ-dominance elimination as necessary to reduce the number of strategies for each player to 42 or fewer: the size we determined our solver could handle. In each round, we identified and removed the strategy requiring minimum δ for elimination, then further pruned by IESDS (i.e., $\delta = 0$). In all cases, we were able reduce the game sufficiently with relatively small cumulative δ. For our six environments, δ-IESDS achieved reduced sizes as follows:

Int/Low: 42×18, with cumulative $\delta = 1.3$
Con/Low: 42×11, with cumulative $\delta = 0.8$
Ava/Low: 22×28, with cumulative $\delta = 0.03$
Int/High: 38×39, with cumulative $\delta = 0.02$
Con/High: 28×35, with cumulative $\delta = 0.02$
Ava/High: 31×38, with cumulative $\delta = 0.03$

We calculate Nash equilibria using Gambit (McKelvey et al., 2014), a general tool for game-theoretic computation. Gambit has some difficulty with games even of this size, so we feed it a series of smaller games, with all combinations of three undominated attacker strategies against the full set of undominated defender strategies. This produces a set of candidate equilibria, which we then filter by testing deviations from the rest of the undominated attacker set. This process will produce all equilibria with attacker support three or fewer in the shrunken game, as long as the subgame solutions are exhaustive.

5.3 Equilibrium Results and Analysis

We found three qualitatively distinct equilibria for this adaptive cyber-defense scenario, which manifest across the six environments in an intuitively sensible pattern (see Table 2). We did not conduct an explicit statistical analysis of the results, as it was quite apparent that the sampling error made no difference to the qualitative equilibrium conclusions.

Maximal Defense. In the *Maximal Defense* (MaxDef) equilibrium, 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. For example, suppose the defender plays ProbeCount-or-Period with $\pi = 1$ (abbreviated by $PCP(1, x)$, as the period is irrelevant at that setting), which means it reimages as soon as it sees a probe. Even if the attacker's probe were successful, it would not maintain control for more than an instant, so the probe had cost without benefit. The best response for the

attacker is therefore No-Op. Against the No-Op attacker, the aggressive PCP defense never actually has to reimage, so it achieves the greatest possible utility (continual control of all servers, no reimaging cost) in this equilibrium.

Table 2. Nash equilibria for the six environments studied

	Defender Utility (u_1^{def}, u_2^{def})		
c_D	**Int** (0.1, 0.7)	**Con** (0.1, 0.2)	**Ava** (0.8, 0.9)
Low (0.3)	MaxDef	MaxDef	MaxDef, PerΔ
High (0.6)	MaxDef, MaxAtt	MaxDef, MaxAtt	MaxDef, PerΔ

MaxDef is an equilibrium profile for all six of our environments. The attacker utility is constant across these instances, and defender utility is maximal in all as well given full control and no reimaging. Technically, there are a large set of MaxDef equilibria, where the attacker plays No-Op and the defender plays some mixture of PCP strategies that are sufficient to deter probes. Specifically, No-Op is a best response for the attacker against $PCP(1, x)$, $PCP(2, 10)$, or $PCP(3, 10)$, and any PCP strategy is a best response against No-Op. Any mixture of the strategies listed against No-Op would therefore constitute an equilibrium, as would mixtures of these along with some probability of playing other PCP strategies, as long as the components of the most aggressive PCP strategies are probable enough to deter the attacker from probing.

In fact, our argument that MaxDef is in equilibrium applies under the general assumptions of this scenario, even with respect to the full space of possible attacker and defender strategies.

Proposition 1. *For any environment parameter settings and any strategy sets that include No-Op for the attacker and $PCP(1, x)$ for the defender, the profile comprising these strategies is a Nash equilibrium.*

Proof. Suppose the defender plays $PCP(1, x)$. The only way an attacker can gain positive utility is to compromise servers through probing. However, with $PCP(1, x)$ the defender immediately takes back control on compromise, and so the attacker ends up accruing zero utility, $u_0^{att} = 0$, regardless. The No-Op strategy achieves zero payoff, which is better than any strategy that involves any probing. Therefore No-Op is a best response to $PCP(1, x)$, among all possible attacker strategies.

Suppose the attacker plays No-Op. In that case, the defender keeps control of all servers, and accrues maximum utility $u_m^{def} = 1$ with any strategy. The strategy $PCP(1, x)$ never reimages and thus incurs zero cost, so overall payoff is maximal. Therefore $PCP(1, x)$ is a best response to No-Op, among all possible defender strategies.

We have established the Nash equilibrium with no reference to variable parameters m, T, α, c_A, c_D, Δ, or u_k^i for $0 < k < m$, thus the result holds for any legal settings. □

For two of our environments, MaxDef is the only equilibrium found. The other four have additional equilibria.

Maximal Attack *Maximal Attack* (MaxAtt) is the flip-side to MaxDef, where the attacker probes sufficiently aggressively to deter active defense. Such deterrence applies when the utility the defender can achieve by taking control of servers through reimaging is not worth its cost. In such a case, the defender's best response is No-Op. When the defender plays No-Op, the attacker maximizes payoff by taking control of the servers as quickly as possible, which for our strategy set is achieved by the periodic (Uniform or MaxProbe) strategies, with $P = 0.1$. The differences between Uniform and MaxProbe, and between *Periodic-A* and *Periodic-B* in this situation are statistically indistinguishable.

MaxAtt is an equilibrium for environments Int/High and Con/High, but not the others. To see why this is the case, consider that when the attacker is probing at high frequency, maintaining control of the server requires reimaging it almost as soon as it comes back up from the last reimage. We can gauge whether this is worthwhile by comparing the utility for controlling servers with the downtime cost (c_D), since both parameters are in units of payoff/time. For environment Int/Low, keeping control of one server is not worthwhile ($u_1^{def} = 0.1 < 0.3 = c_D$), but it is worthwhile to keep control of *two* ($u_2^{def} = 0.7 > 0.6 = 2c_D$). When we double c_D to get environment Int/High, however, it is not worthwhile to keep control of any number of servers against a high-frequency attacker. Therefore MaxAtt is an equilibrium for Int/High, and for Con/High as well. It is not an equilibrium for Ava/High, as defense is worthwhile for one server even at the high downtime cost. At the low cost, MaxAtt is not in equilibrium for any values of u_1^{def} and u_2^{def}, as defending all three servers is worthwhile regardless: $u_3^{def} = 1.0 > 0.9 = 3c_D$.

We illustrate the outcomes of responding to an aggressive attacker (Uniform selection, *Periodic-A*, $P = 0.1$), in Figure 3. The top plot shows defender payoffs for a range of periodic strategies. In both Ava environments, the defender accrues the greatest payoff by choosing a period that maintains control of one server. The maximum is achieved at a period coinciding with reimaging downtime. A higher frequency of reimaging incurs more downtime cost, whereas at lower frequency the defender controls no servers for much of the time. For the Int and Con cases, respectively, the defender's utility function particularly rewards controlling two and three servers. With High downtime cost, the periodic defender cannot make reimaging worthwhile, which is reflected in payoffs increasing toward zero with longer periods. For Int with Low downtime cost, the defender can achieve small positive payoff with a short period, which refutes MaxAtt as an equilibrium for this environment.

The bottom plot shows the response of PCP strategies. Since these have two parameters (π and P), they cannot be ordered linearly on the x-axis. For each environment, we can discern a pattern of payoffs as P is increased for each ProbeCount threshold π. Some are increasing, some decreasing, and the Ava environments in particular exhibit interior maxima. The positive payoffs for $PCP(1, x)$ and $PCP(2, 10)$ refute MaxAtt as an equilibrium in Con/Low.

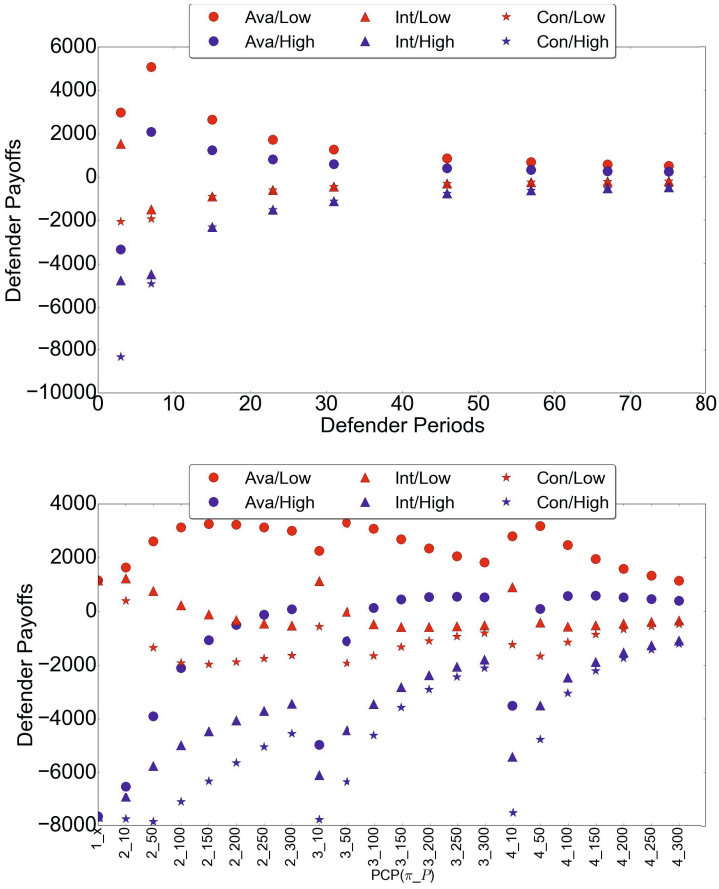

Fig. 3. Defender payoffs versus a maximally aggressive attacker, for all six environments. Marker shapes represent the different utility settings and colors represent different downtime costs. (top) Payoffs for periodic defender strategies, varying P, against $Unif(A, 0.1)$. (bottom) Payoffs for PCP defender strategies, varying π_P, against $Unif(A, 0.1)$.

5.4 Periodic Δ Reimage

Our final equilibrium, *Periodic Δ Reimage* (PerΔ), has the defender playing a periodic (MaxProbe, *Periodic-B*) strategy with $P = 7$, against a high-frequency defender (MaxProbe, *Periodic-B* with $P = 0.1$). As suggested above, that the defender's period equals the downtime interval Δ is not a coincidence. By this strategy, the defender keeps one server under its control (albeit down all the time), leaving the other two to be grabbed by the attacker. These strategies are in equilibrium for the Ava utility environments (both Low and High downtime cost), where the defender gets the lion's share of its possible utility by controlling one server.

PerΔ is not an equilibrium for the other environments, where controlling a single server in this way is not worthwhile.

5.5 Discussion

We offer several observations about these results. First, the game-theoretic solutions produced by our EGTA pipeline (Figure 2) *make sense*. As explained above, it is intuitively clear why the equilibria identified are equilibria for their respective environments. In retrospect, it should have been possible to identify some of these without the extensive simulation and game-theoretic reasoning process undertaken, but in our experience it often takes some concrete simulation to expose the obvious in a complex environment. In any case, the simulations serve to confirm the reasoning given, and the fact that sensible strategy profiles emerged from the search counts as validation of the overall approach. Moreover, having considered a broad variety of alternative strategies provides information about other plausible heuristics that turn out not to be part of equilibrium solutions.

Of course, 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 strategies resemble similar strategies analyzed in studies of FlipIt), but omit many others (e.g., stochastic renewal 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.

Introducing further strategies could potentially overthrow the MaxAtt and PerΔ equilibria we found, though as we showed (Proposition 1), the MaxDef equilibrium will persist regardless of additional strategies.

For the environments with two 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.

It is also important to question whether features of the environment that produce these results are entirely realistic. For instance, it seems strange to give the defender so much credit for controlling a server that is down, particularly if availability is the basis for a particular utility function. We saw that changing the relative cost of downtime compared to server control utility (i.e., the Low versus High environments) could indeed affect equilibria. Moreover, the analysis underscores the necessity of interpreting a particular setting of c_D relative to the utility settings u^{def}.

The fact that MaxDef is always in equilibrium also prompts scrutiny about environment assumptions. The credibility of the defender responding to every probe relies crucially on the power the defender has to perfectly detect such probes. Any inaccuracy in this detection would undermine the maximal defense. If there were a significant prospect of false positives, this policy might be too costly to the defender. Or with false negatives, the attacker could get some traction even against the maximal defense.

6 Conclusions

We studied a simple scenario in adaptive cyber-defense. The model employs abstract models of actions and attacker and defender objectives, yet goes beyond previous models in simultaneously accommodating multiple resources, progressive attack behavior, and asymmetric stealth. Through empirical methods, relying heavily on simulation coupled with game-theoretic reasoning, we identified equilibrium strategy profiles for a variety of environment settings. Though the results must be considered preliminary (sparse coverage of the space of environments, provisional equilibria based on incomplete strategy sets), the pattern of equilibria we found reveal interesting strategic interactions between the attacker and a moving-target defender. In particular, having perfect ability to detect probes gives a defender a powerful deterrent strategy, applicable in a broad range of environment settings.

Our study also illustrates empirical game-theoretic methodology in a salient security domain. The simulation approach allows us to deal with dynamic complexity in the environment, yet still apply standard game-theoretic solution concepts.

Work on this scenario, and modeling adaptive cyber-defense more generally in this framework, is ongoing. In addition to the extensions noted in §5.5, we intend to explore environments with a range of probe efficacy (e.g., settings of α), stochastic downtimes, and alternative attacker utility models. We are also focusing on extending the space of strategies to include those far more adaptive to opponent behavior, including intent inference, and explicit reasoning about threats and counter-threats.

Acknowledgment. This work was supported in part by MURI grant W911NF-13-1-0421 from the US Army Research Office. We thank George Cybenko and other participants in this project for contributions to defining the scenario, and Satinder Singh Baveja for constructive comments on an earlier draft.

References

Cheng, S.F., Wellman, M.P.: Iterated weaker-than-weak dominance. In: 20th International Joint Conference on Artificial Intelligence, Hyderabad, pp. 1233–1238 (2007)

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

Duong, Q., LeFevre, K., Wellman, M.P.: Strategic modeling of information sharing among data privacy attackers. Informatica 34, 151–158 (2010)

Evans, D., Nguyen-Tuong, A., Knight, J.: Effectiveness of moving target defenses. In: Jajodia, et al. (2011)

Jajodia, S., Ghosh, A.K., Swarup, V., Wang, C., Wang, X.S. (eds.): Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats. Springer (2011)

Knuth, D.E., Papadimitriou, C.H., Tsitsiklis, J.N.: A note on strategy elimination in bimatrix games. Operations Research Letters 7, 103–107 (1988)

Laszka, A., Horvath, G., Felegyhazi, M., Buttyán, L.: `FlipThem`: Modeling targeted attacks with `FlipIt` for multiple resources. In: Poovendran, R., Saad, W. (eds.) GameSec 2014. LNCS, vol. 8840, pp. 173–192. Springer, Heidelberg (2014)

Laszka, A., Johnson, B., Grossklags, J.: Mitigating covert compromises: A game-theoretic model of targeted and non-targeted covert attacks. In: Chen, Y., Immorlica, N. (eds.) WINE 2013. LNCS, vol. 8289, pp. 319–332. Springer, Heidelberg (2013b)

Laszka, A., Johnson, B., Grossklags, J.: Mitigation of targeted and non-targeted covert attacks as a timing game. In: Das, S.K., Nita-Rotaru, C., Kantarcioglu, M. (eds.) GameSec 2013. LNCS, vol. 8252, pp. 175–191. Springer, Heidelberg (2013c)

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

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)

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

Pham, V., Cid, C.: Are we compromised? Modelling security assessment games. In: Grossklags, J., Walrand, J. (eds.) GameSec 2012. LNCS, vol. 7638, pp. 234–247. Springer, Heidelberg (2012)

Wellman, M.P.: Methods for empirical game-theoretic analysis (extended abstract). In: 21st National Conference on Artificial Intelligence, Boston, pp. 1552–1555 (2006)

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