Games of Timing for Security in Dynamic Environments

Benjamin Johnson[1], Aron Laszka[2], and Jens Grossklags[3]

[1] CyLab, Carnegie Mellon University, Pittsburgh, USA
[2] Institute for Software Integrated Systems, Vanderbilt University, Nashville, USA
[3] College of Information Sciences and Technology, Pennsylvania State University, University Park, USA

Abstract. Increasing concern about insider threats, cyber-espionage, and other types of attacks which involve a high degree of stealthiness has renewed the desire to better understand the timing of actions to audit, clean, or otherwise mitigate such attacks. However, to the best of our knowledge, the modern literature on games shares a common limitation: the assumption that the cost and effectiveness of the players' actions are time-independent. In practice, however, the cost and success probability of attacks typically vary with time, and adversaries may only attack when an opportunity is present (e.g., when a vulnerability has been discovered).

In this paper, we propose and study a model which captures dynamic environments. More specifically, we study the problem faced by a defender who has deployed a new service or resource, which must be protected against cyber-attacks. We assume that adversaries discover vulnerabilities according to a given vulnerability-discovery process which is modeled as an arbitrary function of time. Attackers and defenders know that each found vulnerability has a basic lifetime, i.e., the likelihood that a vulnerability is still exploitable at a later date is subject to the efforts by ethical hackers who may rediscover the vulnerability and render it useless for attackers. At the same time, the defender may invest in mitigation efforts to lower the impact of an exploited vulnerability. Attackers therefore face the dilemma to either exploit a vulnerability immediately, or wait for the defender to let its guard down. The latter choice leaves the risk to come away empty-handed.

We develop two versions of our model, i.e., a continuous-time and a discrete-time model, and conduct an analytic and numeric analysis to take first steps towards actionable guidelines for sound security investments in dynamic contested environments.

Keywords: Security, Game Theory, Games of Timing, Vulnerability Discovery

1 Introduction

Since at least the Cold War era there has been a considerable interest in the study of games of timing to understand *when* to act in security-relevant decision-making scenarios [1]. The recent rise of insider threats, cyber-espionage, and

other types of attacks which involve a high degree of stealthiness has renewed the desire to better understand the timing of actions to audit, clean, or otherwise mitigate such attacks. However, to the best of our knowledge, the modern literature on games and decision-theoretic approaches (including the FlipIt model [3,31]) shares a common limitation: the assumption that the cost and effectiveness of the players' actions are time-independent. For example, in the FlipIt model and its derivatives (see section on related work), an adversary may make a move at any time for exactly the same fixed cost, and these moves always succeed.

In practice, the cost and success probability of attacks typically vary with time. Moreover, an adversary may only attack when an opportunity is present (e.g., when a vulnerability has been discovered). These observations motivate the development of games of timing which take into account the dynamic environment of contested computing resources. Defenders need to develop an optimal defensive strategy which considers the nature of vulnerability discovery by adversaries. At the same time, the attacker faces the decision-making dilemma on when to exploit an identified vulnerability.

For example, the black hat community knew already for a long time that Microsoft would stop supporting Windows XP in April 2014, which would significantly lower the defense and mitigation effort for this software product.[4] Security professionals conjectured that attackers would begin stockpiling vulnerabilities to exploit them more profitably. However, under what circumstances is such behavior optimal for the attacker, when there is a risk that the vulnerability is rediscovered by an internal security team or external ethical hackers before the planned time of exploitation [22,35]?

In this paper, we propose and study a model which captures dynamic environments. More specifically, we study the problem faced by a defender who has deployed a new service or resource, which must be protected against cyber-attacks. We assume that adversaries discover vulnerabilities according to a given vulnerability-discovery process which is modeled as an arbitrary function of time. Attackers and defenders know that each found vulnerability has a basic lifetime, i.e., the likelihood that a vulnerability is still exploitable at a later date is subject to the efforts by ethical hackers who may rediscover the vulnerability and render it useless for attackers. At the same time, the defender may invest in mitigation efforts to lower the impact of an exploited vulnerability. Attackers therefore face the dilemma to either exploit a vulnerability immediately, or wait for the defender to let its guard down. The latter choice leaves the risk to come away empty-handed.

We develop two versions of our model, i.e., a continuous-time and a discrete-time model, to increase the applicability of our work. We provide fundamental constraints on the shape of equilibria for both models, and give necessary and sufficient conditions for the existence of non-waiting equilibria in terms of the

[4] In July 2011, Microsoft made the announcement that support for the operating system will end in 2014. Note that previously Microsoft already stopped the so-called full mainstream support for Windows XP in April 2009.

shape of the vulnerability discovery function. We further provide numerical results to illustrate important properties of our findings.

The remainder of this paper is organized as follows. In Section 2, we summarize related theoretical and behavioral work on security games of timing. In Section 3, we introduce our game-theoretic model including players and the decision-making environment. In Section 4, we derive theoretical results for our model. In Section 5, we present numerical examples. Finally, in Section 6, we discuss our results and offer concluding remarks.

2 Related Work

2.1 Security Economics and Games of Timing

The economics of security decision-making is a rapidly expanding field covering theoretical, applied, and behavioral research. Theoretical work utilizes diverse game-theoretic and decision-theoretic approaches, and addresses abstract as well as applied scenarios. A central research question has been how to optimally determine security investments [7,32,11,25], e.g., by selecting from different canonical defense actions (i.e., protection, mitigation, risk-transfer) [12,19], and how such investments are influenced by the actions of strategic attackers [6,30]. Another frequently addressed aspect has been the consideration of interdependence of security decision-making and the propagation of risks [4,8,13,14]. Recent surveys summarize these research efforts in great detail [2,15,20].

An often overlooked but critical decision dimension for successfully securing resources is the consideration of *when* to act to successfully thwart attacks. Scholars have studied such time-related aspects of tactical security choices since the cold-war era by primarily focusing on zero-sum games called *games of timing* [1]. The theoretical contributions on some subclasses of these games have been surveyed by [27].

Recently, the question of the optimal timing of security decisions has again become a lively research topic with the development of the FlipIt game [3,31]. In the following, we discuss FlipIt as well as theoretical and behavioral follow-up research.

2.2 Theoretical Analyses of FlipIt

The FlipIt model identifies optimal timing-related security choices under targeted attacks [3,31]. In FlipIt, two players compete for a resource that generates a payoff to the current owner. Players can make costly moves (i.e., "flips") to take ownership of the resource, however, they have to make moves under incomplete information about the current state of possession. In the original FlipIt papers, equilibria and dominant strategies for simple cases of interaction are studied [3,31].

In follow-up research, Pham and Cid studied a version of FlipIt with periodic strategies with random phase. They also considered the impact of a move to check the state of the game (i.e., audit) [26].

Laszka et al. study games of timing with non-covert defender moves. They consider also non-instantaneous attacker moves, and different types of adversaries, e.g., targeting and non-targeting attackers [18]. A follow-up paper further generalizes the results of this line of research [17].

The previous papers considered FlipIt with one resource. This limitation has been addressed with the strategic analysis of the game with multiple contested resources [16]. Similarly, an extension of the game has been proposed with multiple defenders [24].

Feng et al. [5] and Hu et al. [10] study games with multiple layers in which in addition to external adversaries the actions of insiders (who may trade information to the attacker for a profit) need to be considered. Hu et al. [10] study the scenario in a dynamic game framework.

Zhang et al. [34] study the FlipIt game with resource constraints on both players.

Drawing on the setup of FlipIt, Wellman and Prakash develop a discrete-time model with multiple, ordered states in which attackers may compromise a server through cumulative acquisition of knowledge rather than in a one-shot takeover [33].

2.3 Behavioral Studies of FlipIt

Nochenson and Grossklags describe and analyze two experiments which draw from the theoretical model of the FlipIt game [21]. They conduct a Mechanical Turk experiment with over 300 participants in which each participant is matched with a computerized opponent in several fast-paced rounds of the FlipIt game. Preliminary analysis of this experiment shows that participant performance improves over time (however, older participants improve less than younger ones). They also found significant performance differences with regards to gender and a measure of the desire for deep reasoning about a problem (i.e., need for cognition).

In follow-up work, Reitter et al. contrast two experiments where the feedback to the human decision maker in the decision-environment is varied between visual feedback with history, and temporal feedback without history. The authors study the human strategies and develop a model backed by a cognitive architecture, which described human heuristics that practically implement risk-taking preference in timing decisions [28].

Grossklags and Reitter extend these preliminary works with an in-depth analysis of the experimental data of these previous studies [9]. In particular, they study the interaction effects between the psychometric measures including also the general propensity of risk taking with task experience and how those factors explain task performance.

The behavioral studies will help to develop theoretical models which take the imperfections of human decision-making into account. Likewise, theoretical studies of rational behavior serve as an important comparison baseline for experimentally generated human data or measurements from the field.

3 Model

Our model captures the motivational aspects of timing, as it pertains to the discovery, repair, and exploitation of software vulnerabilities. The salient features of our model may be enumerated as follows.

1. The life cycle of a software product is finite with a known end time $t = T$.
2. The rate of vulnerability discovery $V(t)$ is an arbitrary function of time, specified as an exogenous parameter. We make this modeling choice to maximize applicability for varieties of software products and services that may differ in quality, attention, and life cycle.[5]
3. The lifetime of a vulnerability decays at a fixed rate λ without action by either player. This choice is made to account for the fact that unknown vulnerabilities are often repaired by chance only, so that one might reasonably assume they die with some fixed probability in a unit of time.[6]
4. The defender's security investment $d(t)$ is a function of time, and serves to mitigate losses when a vulnerability is exploited.
5. The timing of vulnerability exploitation $a(t)$ is chosen by an attacker for optimal exploitation dependent on the defender's security investments.

To further extend the applicability of our model, we describe and analyze two distinct versions – one with continuous time, and one with discrete time. In the continuous version of the model, attackers and defenders choose strategies as continuous functions of time, and the payoffs are determined by integrating expected losses over the range of all time. In the discrete version, time is divided into a finite number of steps; attackers and defenders choose an action at each time step, and the payoffs are determined by summing the expected outcomes over all time periods. Both versions of the game adhere to the paradigms described above.

We begin by describing the game's players and their respective choices. We then proceed to describe the environment. Finally we discuss the consequences from a configuration of choices. Whenever applicable, we separate the specification and discussion according to either the continuous or the discrete model. For reference, a list of symbols used in this paper may be found in Table 1.

[5] A small number of studies investigate the social utility of vulnerability discovery. On the one hand, Rescorla studied the ICAT dataset of 1,675 vulnerabilities and found very weak or no evidence of vulnerability depletion. He thus suggested that the vulnerability discovery efforts might not provide much social benefit [29]. On the other hand, this conclusion is challenged by Ozment and Schechter, who showed that the pool of vulnerabilities in the foundational code of OpenBSD is being depleted [22,23]. Zhao et al. present evidence that the number of discovered vulnerabilities is declining for a majority of public company-specific vulnerability bounty programs on HackerOne [36].

[6] Unsurprisingly, statistical evidence is lacking regarding how often defenders and attackers discover the same vulnerabilities. However, empirical research by Ozment about the ethical hacker community found that vulnerability rediscovery is common in the OpenBSD vulnerability discovery history [22].

Table 1: List of Symbols

Symbol	Description
R	scaling factor between security costs and losses
λ	vulnerability repair rate
Continuous-time Model	
T	end time
$V(t)$	vulnerability discovery rate at time t
$d(t)$	defender's security investment at time t
$a(t)$	attacker's waiting time before exploiting a vulnerability discovered at time t
Discrete-time Model	
K	number of time periods
$V(k)$	expected number of vulnerabilities discovered in time period k
$d(k)$	defender's security investment in time period k
$a(k)$	attacker's waiting time before exploiting a vulnerability discovered in time period k

3.1 Players and Choices

Our game has two players, a defender and an attacker. The defender's objective is to mitigate damages from vulnerability exploitation through security investment, while the attacker's objective is to maximally exploit vulnerabilities as they are discovered. Neither the attacker nor the defender control the rate of vulnerability discovery $V(t)$, which is an exogenous function of time.

We may construe the defender's investments quite broadly, in ways other than monetary investments. For example, we may understand them as a measure of strictness in policy enforcement, which can be optimized to minimize usability loss.

On the attacker side, it is interesting to note that we would obtain the same results if we modeled the game as one containing several attackers, where each attacker randomly finds vulnerabilities according to a given rate, and then independently chooses the timing of their exploitation. However, for the sake of clear exposition, we frame the interaction as a two-player game with a single attacker.

Continuous-time Model In the continuous-time model over a time interval $[0, T]$, the defender chooses a continuous function $d(t) : [0, T] \to \mathbb{R}_{\geq 0}$ which specifies the level of her security investment at each time t. The attacker chooses a continuous function $a(t) : [0, T] \to \mathbb{R}_{\geq 0}$ which specifies how long to wait before exploiting a vulnerability discovered at time t.

Discrete-time Model In the discrete-time model with discrete time periods $0, 1, \ldots, K$, the defender chooses a function $d(k) : \{0, 1, \ldots, K\} \to \mathbb{R}_{\geq 0}$ specifying her security investment level at each distinct time period. The attacker

chooses a function $a(k) : \{0, 1, \ldots, K\} \to \mathbb{Z}_{\geq 0}$ specifying how many discrete time steps to wait before launching an attack using a vulnerability discovered in the k^{th} time period.

3.2 Environment

Here we construe the environment primarily as the security state of a software system over a finite period of time. More specifically, the rate of vulnerability discovery by attackers, $V(t)$, is a function of time, specified as an exogenous parameter. We anticipate that this modeling choice increases the applicability for different types of software products and services that may differ in quality, attention, and life cycle.

The fixing of vulnerabilities, on the other hand, follows a random process as defenders eventually rediscover vulnerabilities which have been found by the attacker. More specifically, we assume that the lifetime of a vulnerability follows an exponential distribution (parameterized by λ) without action by either player. The net effect of this eventual rediscovery is that an attacker who learns of a vulnerability at one time, cannot simply wait indefinitely for the defender's security investment to lapse.

Continuous-time Model In the continuous-time model, the vulnerability function has the form $V(t) : [0, T] \to \mathbb{R}_{\geq 0}$. The interpretation is that $V(t)$ gives the precise rate at which vulnerabilities are being discovered by the attacker for each moment of time. In terms of our analysis and computation, we will obtain the expected number of vulnerabilities discovered during any fixed time interval by integrating $V(t)$ with respect to t over that time interval.

The vulnerability repair process is determined by an exponential decay function of the form $e^{-\lambda\tau}$. This function determines the probability that a vulnerability still remains exploitable τ time after its discovery. The structured formulation guarantees that this exploit probability decays at a constant rate of λ. An approximate interpretation is that in each unit of time, a constant fraction of its exploit probability is lost.

Discrete-time Model In the discrete-time model, the vulnerability function has the form $V(k) : \{0, 1, \ldots, K\} \to \mathbb{R}_{\geq 0}$. Here, $V(k)$ gives directly the expected number of vulnerabilities discovered during the time period k. Computationally, we may obtain the expected number of vulnerabilities discovered over any sequence of time periods by summing $V(k)$ over those periods.

To capture the analogous fixed rate reduction phenomenon for vulnerability repair in the discrete-time model, we use a geometric distribution function of the form $(1 - \lambda)^{\tau}$, which gives us the probability that a vulnerability is not repaired in τ number of time periods after its discovery. The interpretation is that a λ fraction of a vulnerability's exploit potential is lost in each time period.

3.3 Consequences

Suppose that both defender and attacker have simultaneously chosen their strategies for defense d and wait times a, respectively. The consequences for the defender involve both the defense costs and the loss from vulnerability exploitation. We construe the defense function in terms of direct costs, while the amount of loss resulting from an attack is inversely proportional to the defense rate, scaled by a fixed constant R.

On the attacker's side, we are only concerned with the gain from maximally exploiting the vulnerabilities. Thus, the overall structure is that the defender's payoff is always negative, while the attacker's payoff is always positive. The sum of payoffs related to vulnerability exploitation is zero; but the game itself is not zero-sum, unless the defender abstains from any defensive investment (i.e., when $d \equiv 0$).

Continuous-time Model In the continuous-time model, the defender's objective is to minimize her total losses over the course of the time interval $[0, T]$. The defender's costs over this time interval may be easily computed as

$$\int_{t=0}^{T} d(t)dt,$$

while her losses depend in part on the waiting time of an attacker. If the attacker immediately exploits a vulnerability discovered at time t, the expected loss per unit time due to vulnerabilities discovered around time t may be expressed as

$$\frac{R}{d(t)}.$$

On the other hand, if the attacker instead waits for some time $a(t)$ before exploiting a vulnerability discovered at time t, then we must account for both the decay in vulnerability exploitability as well as adjust the timing relative to the defense investment. In this case, the expected loss per unit of time due to vulnerabilities discovered around time t will be given by

$$e^{-\lambda a(t)} \frac{R}{d(t + a(t))}.$$

Putting everything together along with the vulnerability discovery function, the defender's total payoff in the continuous-time model is given by

$$U_d = -\int_{t=0}^{T} \left(d(t) + V(t) e^{-\lambda a(t)} R \frac{1}{d(t + a(t))} \right) dt; \tag{1}$$

while the attacker's payoff is given by

$$U_a = \int_{t=0}^{T} V(t) e^{-\lambda a(t)} R \frac{1}{d(t + a(t))} dt. \tag{2}$$

Discrete-time Model In the discrete-time model, the defender's objective is to minimize her total losses over the course of the time stages $\{0, 1, \ldots, K\}$. The defender's costs are computed as a sum

$$\sum_{k=0}^{K} d(k),$$

while losses depend on the waiting time of an attacker. Suppose that an attacker waits for $a(k)$ time periods before exploiting a vulnerability discovered in time period k; then, the defender's losses due to vulnerabilities discovered in time step k will be given by

$$(1-\lambda)^{a(k)} \frac{R}{d(k+a(k))}.$$

Assembling everything together, the payoff for the defender in the discrete-time model is given by

$$U_d = -\sum_{k=0}^{K} \left(d(k) + V(k)(1-\lambda)^{a(k)} \frac{R}{d(k+a(k))} \right); \tag{3}$$

while the payoff for the attacker is given by

$$U_a = \sum_{k=0}^{K} V(k)(1-\lambda)^{a(k)} \frac{R}{d(k+a(k))}. \tag{4}$$

4 Analysis

In this section, we analyze the model to find applicable consequences for the software vulnerability scenario. We will primarily focus on Nash equilibrium configurations, in which each player is responding optimally in the current context.

We begin by giving a result in the continuous-time model that constrains the attacker's strategy at the temporal boundaries.

Proposition 1. *If $V(0) > 0$, then every equilibrium in the continuous-time model satisfies $a(0) = 0$ and $a(T) = 0$. In words, the attacker should never wait to attack at either the beginning or the end of the game.*

Proof. Suppose $a(0) > 0$. Since there is no previous time at which the attacker may have discovered a vulnerability, the defender may safely choose $d(0) = 0$ as an optimal investment. However, if the attacker knew $d(0) = 0$, she would rather prefer not to wait, in order to cause maximum damage in case a vulnerability were found at that time. This contradiction shows $a(0) > 0$ cannot be an equilibrium if $V(0) > 0$.

The second part of the proposition is more trivially deduced since it would not benefit the attacker to wait longer because there is no time remaining at the end of the game. In fact, for this reason more generally, the attacker's strategy in equilibrium must satisfy the constraint $a(t) \leq T - t$. □

Our second result constrains the attacker's strategy in any pure-strategy equilibrium. These conditions are considerably more restrictive than those in the continuous-time case. They tell us that if there is an ubiquitous risk of vulnerability discovery, then there can be no pure-strategy equilibrium in which the attacker uses any positive wait times.

Proposition 2. *If $V(k) > 0$ for each time period k, then for every pure-strategy equilibrium in the discrete-time game, we have $a(k) = 0$ for each $k = 0, 1, \ldots, K$. In words, if the attacker uses any positive wait time in the discrete-time game, then it must be part of a mixed strategy.*

Proof. We prove the result by induction on the number of time periods. When $k = 0$, the claimed result is perfectly analogous to the continuous-time model's result from the previous proposition. Obviously, there can be no previous vulnerability discovery. If the attacker waits to attack in round 0, then the defender can optimally save herself the trouble of making any security investment in round 0 (i.e., $d(0) = 0$). But if $V(0) > 0$, then this configuration is clearly not an optimal response configuration for the attacker.

But now that we know $a(0) = 0$, a very similar argument also holds for $k = 1$. We do not have any vulnerabilities from the one earlier round, because the attacker did not wait in round 0. If the attacker now waits in round 1, the defender may optimally choose not to invest in security protection in this round (i.e., $d(1) = 0$). But this configuration is not optimal for the attacker and so cannot be part of an equilibrium. The argument can now be iterated inductively for $k = 2, \ldots, K$. □

The crux of these two results is that the attacker may only optimally wait to attack in a given time period if there is some attack probability arising from a previous time period. In the continuous case, this implies only that the attacker cannot wait at the beginning of the game, because continuously increasing the wait time from $t = 0$ can still lead to positive attack probability at every point in time. On the discrete side, however, this observation precludes having any simple optimal attack strategy in which the attacker waits at all.

The next two propositions give necessary and sufficient conditions for "never waiting" to be the attacker's strategy in an equilibrium. In both the continuous-time model and the discrete-time model, the conditions involve only a simple relation between the vulnerability discovery function V and the discovery rate λ.

Proposition 3. *In the continuous-time model, there exists an equilibrium in which the attacker never waits before attacking if the vulnerability function satisfies*

$$\frac{V(t+a)}{V(t)} \geq e^{-2\lambda a} \tag{5}$$

for every $t \in [0, T]$ and $a \in [0, T - t]$.

Proof. Suppose that the attacker never waits. Let us consider the defender's best response to this strategy. Simplifying Equation(1), the defender's utility function

becomes

$$-\int_0^T \left(d(t) + V(t)\frac{R}{d(t)} \right) dt.$$

This utility is maximized by choosing $d(t)$ at each time t to minimize the cost plus risk. Setting

$$\frac{d}{dx}\left(x + V(t)\frac{R}{x} \right) = 0$$

and solving for x, we obtain the optimal $d(t)$ as

$$d(t) = \sqrt{V(t)R}. \tag{6}$$

Now, the part of the equilibrium condition that says $a(t) = 0$ is the attacker's best response function implies that for every t and a, we have

$$\frac{V(t)R}{d(t)} \geq \frac{V(t)Re^{-\lambda a}}{d(t+a)}.$$

Incorporating the defender's strategy and simplifying, we obtain

$$\begin{aligned} \frac{d(t+a)}{d(t)} &\geq e^{-\lambda a} \\ \frac{\sqrt{V(t+a)R}}{\sqrt{V(t)R}} &\geq e^{-\lambda a} \\ \frac{V(t+a)}{V(t)} &\geq e^{-2\lambda a}. \end{aligned}$$

Now conversely, suppose that

$$\frac{V(t+a)}{V(t)} \geq e^{-2\lambda a}.$$

Let $d(t) = \sqrt{V(t)R}$ be the defender's investment strategy. Because the sequence of inequalities above is reversible, we have that $a(t)$ is a best response to $d(t)$; and we have already showed that $d(t)$ is a best response to $a(t)$. So there exists an equilibrium in which the attacker never waits. □

The following proposition gives an analogous result for the discrete-time model.

Proposition 4. *In the discrete-time model, there is an equilibrium in which the attacker never waits before attacking if the vulnerability function satisfies*

$$\frac{V(k+a)}{V(k)} \geq (1-\lambda)^{2a} \tag{7}$$

for every $k \in \{0, \ldots, K-1\}$ *and* $a \in \{1, \ldots, K-k\}$.

Proof. Suppose that the attacker never waits. Let us consider the defender's best response to this strategy. Simplifying Equation(3), the defender's utility function becomes

$$-\sum_{k=0}^{K}\left(d(k)+V(k)\frac{R}{d(k)}\right).$$

This utility is maximized by choosing $d(k)$ at each step k to minimize the cost plus risk, giving

$$d(k)=\sqrt{V(k)R}. \tag{8}$$

To say that $a(k)=0$ is the attacker's best response function now implies that for every k and a, we have

$$\frac{V(k)R}{d(k)}\geq\frac{V(k)R(1-\lambda)^a}{d(k+a)}.$$

Incorporating the defender's strategy and simplifying, we obtain

$$\begin{aligned}\frac{d(k+a)}{d(k)}&\geq(1-\lambda)^a\\ \frac{\sqrt{V(k+a)R}}{\sqrt{V(k)R}}&\geq(1-\lambda)^a\\ \frac{V(k+a)}{V(k)}&\geq(1-\lambda)^{2a}.\end{aligned}$$

The argument that the condition implies existence of an equilibrium is analogous to the continuous version. □

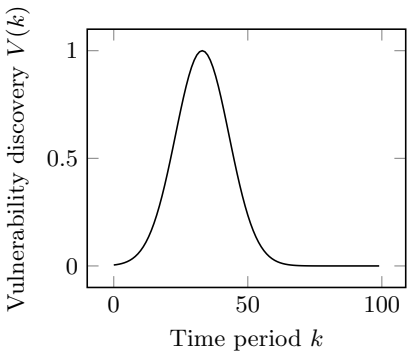
5 Numerical Examples

In this section, we present numerical examples to illustrate our model and our theoretical results, focusing on the vulnerability-discovery function and the defender's equilibrium strategy. For these numerical examples, we use the discrete-time version of our model.

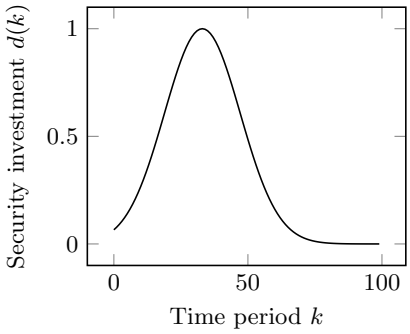
First, in Figures 1 and 2, we study two example vulnerability functions with the corresponding equilibrium defense strategies. In the first example (Figure 1), the vulnerability discovery rate grows and decays exponentially. More formally, the vulnerability discovery rate in this example is given by the following formula:

$$V(k)=e^{-\frac{(k-33)^2}{200}}. \tag{9}$$

In the second example (Figure 2), the vulnerability discovery rate grows and decays linearly (i.e., according to an affine function). In both cases, we let $R=1$, $K=100$, and $\lambda=0.3$.

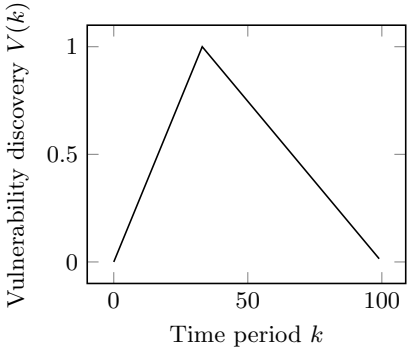


(a) Vulnerability discovery rate as a function of time

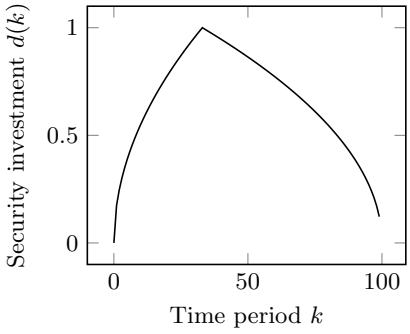


(b) Defender's equilibrium security investment as a function of time

Fig. 1: Example based on exponentially growing and decaying vulnerability discovery rate with the corresponding equilibrium defense strategy.



(a) Vulnerability discovery rate as a function of time



(b) Defender's equilibrium security investment as a function of time

Fig. 2: Example based on linearly growing and decaying vulnerability discovery rate with the corresponding equilibrium defense strategy.

We can see that, in both examples, the rise and fall of the defender's security investment is dampened compared to those of the vulnerability functions. However, the security investments are very far from being constant, which indicates that dynamic environments play an important role in determining equilibrium investments.

Second, in Figure 3, we study the condition given by Proposition 4. Recall that Proposition 4 establishes a threshold on the maximum rate of decrease in vulnerability discovery such that the attacker never waiting is an equilibrium. In

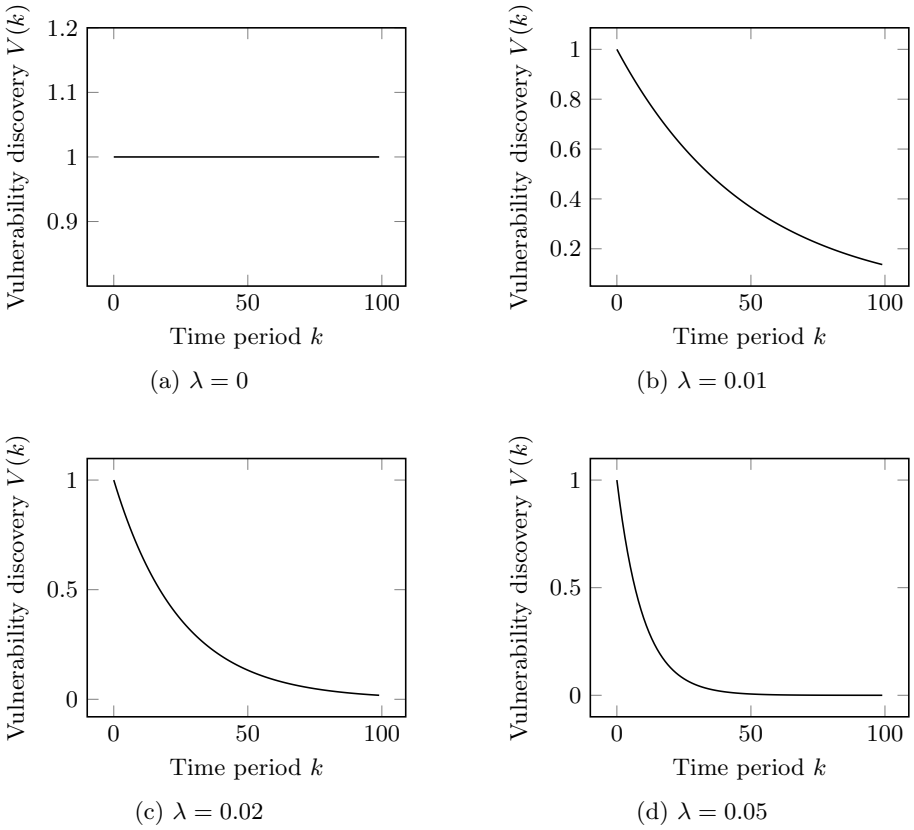


Fig. 3: Threshold vulnerability functions $V(k)$ for Proposition 4 with various values of λ.

Figure 3, for various values of λ, we plot vulnerability discovery functions that decrease with this maximum rate.

Firstly, in Figure 3(a), we can see that if $\lambda = 0$, then the vulnerability discovery rate has to be constant in order for the attacker not waiting to be an equilibrium. The explanation for this corner case is that $\lambda = 0$ means that the attacker can stockpile vulnerabilities without taking any risk; hence, the attacker will wait only if security investments are constant over time, which implies that the vulnerability discovery rate must also be constant for the equilibrium to exist. Secondly, in Figures 3(b), 3(c), and 3(d), we see that the higher the value of λ, the more steeply the vulnerability discovery rate may decrease. Again, the explanation for this is that higher values of λ mean higher risk for stockpiling vulnerabilities; hence, the higher λ is, the more steeply the discovery rate can decrease without the attacker opting to wait.

6 Conclusion

The recent rise of attacks involving a high degree of stealthiness has sparked considerable interest in games of timing for security. However, to the best of our knowledge, the previously proposed models in the recent literature share a common limitation: the assumption that the cost and effectiveness of the attackers' actions are time-independent. In this paper, we proposed and studied a model which captures dynamic environments, i.e., in which the attackers' actions depend on the availability of exploitable vulnerabilities. More specifically, we assumed that attackers discover vulnerabilities according to a given vulnerability-discovery process, which we modeled as an arbitrary function of time. Based on this assumption, we formulated a two-player game of timing between a defender, who tries to protect a service or resource through security investments, and an attacker, who can choose when to exploit a vulnerability. The most interesting novel feature of our model is the attacker's dilemma: whether to wait in hope of exploiting the vulnerability at a time when security is lower, but risking that the vulnerability is rediscovered and fixed in the meantime.

In our theoretical analysis, we primarily focused on characterizing equilibria in which the attacker does not stockpile vulnerabilities (i.e., never waits to exploit a vulnerability). The question of vulnerability stockpiling is interesting in many practical scenarios, most importantly in the case of software products that are widely used even after their end of official support. Our results relate the vulnerability discovery process to the rate of repairing vulnerabilities, and hence provide guidelines for finding vulnerability repair rates that will not lead to a vulnerability stockpiling equilibrium in practice. In our numerical examples, we considered multiple specific vulnerability functions, and studied the corresponding equilibrium strategies.

There are multiple directions for extending our current work. Firstly, we plan to provide a theoretical characterization of the game's equilibria in the case when the attacker does not stockpile vulnerabilities (i.e., when never waiting is not an equilibrium). Secondly, we plan to study and characterize the Stackelberg equilibria of our game. In our current work, we assume that the defender and the attacker choose their strategies at the same time, which captures scenarios with uninformed players. However, in [17], it was shown – for a different timing-game model – that a defender can substantially decrease its losses by publicly committing to a strategy and letting the attacker choose its strategy in response. We expect that a similar result holds for the model presented in this paper as well.

Acknowledgment

We thank the anonymous reviewers for their helpful comments. This work was supported in part by the National Science Foundation (CNS-1238959).

References

1. Blackwell, D.: The noisy duel, one bullet each, arbitrary accuracy. Tech. rep., The RAND Corporation, D-442 (1949)
2. Böhme, R., Schwartz, G.: Modeling cyber-insurance: Towards a unifying framework. In: 9th Workshop on the Economics of Information Security (WEIS) (2010)
3. Bowers, K., Van Dijk, M., Griffin, R., Juels, A., Oprea, A., Rivest, R., Triandopoulos, N.: Defending against the unknown enemy: Applying FlipIt to system security. In: Proceedings of the 3rd Conference on Decision and Game Theory for Security (GameSec), pp. 248–263. Springer (2012)
4. Chen, P., Kataria, G., Krishnan, R.: Correlated failures, diversification, and information security risk management. MIS Quarterly 35(2), 397–422 (Jun 2011)
5. Feng, X., Zheng, Z., Hu, P., Cansever, D., Mohapatra, P.: Stealthy attacks meets insider threats: A three-player game model. Tech. rep.
6. Fultz, N., Grossklags, J.: Blue versus red: Towards a model of distributed security attacks. In: Dingledine, R., Golle, P. (eds.) Financial Cryptography and Data Security, Lecture Notes in Computer Science, vol. 5628, pp. 167–183. Springer (2009)
7. Gordon, L., Loeb, M.: The economics of information security investment. ACM Transactions on Information and System Security 5(4), 438–457 (Nov 2002)
8. Grossklags, J., Christin, N., Chuang, J.: Secure or insure?: A game-theoretic analysis of information security games. In: Proceedings of the 17th International World Wide Web Conference. pp. 209–218 (2008)
9. Grossklags, J., Reitter, D.: How task familiarity and cognitive predispositions impact behavior in a security game of timing. In: Proceedings of the 27th IEEE Computer Security Foundations Symposium (CSF). pp. 111–122 (2014)
10. Hu, P., Li, H., Fu, H., Cansever, D., Mohapatra, P.: Dynamic defense strategy against advanced persistent threat with insiders. In: Proceedings of the 34th IEEE International Conference on Computer Communications (INFOCOM) (2015)
11. Ioannidis, C., Pym, D., Williams, J.: Investments and trade-offs in the economics of information security. In: Proceedings of the 13th International Conference on Financial Cryptography and Data Security, pp. 148–166. Springer (2009)
12. Johnson, B., Böhme, R., Grossklags, J.: Security games with market insurance. In: Baras, J., Katz, J., Altman, E. (eds.) Decision and Game Theory for Security, Lecture Notes in Computer Science, vol. 7037, pp. 117–130. Springer (2011)
13. Johnson, B., Laszka, A., Grossklags, J.: The complexity of estimating systematic risk in networks. In: Proceedings of the 27th IEEE Computer Security Foundations Symposium (CSF). pp. 325–336 (2014)
14. Kunreuther, H., Heal, G.: Interdependent security. Journal of Risk and Uncertainty 26(2), 231–249 (2003)
15. Laszka, A., Felegyhazi, M., Buttyan, L.: A survey of interdependent information security games. ACM Computing Surveys 47(2), 23:1–23:38 (Aug 2014)
16. Laszka, A., Horvath, G., Felegyhazi, M., Buttyán, L.: FlipThem: Modeling targeted attacks with FlipIt for multiple resources. In: Proceedings of the 5th Conference on Decision and Game Theory for Security (GameSec), pp. 175–194. Springer (2014)
17. Laszka, A., Johnson, B., Grossklags, J.: Mitigating covert compromises. In: Proceedings of the 9th Conference on Web and Internet Economics (WINE), pp. 319–332. Springer (2013)
18. Laszka, A., Johnson, B., Grossklags, J.: Mitigation of targeted and non-targeted covert attacks as a timing game. In: Proceedings of the 4th Conference on Decision and Game Theory for Security (GameSec), pp. 175–191. Springer (2013)

19. Lelarge, M., Bolot, J.: Economic incentives to increase security in the internet: The case for insurance. In: Proceedings of the 33rd IEEE International Conference on Computer Communications (INFOCOM). pp. 1494–1502 (2009)
20. Manshaei, M., Zhu, Q., Alpcan, T., Bacşar, T., Hubaux, J.: Game theory meets network security and privacy. ACM Computing Surveys 45(3), 25:1–25:39 (Jul 2013)
21. Nochenson, A., Grossklags, J.: A behavioral investigation of the FlipIt game. In: 12th Workshop on the Economics of Information Security (WEIS) (2013)
22. Ozment, A.: The likelihood of vulnerability rediscovery and the social utility of vulnerability hunting. In: Proceedings of the 4th Workshop on the Economics of Information Security (WEIS) (2005)
23. Ozment, A., Schechter, S.: Milk or wine: Does software security improve with age? In: Proceedings of the 15th USENIX Security Symposium (2006)
24. Pal, R., Huang, X., Zhang, Y., Natarajan, S., Hui, P.: On security monitoring in sdns: A strategic outlook
25. Panaousis, E., Fielder, A., Malacaria, P., Hankin, C., Smeraldi, F.: Cybersecurity games and investments: A decision support approach. In: Proceedings of the 5th Conference on Decision and Game Theory for Security (GameSec), pp. 266–286. Springer (2014)
26. Pham, V., Cid, C.: Are we compromised? Modelling security assessment games. In: Proceedings of the 3rd Conference on Decision and Game Theory for Security (GameSec), pp. 234–247. Springer (2012)
27. Radzik, T.: Results and problems in games of timing. Lecture Notes-Monograph Series, Statistics, Probability and Game Theory: Papers in Honor of David Blackwell 30, 269–292 (1996)
28. Reitter, D., Grossklags, J., Nochenson, A.: Risk-seeking in a continuous game of timing. In: Proceedings of the 13th International Conference on Cognitive Modeling (ICCM). pp. 397–403 (2013)
29. Rescorla, E.: Is finding security holes a good idea? IEEE Security & Privacy 3(1), 14–19 (January-February 2005)
30. Schechter, S., Smith, M.: How much security is enough to stop a thief? In: Wright, R. (ed.) Financial Cryptography, Lecture Notes in Computer Science, vol. 2742, pp. 122–137. Springer (2003)
31. Van Dijk, M., Juels, A., Oprea, A., Rivest, R.: Flipit: The game of "stealthy takeover". Journal of Cryptology 26(4), 655–713 (2013)
32. Varian, H.: System reliability and free riding. In: Camp, J., Lewis, S. (eds.) Economics of Information Security, pp. 1–15. Kluwer Academic Publishers, Dordrecht, The Netherlands (2004)
33. Wellman, M., Prakash, A.: Empirical game-theoretic analysis of an adaptive cyber-defense scenario (preliminary report). In: Proceedings of the 5th Conference on Decision and Game Theory for Security (GameSec), pp. 43–58. Springer (2014)
34. Zhang, M., Zheng, Z., Shroff, N.: Stealthy attacks and observable defenses: A game theoretic model under strict resource constraints. In: Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP). pp. 813–817
35. Zhao, M., Grossklags, J., Chen, K.: An exploratory study of white hat behaviors in a web vulnerability disclosure program. In: Proceedings of the ACM Workshop on Security Information Workers. pp. 51–58 (2014)
36. Zhao, M., Grossklags, J., Liu, P.: An empirical study of web vulnerability discovery ecosystems. In: Proceedings of the 22nd ACM Conference on Computer and Communications Security (CCS) (2015)