An Initial Study on Personalized Filtering Thresholds in Defending Sequential Spear Phishing Attacks

Mengchen Zhao[1], **Bo An**[1], **Christopher Kiekintveld**[2]
[1]School of Computer Engineering, Nanyang Technological University, Singapore 639798
[2]University of Texas at El Paso, El Paso, TX 79968
[1]zhao0204@e.ntu.edu.sg, boan@ntu.edu.sg
[2]cdkiekintveld@utep.edu

Abstract

Different from spam and regular phishing attacks, spear phishing attacks target a small group of people, and the attackers usually make elaborate plans before attacking. There is existing work on classifying spear phishing emails where a threshold value is used to balance misclassified normal emails and misclassified malicious emails. However, most existing systems use a uniform threshold for all users, while in reality users may differ in how susceptible they are to phishing attacks and their access to critical information. Existing work on setting personalized thresholds assumes that the attacker compromises multiple users simultaneously to maximize his expected utility. However, an attacker may be only interested in specific credential information, which could be accessed by a group of users. In this situation, a sequential attack is more reasonable for the attacker to reduce the cost of launching attacks and the likelihood of detection. We propose a Stackelberg game model to calculate the optimal solution for the sequential attack situation and formulate a bilevel optimization problem for the defender. By exploiting the structure of the bilevel problem, we propose a single level formulation called PEDS that is equivalent to the bilevel problem. Experimental results show that PEDS can solved within 60 seconds even when the number of users is 70, and the thresholds computed by PEDS lead to significant higher defender utilities as compared with existing approaches.

1 Introduction

Traditional cyber attacks via emails are mainly spam and regular phishing attacks, where the attackers send similar malicious emails to a large number of users with low costs. However, such attacks rarely succeed due to the improvement of detection techniques and people's anti-phishing awareness. In recent years, as an important part of Advanced Persistent Threat (APT) [TrendLabs, 2012], spear phishing attacks have caused huge losses. Spear phishers usually target a small group of people, in consideration of their susceptibilities, vulnerabilities and confidentiality levels. For example, in 2011, RSA, a company who sells security solutions, was breached by a spear phishing attack [Zetter, 2011]. The attacker targeted a small group of RSA employees and sent them emails titled "2011 Recruitment Plan" with a malicious Excel attachment. When the trojan was downloaded, the attackers harvested confidential information and made their way up the RSA food-chain via both IT and non-IT personnel accounts, until they finally obtained privileged access to the targeted system. The targeted data and files were stolen and sent to an external compromised machine at a hosting provider. As a result, the intruders succeeded in stealing information related to the company's SecurID two-factor authentication products, putting some RSA clients at risk.

To defend against spear phishing attacks, besides improving users' anti-phishing awareness, blocking malicious emails from reaching users is very important. One typical way is to employ a filter to block spear phishing emails before they reach the users. A filter scores every incoming email according to their likelihood of being malicious emails [Bergholz et al., 2010]. Emails with scores higher than a pre-specified threshold will be filtered. The defender faces a tradeoff while setting the threshold [Sheng et al., 2009]. If the threshold is too high, malicious emails will easily pass the filtering system. Conversely, if the threshold is too low, normal emails will be filtered. Recent work has shown that the effectiveness of filtering can be improved if thresholds are personalized according to individuals' values and susceptibilities [Laszka et al., 2015]. In this paper, we consider situations where an attacker compromises a set of users, who could possibly access a specific credential or data of interest to the attacker. In such situations, a sequential attack is preferable to the attacker since he does not benefit more by compromising more users while he sustains higher costs (e.g., crafting phishing emails, investigating the users) when attacking more users.

There is a lot of work on classifying spear phishing emails [Higbee et al., 2014; Deshmukh et al., 2014]. For example, based on a random forest algorithm, an accuracy

of 97.76% [1] in identifying spear phishing emails with only 2% false positive rate can be achieved [Dewan et al., 2014]. For the personalized thresholds setting, a game-theoretic model is proposed [Laszka et al., 2015], where an attacker attacks a carefully chosen subgroup of users. However, in the situation where the attacker launches sequential attacks to get a credential, their algorithm fails to find the optimal defender strategy since the algorithm assumes that the attacker always gets a higher utility by compromising more users and ignores the costs of attacking.

In this paper, we first propose a Stackelberg game model extending the existing security literature [Korzhyk et al., 2011; An et al., 2011; Shieh et al., 2012; Tambe et al., 2014; Gan et al., ; Yin et al., 2014; Yin et al., 2015; An et al., 2013] to the case with sequential attacks. We model the attacker's decision making as a Markov Decision Process and formulate a bilevel optimization problem for the defender, where the attacker's best response, as a constraint of the bilevel problem, is characterized by a linear program. The main challenge of solving this problem is that the probability of losing the credential, as a part of the upper level objective, is a complex function and is implicitly defined. To address this challenge, we exploit the structure of the MDP and find an equivalent representation of the probability of losing the credential, which enables us to reduce the bilevel problem into a single level problem. We also show that the optimal policy for the attacker is not heuristic, i.e., the attacker does not always attack the user with the highest immediate payoff. Additionally, an attacker policy yields a sequence of users to be attacked. However, we show that the defender's utility does not depend on the order of the users in the sequence. Experimental results show that our algorithm can solve games with 70 users in only 60 seconds, and the thresholds computed by our algorithm lead to significant higher defender utility as compared with existing approaches.

2 Model

There is a defender and an attacker in the spear phishing security game. The defender (e.g., an organization) has a *credential* that could possibly be accessed by a set of users $U = \{1, 2, ..., |U|\}$. The attacker, wanting to gain access to the credential[2], sends spear phishing emails to the users based on an attack plan that he calculate. When making the attack plan, the attacker takes into account the susceptibility, confidentiality level of the users, as well as the cost of attacking. We denote by a_u the susceptibility of user u, meaning that u will be compromised with probability a_u given that a spear phishing email is delivered to her [3]. We

[1] Although the accuracy seems satisfying, however, if the attacker launches persistent attacks (e.g., [Varma, 2010]), the success rate of the attack is considerable.

[2] We use the generic term "credential" here to mean any critical data or access privilege that the attacker is seeking to gain.

[3] a_u can be measured by sending probe emails to the users [Sheng et al., 2010], [Kelley, 2010] [Jagatic et al., 2007]

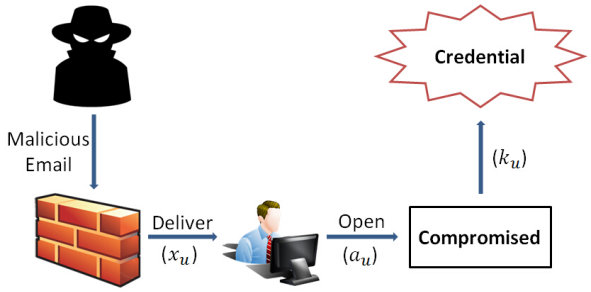

Figure 1: Spear Phishing Attack Flow

denote by k_u the confidentiality level of user u[4], meaning that user u has access to the credential with probability k_u. The attacker sustains some costs when launching attacks, such as crafting phishing emails, investigating users, writing malware, as well as the risk of alarming the organization. We denote by c_u the cost of attacking user u.

While receiving emails, the filter first scores them according to their likelihood of being malicious emails, and then delivers only those with scores lower than a given threshold. In such a way, it is possible that malicious emails are misclassified as normal ones. We call such misclassifications *false negatives*. On the other side, some normal emails might be misclassified as malicious ones. We call such misclassifications *false positives*. We ignore ordinary phishing and spam emails in our model since they are usually much less harmful than spear phishing emails, but note that they can be handled using similar mechanisms. A threshold for a user is a parameter set by the defender that determines a probability of false negatives (denoted as x_u), and a probability of false positives (denoted as y_u). The relationship between x_u and y_u can be characterized by a function $\phi : [0, 1] \to [0, 1]$, $y_u = \phi(x_u)$ [Laszka et al., 2015]. By adjusting the thresholds, the organization can determine a pair (x_u, y_u) for each user. Note that ϕ is a property of the email classifier in the filter and is always non-increasing.

There are three steps for a spear phishing attack to succeed. First, a spear phishing email towards user u passes the email filter and hence is delivered to u. The success rate of this step is the false negative probability x_u. Second, user u is deceived and compromised, probably by clicking on a malicious link or an attachment of the email. The success rate of this step is a_u. We allow the attacker to attack the same user multiple times if either of the above two steps fails. We assume that if a user is compromised, her private information can be harvested by the attacker. Last, the attacker harvests the private information of user u to see if she holds the credential. The success rate of this step is k_u. Figure 1 shows the flow of such an attack.

[4] In many cases, both the security team of the organization and the attacker have knowledge about the users' confidentiality levels. However, they may not know for sure which users hold the credential.

2.1 Stackelberg Game

We model the interaction between the defender and the attacker as a Stackelberg game. The defender moves first by choosing a false negative probability vector $\mathbf{x}$. After observing $\mathbf{x}$[5], the attacker determines a policy π and launches attacks following it. Due to Observation 1, without loss of generality, we consider only attacker policies that exclude concurrent attacks.

Observation 1. *A concurrent attack can be modeled as an equivalent sequential attack.*

A concurrent attack towards n users means that the attacker sends a spear phishing email to each of the users concurrently. The same result can be achieved in our model if the attacker sequentially sends spear phishing emails to these users. Therefore, sequential attacks are more general than concurrent attacks. Actually, a sequential attack gives the attacker more flexibility than a concurrent attack, i.e., the attacker can attack the same user multiple times, and stop attacking whenever he gets the credential.

In the Stackelberg game, the follower (attacker) plays a best response to the defender's strategy $\mathbf{x}$ that maximizes his expected utility. We denote by $\pi_{\mathbf{x}}$ the attacker's optimal policy given $\mathbf{x}$. The defender's expected utility consists of the expected loss from both the false negatives and the false positives. In terms of the false positives, we denote by h_u the expected loss of not delivering normal emails sent to user u. Then the total loss the defender sustains from the false positives is $\sum_{u\in U}\phi(x_u)h_u$. The loss caused by the false negatives is the expected loss of losing the credential. We denote by $\theta(\mathbf{x},\pi_{\mathbf{x}})$ the probability that the attacker gets the credential given the strategy profile $(\mathbf{x},\pi_{\mathbf{x}})$, and by L the expected value of the credential. We assume that if the attacker gets the credential, he gains a value L and the defender losses a value L. Then the defender's expected utility can be represented by:

$$P_d(\mathbf{x},\pi_{\mathbf{x}}) = -\theta(\mathbf{x},\pi_{\mathbf{x}})L - \sum_{u\in U}\phi(x_u)h_u.$$

We denote by $P_a(\mathbf{x},\pi_{\mathbf{x}})$ the attacker's expected utility. For each attack, the attacker suffers a cost c_u. If the attack succeeds (i.e., the attacker gets the credential), he gains a value L. Otherwise he gains nothing. We will show how to compute $P_a(\mathbf{x},\pi_{\mathbf{x}})$ in Section 3.

We consider the Stackelberg equilibrium as our solution concept. In Section 4, by Theorem 1, we show that the defender's expected utility remains the same no matter how the attacker breaks ties among multiple optimal policies.

Definition 1. *Stackelberg equilibrium strategy: If a strategy profile $(\mathbf{x}^*,\pi_{\mathbf{x}^*})$ such that $P_d(\mathbf{x}^*,\pi_{\mathbf{x}^*}) \geq P_d(\mathbf{x},\pi_{\mathbf{x}})$ holds for any possible $\mathbf{x}$, it is a Stackelberg equilibrium strategy profile.*

[5]We assume that $\mathbf{x}$ is observable since in many cases the spear phishers can exploit the vulnerabilities of user endpoints [Choo, 2011]. Therefore, they may obtain the information about users' protection levels, such as threshold values.

3 Optimal Attacker Policy

We model the attacker's decision making problem as a Markov Decision Process (MDP), which can be represented as a tuple $(\mathcal{S},\mathcal{A},T,R,\pi)$. $\mathcal{S} = \{s|s\subseteq U, s\neq\emptyset\}\cup\{s^n,s^y\}$ is the state set that consists of *normal states* and two *terminal states* s^n, s^y. A normal state is a non-empty subset of the user set U that represents the users who have not been compromised by the attacker. Specially, the initial state $s_0 = U$. The terminal state s^n includes two kinds of situations: (1) The attacker chooses to stop attacking without getting the credential even though there are still some users that have not been compromised. (2) All the users have been compromised but none of them holds the credential. s^y represents the state where the attacker gets the credential. $\mathcal{A} = \{a|a = u\in U \text{ or } a = 0\}$ is the set of attacker's actions where $a = u$ means that the attacker chooses to attack user u, and $a = 0$ means that the attacker chooses to stop attacking. Moreover, we denote by $\mathcal{A}^s = \{a|a = u\in s \text{ or } a = 0\}$ the attacker's action set at state s, in the sense that the attacker only attacks the users that have not been compromised. $T : \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$ is the transition function where $T(s,a,s')$ represents the probability that s transitions to s' by executing action a. $R : \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$ is the reward function where $R(s,a,s')$ represents the attacker's reward when s transitions to s' by executing action a. $\pi : \mathcal{S}\to\mathcal{A}$ is the policy function that specifies an action at each state.

Now we define T and R. At any non-terminal state s, if the attacker chooses to stop attacking, s transitions to s^n with probability $T(s,a{=}0,s^n){=}1$, and with reward $R(s,a{=}0,s^n){=}0$. If the attacker chooses to attack user u, there are three possible transitions: (1) If the malicious email fails to pass the filter or it passes the filter but fails to compromise user u, then s transitions to itself. (2) If u is compromised but she does not hold the credential , s transitions to $s^{-u} = s\setminus\{u\}$. In the special case where $s = \{u\}$ has only one user, s^{-u} means that none of the users hold the credential, therefore $s^{-u} = s^n$. (3) If u is compromised and the attacker gets the credential via u, s transitions to the terminal state s^y. The transition function can be summarized as follows.

$$T(s,a,s') = \begin{cases} 1, \text{ if } a = 0, s' = s^n; \\ 1 - x_u a_u, \text{ if } a = u\in\mathcal{A}^s, s' = s; \\ a_u x_u(1-k_u), \text{ if } a = u\in\mathcal{A}^s, s' = s^{-u} \\ x_u a_u k_u, \text{ if } a = u\in\mathcal{A}^s, s' = s^y. \end{cases}$$

The reward function R can be summarized as follows.

$$R(s,a,s') = \begin{cases} 0, \text{ if } a = 0, s' = s^n; \\ -c_u, \text{ if } a = u\in\mathcal{A}^s, s' = s \text{ or } s^{-u} \\ -c_u + L, \text{ if } a = u\in\mathcal{A}^s, s' = s^y. \end{cases}$$

We denote by V^π the *value function*. $V^\pi(s)$ represents the attacker's expected utility when his current state is s and he fol-

lows a policy π afterwards. We denote by V^* the value function when the attacker follows the optimal policy $\pi_{\mathbf{x}}$. $V^*(s_0)$ is the attacker's maximum expected utility at initial state s_0, i.e.,

$$P_a(\mathbf{x}, \pi_{\mathbf{x}}) = V^*(s_0).$$

According to *Bellman Equations* [Bellman, 1961], $V^*(s)$ satisfies:

$$V^*(s) = \sum_{s' \in \mathcal{S}} T(s, \pi_{\mathbf{x}}(s), s')[R(s, \pi_{\mathbf{x}}(s), s') + V^*(s')].$$

Substituting T and R with their specific forms defined above, we have: $V^*(s) = 0$ if $\pi_{\mathbf{x}}(s) = 0$; otherwise

$$\begin{aligned} V^*(s) = &(1 - a_u x_u)(V^*(s) - c_u) + a_u x_u k_u (L - c_u) \\ &+ a_u x_u (1 - k_u)(V^*(s^{-u}) - c_u) \end{aligned} \quad (1)$$

where $u = \pi_{\mathbf{x}}(s)$.

3.1 Solving the MDP

We solve the MDP using linear programming (LP) [Schweitzer and Seidmann, 1985]. Specifically, $V^*(s)$ $(s \in \mathcal{S})$ are the solutions of the following LP.

$$\begin{aligned} \min_{\mathbf{V}^*} \quad & \sum_{s \in \mathcal{S}} V^*(s), \\ \text{s.t.} \quad & V^*(s) \geq \sum_{s' \in \mathcal{S}} T(s, a, s') \left[R(s, a, s') + V^*(s')\right], \\ & \forall a \in \mathcal{A}^s, \forall s \in \mathcal{S}. \end{aligned}$$

The optimal policy $\pi_{\mathbf{x}}$ can then be obtained by:

$$\pi_{\mathbf{x}}(s) = \arg\max_{a \in \mathcal{A}^s} Q(s, a), \forall s \in \mathcal{S},$$

where

$$Q(s, a) = \sum_{s' \in \mathcal{S}} T(s, a, s') \left[R(s, a, s') + V^*(s')\right], \forall s \in \mathcal{S}.$$

We assume that if $a'{=}u$ and $a''{=}0 \in \arg\max_{a \in \mathcal{A}^s} Q(s, a)$, then $\pi_{\mathbf{x}}(s) = a''$. Intuitively, it means that the attacker prefers stopping attack rather than launching another attack if there is a tie.

3.2 An Example

We given an simple example of the attacker's decision making problem and show that the optimal policy is not heuristic. Assume $U = \{u_1, u_2\}$, $\mathbf{c} = \{5, 60\}$, $\mathbf{a} = \{1, 1\}$, $\mathbf{x} = \{0.4, 0.9\}$, $\mathbf{k} = \{0.4, 0.9\}$ and $L = 100$. Then there are 5 possible states: $\mathcal{S} = \{s_0, s_1, s_2, s^n, s^y\}$ where $s_0 = \{u_1, u_2\}$, $s_1 = \{u_1\}$, $s_2 = \{u_2\}$. By solving the MDP, we obtain the results as follows:

$V^*(s_0) = 41.5$, $\pi_{\mathbf{x}}(s_0) = \{u_1\}$,
$V^*(s_1) = 27.5$, $\pi_{\mathbf{x}}(s_1) = \{u_1\}$,
$V^*(s_2) = 23.3$, $\pi_{\mathbf{x}}(s_2) = \{u_2\}$,
$V^*(s^n) = 0$, $V^*(s^y) = 0$.

From the results we can see that in the initial state s_0 the attacker's optimal policy is to attack user u_1. However, the expected immediate reward of attacking u_1 is $E(u_1) = La_1x_1k_1 - c_1 = 100 \times 0.4 \times 0.4 - 5 = 11$, which is less than $E(u_2) = La_2x_2k_2 - c_2 = 100 \times 0.9 \times 0.9 - 60 = 21$. In other words the attacker's optimal policy is not, as a heuristic, attacking the user with the highest immediate reward.

4 Optimal Defender Srategy

In this section, we first formulate the defender's problem as a bilevel optimization problem. Then, by exploiting the structure of the MDP, we propose two lemmas that give an equivalent and explicit form of $\theta(\mathbf{x}, \pi_{\mathbf{x}})$. Then we show that the attacker can break ties arbitrarily without affecting the defender's expected utility. Finally, based on the result of Theorem 2, we propose a formulation called PEDS, which is a single level problem and is equivalent to the bilevel problem of the defender.

4.1 Bilevel Formulation

By adjusting the thresholds, the defender seeks the optimal false negative probability vector $\mathbf{x}^*$ that maximizes her utility. Solving the defender's problem is equivalent to solving the following bilevel optimization problem (denoted as Problem 1):

$$\begin{aligned} \max_{\mathbf{x}} \quad & -\theta(\mathbf{x}, \pi_{\mathbf{x}})L - \sum_{u \in U} \phi(x_u) h_u, \\ \text{s.t.} \quad & x_u \in [0, 1], \ \forall u \in U, \\ & Q(s, \pi_{\mathbf{x}}(s)) \geq Q(s, a), \forall a \in \mathcal{A}^s, \forall s \in \mathcal{S}, \\ & \min_{\mathbf{V}^*} \sum_{s \in \mathcal{S}} V^*(s), \\ & \text{s.t.} \quad V^*(s) \geq \sum_{s' \in \mathcal{S}} T(s, a, s') \left[R(s, a, s') + V^*(s')\right], \\ & \qquad \forall a \in \mathcal{A}^s, \forall s \in \mathcal{S}, \end{aligned}$$

where $Q(s, a) = \sum_{s' \in \mathcal{S}} T(s, a, s') \left[R(s, a, s') + V^*(s')\right]$.

The lower level LP computes the optimal values of all of the states. The second constraint in the upper level, which is equivalent to $\pi_{\mathbf{x}} = \arg\max_{a \in \mathcal{A}^s} Q(s, a)$, restricts $\pi_{\mathbf{x}}$ to be the optimal policy. The challenge of solving Problem 1 is that $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ is implicitly defined. In fact, $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ is the probability that the attacker ends in the terminal state s^y given he follows the optimal policy $\pi_{\mathbf{x}}$. Lemma 1 and Lemma 2 show that $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ has an equivalent, explicit form.

Before introducing Lemma 1 and Lemma 2, we need to introduce the concepts of *reachable states* and *potential target set*. Once a policy is determined, the MDP is reduced to a Markov chain where only part of the states are connected if we consider the Markov chain as a graph. For example, if $s_0{=}\{u_1, u_2\}$ and $\pi(s_0){=}u_1$, then state $s{=}\{u_1\}$ is not

connected. We define reachable states as the states that are connected in the Markov chain. We denote by $\Delta(\pi)$ the set of reachable states given the policy π. A policy π projects each reachable state $s \in \Delta(\pi)$ to an action $a = u \in \mathcal{A}^s$ or $a = 0$. We denote by $\Gamma(\pi)$ the potential target set, which is the set of users that are projected from the reachable states under the policy π, i.e., $\Gamma(\pi) = \{u \in U | u = \pi(s), s \in \Delta(\pi)\}$. While attacking, the attacker may not go through all the reachable states and may get the credential at some reachable state. However, if the attacker successfully compromises all the users but none of them holds the credential, he will go through all the reachable states except s^y.

The reachable states set under the optimal policy $\pi_{\mathbf{x}}$ can be represented as $\Delta(\pi_{\mathbf{x}}) = \{s_0, s_1, ..., s_r\} \cup \{s^n, s^y\}$ where $s_{i+1} = s_i^{-u}$, $u = \pi_{\mathbf{x}}(s_i)$. Lemma 1 gives a necessary and sufficient condition for which users that should be in $\Gamma(\pi_{\mathbf{x}})$. Lemma 2 shows how $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ is computed based on the result of Lemma 1. Here we only give the sketches of the proofs. See Appendix for detailed proofs. Theorem 1 and Theorem 2 are directly derived from the two lemmas.

Lemma 1. *$u \in \Gamma(\pi_{\mathbf{x}})$ if and only if $x_u > \frac{c_u}{La_u k_u}$.*

Proof. (Sketch.) For the "if direction", we first prove that the attacker prefers to attack the a user who satisfies $x_u > \frac{c_u}{La_u k_u}$ than to stop attacking. Then based on the structure of $\Delta(\pi_{\mathbf{x}})$ we know that if u is not attacked at s_r, it should be attacked at some state s_i where $i < r$. For the "only if" direction, if $u \in \Gamma(\pi_{\mathbf{x}})$, then there is a state $s \in \Delta(\pi_{\mathbf{x}})$ such that $\pi_{\mathbf{x}}(s) = u$. Then by analysing the structure of the optimal value function $V^*(s)$, we can derive the result $x_u > \frac{c_u}{La_u k_u}$. □

Lemma 2. *$\theta(\mathbf{x}, \pi_{\mathbf{x}}) = 1 - \prod_{u \in \Gamma(\pi_{\mathbf{x}})}(1 - k_u)$.*

Proof. (Sketch.) As we mentioned before, the MDP is reduced to a Markov chain given the optimal policy $\pi_{\mathbf{x}}$. The Markov chain has two absorbing states s^n and s^y. $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ is actually the probability that the attacker ends in s^y. Based on the result of Lemma 1, we know which users are in the potential target set and can specify the transition probability matrix of the Markov chain. Hence we can directly compute $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ based on existing formulas [Grassmann et al., 1985]. □

Theorem 1. *The defender's expected utility remains the same no matter how the attacker breaks ties among multiple optimal policies.*

Proof. Recall that the defender's utility function is

$$P_d(\mathbf{x}, \pi_{\mathbf{x}}) = -\theta(\mathbf{x}, \pi_{\mathbf{x}})L - \sum_{u \in U} \phi(x_u)h_u.$$

Using the result of Lemma 1, $\Gamma(\pi_{\mathbf{x}})$ can be represented as $\{u \in U | x_u > \frac{c_u}{La_u k_u}\}$, then $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ can be represented as

$$\theta(\mathbf{x}, \pi_{\mathbf{x}}) = 1 - \prod_{u \in \{u' \in U | x_{u'} > \frac{c_{u'}}{La_{u'} k_{u'}}\}} (1 - k_u).$$

For any other optimal policy $\pi'_{\mathbf{x}}$, we have

$$\theta(\mathbf{x}, \pi'_{\mathbf{x}}) = 1 - \prod_{u \in \{u' \in U | x_{u'} > \frac{c_{u'}}{La_{u'} k_{u'}}\}} (1 - k_u).$$

Therefore, $\theta(\mathbf{x}, \pi_{\mathbf{x}}) = \theta(\mathbf{x}, \pi_{\mathbf{x}})'$. The defender's expected utility will be the same when the attacker chooses any other optimal policy. □

Theorem 2. *For any user $u \in U$, if $\frac{c_u}{La_u k_u} \geq 1$, the optimal false negative probability $x_u^* = 1$. Otherwise, $x_u^* \in \{\frac{c_u}{La_u k_u}, 1\}$.*

Proof. In the case that $\frac{c_u}{La_u k_u} \geq 1$, based on Lemma 1 and the fact that $x_u \in [0, 1]$, we have $u \notin \Gamma(\pi_{\mathbf{x}})$. Then for any $x_u \in [0, 1]$, according to Lemma 2, $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ remains a constant value. Note that $\phi(x_u)$ is non-increasing with respect to x_u. Therefore, for any $x_u \in [0, 1]$, we can always increase the objective value $P_d = -\theta(\mathbf{x}, \pi_{\mathbf{x}})L - \sum_{u \in U} \phi(x_u)h_u$ by replacing x_u with 1.

In the case that $\frac{c_u}{La_u k_u} \in [0, 1]$, similarly, for any $x_u \in [0, \frac{c_u}{La_u k_u}]$, we have $u \notin \Gamma(\pi_{\mathbf{x}})$ and $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ remains constant. Therefore, for any $x_u \in [0, \frac{c_u}{La_u k_u})$, we can always increase the objective value P_d by replacing x_u with $\frac{c_u}{La_u k_u}$. And for any $x_u \in (\frac{c_u}{La_u k_u}, 1]$, we have $u \in \Gamma(\pi_{\mathbf{x}})$ and $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ remains constant. Therefore, for any $x_u \in [\frac{c_u}{La_u k_u}, 1]$, we can always increase the objective value P_d by replacing x_u with 1. Consequently, the optimal false negative value x_u^* could only be $\frac{c_u}{La_u k_u}$ or 1. □

4.2 Reduced Single Level Problem

We propose a single level formulation called PEDS (Personalized thrEsholds in Defending Sequential spear phishing attacks) based on Theorem 2. The key idea of PEDS is reducing the continuous search space to some isolated points by restricting x_u to two values $x_u = \frac{c_u}{La_u k_u}$ and $x_u = 1$. In this way, not only the lower level problem is eliminated, but also the non-linearity of the upper level objective is overcome. We introduce a new variable ω_u so that $\theta(\mathbf{x}, \pi_{\mathbf{x}})$ can be rewritten as $\theta(\mathbf{x}, \pi_{\mathbf{x}}) = 1 - \prod_{u \in U} \omega_u$. Then, PEDS can be represented as follows.

$$\begin{aligned}
\max \quad & -(1 - \prod_{u \in U} \omega_u)L - \sum_{u \in U} \phi(x_u)h_u, \\
\text{s.t.} \quad & x_u = 1, \ \forall u \in \mathcal{U}, \\
& \omega_u = 1, \ \forall u \in \mathcal{U}, \\
& x_u = \frac{c_u}{La_u k_u} + (1 - \frac{c_u}{La_u k_u})\beta_u, \ \forall u \in U \setminus \mathcal{U}, \\
& \omega_u = 1 - k_u \beta_u, \ \forall u \in U \setminus \mathcal{U}, \\
& \beta_u \in \{0, 1\}, \ \forall u \in U \setminus \mathcal{U},
\end{aligned}$$

where $\mathcal{U} = \{u | \frac{c_u}{La_u k_u} \geq 1, u \in U\}$. For the users $u \in U \setminus \mathcal{U}$, $\beta_u = 0$ indicates that user u does not belong to the potential

target set $\Gamma(\pi_{\mathbf{x}})$, and therefore, $x_u^* = \frac{c_u}{La_uk_u}$. Similarly, $\beta_u = 1$ indicates that user u belongs to $\Gamma(\pi_{\mathbf{x}})$ and $x_u^* = 1$.

Although the objective of PEDS is non-linear, however, since the decision variables of PEDS are $\beta_u (u \in U \backslash \mathcal{U})$, which are all binary variables, we can always find the optimal solutions by, in the worst case, trying all the possible combinations.

5 Experimental Evaluation

To evaluate our approach, we did two sets of experiments. First, we solve PEDS using the Cplex CP Optimizer and show that it can be solved within 60 seconds even when the numer of users reaches 70. Second, we compare our personalized thresholds with the optimal uniform thresholds, as well as the personalized thresholds computed by [Laszka et al., 2015]. Experimental results show that our thresholds lead to significant higher defender utilities than the two baseline approaches.

In both experiments, we assume that an email classifier is given and the false positive function is characterized by $\phi(x_u) = \frac{0.0127}{x_u + 0.0125} - 0.0125$[6], as shown in Figure 2. We set the key information value $L = 200$. Other parameters are uniformly randomly generated. Specifically, the susceptibility measurements a_u are generated from $[0, 0.5]$; the probabilities that the users hold the credential k_u are generated from $[0, 0.2]$; the costs c_u are generated from $[0, 10]$; and the false positive losses h_u are generated from $[0, 100]$.

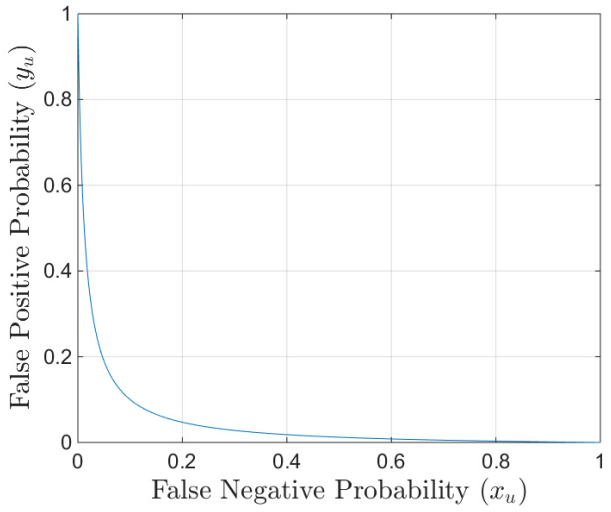

Figure 2: Functional Relationship Between False Positive Probability and False Negative Probability

[6]This function is an estimation from the existing work [Laszka et al., 2015], which trained a point-wise false positive function using real-world datasets.

5.1 Scalability

We use Cplex CP Optimizer to solve PEDS, which employs automatic search methods such as large neighborhood search, random restart and impact-based search. Figure 3 shows the runtime performance. For each experiment set, the runtime shown in the figure is the mean of the results of 50 trials. In Figure 3, the number of users refers to the number of users in $U \backslash \mathcal{U}$, which is equal to the number of decision variables. The results show that PEDS can be solved within 110 seconds even when the number of users reaches 80. In reality, spear phishers usually target a specific group of people or employees from a specific department/organization. For example, in January 2015, a former U.S. Nuclear Regulatory Commission (NRC) employee launched a spear phishing attack towards the Department of Energy of NRC, where about 80 employees were targeted [FBI, 2015]. We argue that the scalability of PEDS is good enough for most cases, though it needs further improvement.

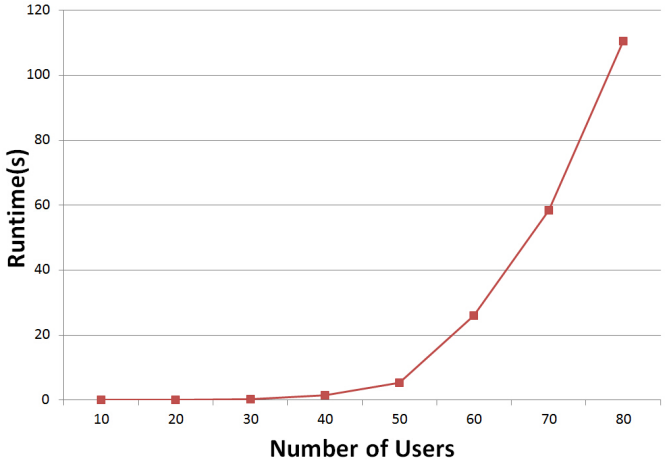

Figure 3: Runtime Performance

5.2 Solution Quality Comparisons

Baseline 1: The first baseline we consider is the optimal uniform threshold, where the defender chooses an optimal uniform threshold x^* for every user. Based on the results of Lemma 1 and Lemma 2, we formulate the problem of finding x^* as follows.

$$
\begin{aligned}
\max \quad & -(1 - \prod_{u \in U} \omega_u)L - \sum_{u \in U} \phi(x_u)h_u, \\
\text{s.t.} \quad & (\beta_u - 0.5)(x^* - \frac{c_u}{La_uk_u}) \geq 0, \ \forall u \in U, \\
& \omega_u = 1 - k_u\beta_u, \ \forall u \in U, \\
& \beta_u \in \{0, 1\}, \ \forall u \in U, \\
& x^* \in [0, 1],
\end{aligned}
$$

where $\beta_u = 0$ indicates that user u does not belong to the potential target set $\Gamma(\pi_{\mathbf{x}})$, and $\beta_u = 1$ indicates that user u

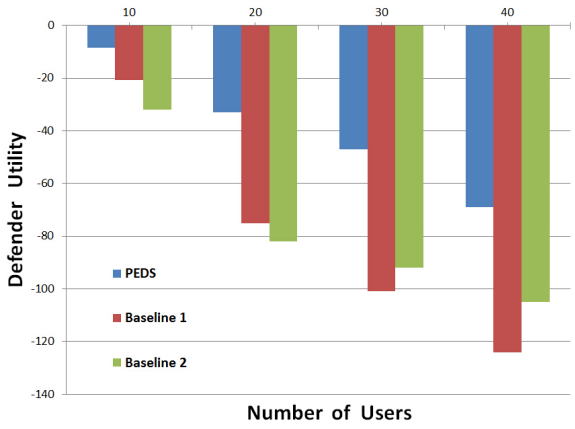

Figure 4: PEDS VS. Two Baselines

belongs to $\Gamma(\pi_{\mathbf{x}})$.

Baseline 2: The second baseline is the algorithm introduced by [Laszka et al., 2015]. In their work they assume that the defender sustains losses from non-targeted malicious emails (e.g., spam) besides spear phishing emails. We set the losses from non-targeted malicious emails to zero when executing their algorithm. Moreover, they assume that each user has an expected loss from false negatives if it is targeted by the attacker, which can be represented as $Lx_u a_u k_u$ using our notations.

The experimental results are shown in Figure 3. We conduct four groups of experiments with the number of users being 5, 10, 15 and 20. Since the parameters are randomly generated, the defender utilities shown in the figure are the means of results of 50 runs. From the results we can see that the thresholds computed by PEDS leads to significantly higher defender utilities than both baselines.

6 Conclusion

Building on previous research on classifying spear phishing emails, in this paper we investigate how to set appropriate thresholds to defend against spear phishers who launch sequential attacks. This paper makes four main contributions. (1) We propose a Stackelberg game model in which the attacker launches sequential attacks to get a credential. (2) We model the attacker's decision making as an MDP, which captures the strategic behavior of the attacker. (3) We provide analysis of the structure of the defender's bilevel problem that leads to a more efficient formulation called PEDS. In addition, PEDS overcomes the non-linearity of the objective function by reducing the search space for optimal solutions to only some isolated points. (4) We show that PEDS can scale up to at least 70 users and leads to significantly higher defender utility than two existing approaches.

Overall, our results can be used by governments, companies and other institutes who want to protect some secret credentials by mitigating spear phishing attacks. Our work also arouses some potential fields for future work. The limitations of this work include: (1) We assume that the game is fully observable, whereas in some situations there may be some uncertainty about the parameters and strategies. (2) We assume that a user's susceptibility remains constant during attacks, while in reality it may change over time.

References

[An et al., 2013] An, B., Brown, M., Vorobeychik, Y., and Tambe, M. (2013). Security games with surveillance cost and optimal timing of attack execution. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*, pages 223–230.

[An et al., 2011] An, B., Tambe, M., Ordonez, F., Shieh, E. A., and Kiekintveld, C. (2011). Refinement of strong stackelberg equilibria in security games. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*, pages 587–593.

[Bellman, 1961] Bellman, R. (1961). *Adaptive control processes: A guided tour*, volume 4. Princeton university press.

[Bergholz et al., 2010] Bergholz, A., De Beer, J., Glahn, S., Moens, M., Paaß, G., and Strobel, S. (2010). New filtering approaches for phishing email. *Journal of computer security*, 18(1):7–35.

[Choo, 2011] Choo, K. R. (2011). The cyber threat landscape: Challenges and future research directions. *Computers & Security*, 30(8):719–731.

[Deshmukh et al., 2014] Deshmukh, P., Shelar, M., and Kulkarni, N. (2014). Detecting of targeted malicious email. In *IEEE Global Conference on Wireless Computing and Networking (GCWCN'14)*, pages 199–202.

[Dewan et al., 2014] Dewan, P., Kashyap, A., and Kumaraguru, P. (2014). Analyzing social and stylometric features to identify spear phishing emails. In *APWG Symposium on Electronic Crime Research*, pages 1–13.

[FBI, 2015] FBI, W. F. O. (2015). Former US Nuclear Regulatory Commission employee charged with attempted spear-phishing cyber-attack on Department of Energy computers. http://www.fbi.gov/washingtondc/press-releases/2015/former-u.s.-nuclear-regulatory-commission-employee-charged-with-attempted-spear-phishing-cyber-attack-on-department-of-energy-computers/.

[Gan et al.,] Gan, J., An, B., and Vorobeychik, Y. Security games with protection externalities. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 914–920.

[Grassmann et al., 1985] Grassmann, W. K., Taksar, M. I., and Heyman, D. P. (1985). Regenerative analysis and steady state distributions for Markov chains. *Operations Research*, 33(5):1107–1116.

[Higbee et al., 2014] Higbee, A., Belani, R., and Greaux, S. (2014). Collaborative phishing attack detection. US Patent 8,719,940.

[Jagatic et al., 2007] Jagatic, T. N., Johnson, N. A., Jakobsson, M., and Menczer, F. (2007). Social phishing. *Communications of the ACM*, 50(10):94–100.

[Kelley, 2010] Kelley, P. G. (2010). Conducting usable privacy & security studies with amazons mechanical turk. In *Proceedings of the Sixth Symposium on Usable Privacy and Security (SOUPS'10)*.

[Korzhyk et al., 2011] Korzhyk, D., Yin, Z., Kiekintveld, C., Conitzer, V., and Tambe, M. (2011). Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, 41(2):297–327.

[Laszka et al., 2015] Laszka, A., Vorobeychik, Y., and Koutsoukos, X. (2015). Optimal personalized filtering against spear-phishing attacks. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 958–964.

[Schweitzer and Seidmann, 1985] Schweitzer, P. J. and Seidmann, A. (1985). Generalized polynomial approximations in Markovian decision processes. *Journal of mathematical analysis and applications*, 110(2):568–582.

[Sheng et al., 2010] Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L. F., and Downs, J. (2010). Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 373–382.

[Sheng et al., 2009] Sheng, S., Kumaraguru, P., Acquisti, A., Cranor, L., and Hong, J. (2009). Improving phishing countermeasures: An analysis of expert interviews. In *Proceedings of the 4th APWG eCrime Researchers Summit*, pages 1–15.

[Shieh et al., 2012] Shieh, E. A., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., and Meyer, G. (2012). Protect: An application of computational game theory for the security of the ports of the united states. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*, pages 2173–2179.

[Tambe et al., 2014] Tambe, M., Jiang, A. X., An, B., and Jain, M. (2014). Computational game theory for security: Progress and challenges. In *AAAI Spring Symposium on Applied Computational Game Theory*.

[TrendLabs, 2012] TrendLabs (2012). Spear-phishing email: Most favored APT attack bait. Technical report, Trend Micro.

[Varma, 2010] Varma, R. (2010). Combating Aurora. Technical report, McAfee Labs.

[Yin et al., 2014] Yin, Y., An, B., and Jain, M. (2014). Game-theoretic resource allocation for protecting large public events. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*, pages 826–834.

[Yin et al., 2015] Yin, Y., Xu, H., Gan, J., An, B., and Jiang, A. X. (2015). Computing optimal mixed strategies for security games with dynamic payoffs. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*.

[Zetter, 2011] Zetter, K. (2011). Researchers uncover RSA phishing attack, hiding in plain sight. http://www.wired.com/2011/08/how-rsa-got-hacked/.

Appendix

Proof of Lemma 1

Lemma 1. $u \in \Gamma(\pi_{\mathbf{x}})$ if and only if $x_u > \frac{c_u}{La_uk_u}$.

Proof. If direction: If $u \notin s_r$, there is a state $s_i \in \Delta(\pi_{\mathbf{x}})$ such that $\pi_{\mathbf{x}}(s_i) = u$. Therefore $u \in \Gamma(\pi_{\mathbf{x}})$. If $u \in s_r$, then

$$\begin{aligned} Q(s_r, a{=}u) &= (1 - a_u x_u)(V^*(s_r) - c_u) + a_u x_u k_u (L - c_u) \\ &\quad + a_u x_u (1 - k_u)(V^*(s_r^{-u}) - c_u) \\ &\geq (1 - a_u x_u)(Q(s_r, a{=}u) - c_u) \\ &\quad + a_u x_u k_u (L - c_u) + a_u x_u (1 - k_u)(0 - c_u). \end{aligned}$$

By adjusting the terms we have:

$$\begin{aligned} Q(s_r, a{=}u) &\geq -\frac{c_u}{a_u x_u} + Lk_u \\ &> 0 = Q(s_r, a{=}0). \end{aligned}$$

This means that at state s_r the attacker's optimal choice is $\pi_{\mathbf{x}}(s_r) = u$. Therefore $u \in \Gamma(\pi_{\mathbf{x}})$.

Only if direction: First, consider state s and s^{-u}. If we restrict the attacker's policy so that he never attacks u, we will have $V^*(s){=}V^*(s^{-u})$. If without the restriction, we have $V^*(s){\geq}V^*(s^{-u})$. In other words, adding a user to a state does not decrease its value. We prove that if $\pi_{\mathbf{x}}(s){=}u$, then $x_u > \frac{c_u}{La_uk_u}$. By definition we have:

$$\begin{aligned} V^*(s) &= (1 - a_u x_u)(V^*(s) - c_u) + a_u x_u k_u (L - c_u) \\ &\quad + a_u x_u (1 - k_u)(V^*(s^{-u}) - c_u) \end{aligned}$$

By adjusting the terms we have:

$$V^*(s) = -\frac{c_u}{a_u x_u} + Lk_u + (1 - k_u)V^*(s^{-u}).$$

Since $V^*(s){\geq}V^*(s^{-u})$, we have:

$$-\frac{c_u}{a_u x_u} + Lk_u \geq k_u V^*(s^{-u}) \geq 0$$

Note that if $-\frac{c_u}{a_u x_u}{+}Lk_u{=}0$, we have $V^*(s){=}V^*(s^{-u}){=}0$ and $s{=}\{u\}$. Due to the setting that the attacker always prefers stopping attack rather than launching another attack, we have $\pi_{\mathbf{x}}(s){=}0$, which contradicts the assumption that $\pi_{\mathbf{x}}(s){=}u$. Therefore, $-\frac{c_u}{a_u x_u}{+}Lk_u > 0$, equivalently, $x_u > \frac{c_u}{La_uk_u}$. □

Proof of Lemma 2

Lemma 2. $\theta(\mathbf{x}, \pi_{\mathbf{x}}) = 1 - \prod_{u \in \Gamma(\pi_{\mathbf{x}})}(1 - k_u)$.

Proof. Recall that the reachable states set is represented as $\Delta(\pi_{\mathbf{x}}) = \{s_0, s_1, ..., s_r\} \cup \{s^n, s^y\}$. We denote by $M_{\Delta(\pi_{\mathbf{x}})}$ the transition probability matrix, whose entry M_{ij} represents the probability that state s_i transitions to s_j under policy $\pi_{\mathbf{x}}$ (WLOG, we define s_{r+1}=s^n and s_{r+2}=s^y). There are two cases for s_r: (1) $\pi_{\mathbf{x}}(s_r) = u \in \mathcal{A}^{s_r}$ and (2) $\pi_{\mathbf{x}}(s_r) = 0$. If case (1), s_r could transition to itself, s^n or s^y. Thus $M_{\Delta(\pi_{\mathbf{x}})}$ has the form like (denote d_i=$a_{u^i}x_{u^i}$ and $k_i = k_{u^i}$):

$$\begin{bmatrix} 1-d_0 & d_0(1-k_0) & & & & d_0k_0 \\ & 1-d_1 & d_1(1-k_1) & & & d_1k_1 \\ & & \ddots & \ddots & & \vdots \\ & & & 1-d_r & d_r(1-k_r) & d_rk_r \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}$$

Precisely, $M_{\Delta(\pi_{\mathbf{x}})}$ can be represented as:

$$M_{\Delta(\pi_{\mathbf{x}})} = \begin{bmatrix} A & B \\ 0 & I_2 \end{bmatrix}$$

where A is r+1 dimensional square matrix, I_2 is 2 dimensional unit diagonal matrix and B is $(r+1) \times 2$ matrix. We introduce a $(r+1) \times 2$ matrix E:

$$E = FB, \text{ where } F = (I_{r+1} - A)^{-1}$$

Note that s^n and s^y are absorbing states. According to the properties of absorbing Markov chain, s_0 will eventually end in state s^n or s^y with probability E_{11} and E_{12} respectively, and E_{11}+E_{12}=1. Therefore, $\theta(\mathbf{x}, \pi_{\mathbf{x}})$=$E_{12}$. We can directly calculate E_{11} by matrix calculation:

$$\begin{aligned} E_{11} &= \sum_{i=1}^{r+1} F_{1i}B_{i1} \\ &= F_{1,r+1}B_{r+1,1} \\ &= \frac{\prod_{i=0}^{r-1}(1-k_{u^i})}{d_{u^r}} d_{u^r}(1-k_{u^r}) \\ &= \prod_{i=0}^{r}(1-k_{u^i}) \\ &= \prod_{u \in \Gamma(\pi_{\mathbf{x}})}(1-k_u) \end{aligned}$$

Then $E_{12} = 1 - E_{11} = 1 - \prod_{u \in \Gamma(\pi_{\mathbf{x}})}(1 - k_u)$. If case (2), s_r transitions to s^n with probability 1. Thus $M_{\Delta(\pi_{\mathbf{x}})}$ has the form like (denote d_i=$a_{u^i}x_{u^i}$ and $k_i = k_{u^i}$):

$$\begin{bmatrix} 1-d_0 & d_0(1-k_0) & & & & & d_0k_0 \\ & 1-d_1 & d_1(1-k_1) & & & & d_1k_1 \\ & & \ddots & \ddots & & & \vdots \\ & & & 1-d_{r-1} & d_{r-1}(1-k_{r-1}) & 0 & d_{r-1}k_{r-1} \\ & & & & & 1 & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix}$$

Similarly,

$$\begin{aligned} E_{11} &= \sum_{i=1}^{r+1} F_{1i}B_{i1} \\ &= F_{1,r+1} \\ &= \prod_{i=0}^{r-1}(1-k_{u^i}) \\ &= \prod_{u \in \Gamma(\pi_{\mathbf{x}})}(1-k_u) \end{aligned}$$

Then, we still have $E_{12} = 1 - E_{11} = 1 - \prod_{u \in \Gamma(\pi_{\mathbf{x}})}(1 - k_u)$. □