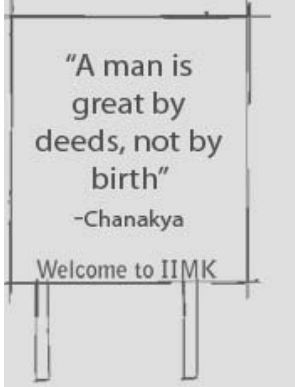

INDIAN INSTITUTE OF MANAGEMENT KOZHIKODE

Working Paper

IIMK/WPS/475/ECO/2021/12

September 2021

A Game-Theoretic Modeling of Deception-based Security System with Strategic Signaling

Anirban Ghatak[1]
Balamurali Rajendran[2]
Deepak Gujraniya[3]
Satnam Singh[4]

[1]Assistant Professor, Economics, Indian Institute of Management Kozhikode, IIMK Campus, Kunnamangalam, Kozhikode, Kerala 673 570, India; Email: aghatak@iimk.ac.in, Phone number: +91 495 2809655

[2]Acalvio Technologies Inc, Embassy Golf Links Business Park, Domlur, Bengaluru, Karnataka 560 071, India; Email: balamurali@acalvio.com

[3]Acalvio Technologies Inc, Embassy Golf Links Business Park, Domlur, Bengaluru, Karnataka 560 071, India; deepak@acalvio.com

[4]Acalvio Technologies Inc, Embassy Golf Links Business Park, Domlur, Bengaluru, Karnataka 560 071, India; satsingh@acalvio.com

A Game-Theoretic Modeling of Deception-based Security System with Strategic Signaling

Abstract. Deception technologies are gaining popularity in the domain of cyber-defense. This paper attempts to model deception as a strategic decision in a non-cooperative game setting. We have modeled the interaction between cyber security systems and the hacker as an attacker-defender game. A costless exponential learning scheme is introduced for the attacker wherein the game is played on an abstract network graph. The game is simulated on an active directory user network for privilege escalation attack scenario. Deceptions, in the form of fake users, are planted across the network. The strategy of the game lies in the placement of decoys at a different location in the network to obstruct the attackers desired path for achieving his objective. The results demonstrate that even the simplest deception-based security system significantly slows the attacker to achieve his objectives. Moreover, the results suggest that the network parameters and cost shading associated with nodes play an essential role in deciding the outcome.

Keywords: Cyber-Security; Game Theory; Deception; Simulation; Attacker-Defender Game

1 Introduction

Traditional cyber security defense relies on perimeter-based approaches (Zaliva, 2008). These approaches utilize anomaly detection systems to suface our dubious event by analyzing security data lakes. Data lakes are data stores where logs from different systems inside a security network are collected. Security data lakes are huge and munches millions of security events per second from various data sources. Any anomalous event is detected and shown to the security analyst to check the fidelity and authenticity of the alert. However, these systems are not robust due to the following reasons:

1. A large number of false positives (Axelsson, 2000)
2. Capturing, storing and indexing data lake is an expensive and a complex process

Moreover, a large number of false alerts take a toll on the security analyst, leading to scenarios where real alerts are missed. These systems follow a passive defense strategy wherein the objective is to *prevent an attack*. This seldom works as mean *time-to-compromise* a target system is less and has been steadily decreasing (Leversage and Byres, 2008). The traditional network perimeter - where many of these prevention technologies are typically deployed - has become porous and is routinely being breached. The proliferation of cloud computing, mobility and bring your own device (BYOD), and Internet-facing applications have rendered these perimeter defenses ineffective (inc, 2017).

Deception technology is fast emerging as an active form of cyber security defense (Mitnick and Simon, 2011; Almeshekah, 2015; Yuill et al., 2006) and is being used to mitigate above scenarios. Deception technologies focus on creating traps (deceptions/decoys) and lures that are deployed within the existing IT infrastructure. The deceptions used are not part of the regular operations but is only revealed during a cyber-attack. The attacker or the intruder expend time and effort to locate and access deception distributed across the enterprise network. They do so thinking

that deceptions are real but in reality are set up specifically to invite an attack. Any operation on deception is a positive affirmation of a compromise. In other words, in a deception- based solution, a highly positive anomaly announces itself, thus mitigating the false positive deluge (inc, 2017).

In this paper, we formulate a non-cooperative attacker-defender game to model the interaction between the attacker and defender using deception as a tool for active defense. The idea of modeling the interaction between hackers and security system as a game is not novel (Zhuang et al., 2010; Xu and Zhuang, 2016). However, defining the model of the game using deception within a graph framework has not been attempted before. In our framework, each atomic deception unit is considered as a node in a graph comprising of real service units. We refer to this graph as an abstract network graph (ANG). ANG is an isomorphic abstraction over the real network graph. Each atomic functional unit forms a part of ANG. Thus, a host machine comprising of individual functional units forms sub-graph in itself. For example, an enterprise host machine has a network card(NC) which is attached to the motherboard, controlled by the CPU. NC, motherboard, and CPU may be considered as nodes of an enterprise ANG. Any application or process running on this host machine will also be part of the ANG. A sample ANG representing different types of nodes are shown in Figure 1. We devise strategies for placing deceptions insider ANG to maximize the chances of defender winning. Different attack scenarios are modeled and simulated to enumerate different possibilities attacker might follow. The key idea is to deceive the attacker and mislead him, thereby exhausting his resources.

One of the resources sought after by attackers is the Active Directory (Chadwick, 2005; Metcalf, 2016). Active directory services control the access rights for a broad range of directory-based identity-related services. To make the modeling more realistic, we chose Active directory attacks for game simulation. The attacker tries to take control of the AD through different strategies. We focus our modeling on one form

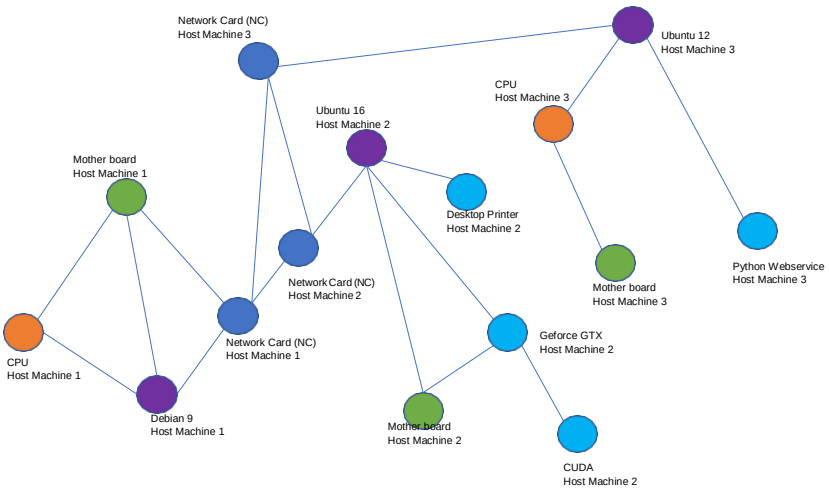

Fig. 1. A sample Abstract Network Graph (ANG) in an Enterprise

of privilege escalation using password reset methodology Metcalf (2016). This form of attack is commonly known as Reset-the-Password attack. The underlying idea is to exploit unauthorized access grant over users authentication. To mitigate such an exploitation, we place deceptions in the form of fake users with fake credentials to mislead the attacker. We present our results and analysis of the simulation of these attacks.

The key contributions and observations of this work are:

- A novel graph-based approach for the formulation of attacker-defender games using deception
- Empirically shows that deploying deception significantly increases the attacker's work to achieve his goal.
- Shows that increasing the number of nodes in graph i.e., number of users by adding more deceptions is beneficial.
- Identifies that Graph property plays an important role in the outcome of the duel between attacker and defender.

Even though we model the game for active directory attacks, our game model is scalable and robust to model any deception

based defense strategy. Rest of the paper is as follows: In Section 2, we describe the deception and concept behind ANG. We contextualize our work is section 3. The game formulation and encompassing model is explained in the Section 4. Section 5 explain our experiment setup. In the next section, results and discussion of the simulation are presented. Finally concluding the paper in Section 7 with some future pointers.

2 Deception for Active Defense

Deception-based threat detection provides an effective alternative to anomaly detection (inc, 2017). Any component in an enterprise network such as computer system, service, credential, a data item can be used for deception based detection. Deception seamlessly blends into the enterprise network and can be activated only when required. Deception in enterprise security can be categorized into:

- Decoy: A fabricated software or service for the attacker to target such as a PHP website.
- Breadcrumbs: Signals leading to the decoy. For instance, a shared folder with credentials leading to a decoy.
- Baits: Honeytokens in the counterfeit form data or fake credential to a service which may be valuable to the attacker. For example, fake credit card and personal information of fake users in the enterprise.
- Lures: It makes a decoy, breadcrumbs, baits more attractive to the attacker. For instance a service decoy with factory settings.

Each deception component has a certain form of a signal associated with it. For instance, if it is a web server decoy, then there is an acknowledgment signal for every request posted. Additionally, these components can point to another set of services. A web server will have a remote connection to database service, which in turn has got its signals. This is true for both decoys as well as real services or applications. In terms of the signals emitted and interlinked, components in an enterprise network can

be considered as a part of a large graph. This network is the abstract network graph (refer Figure 1). ANG forms the fabric over which deceptions can be deployed. Based on what needs to be seen, an underlying signal network over ANG could be deployed.

2.1 Active Directory - Privilege Escalation Attack

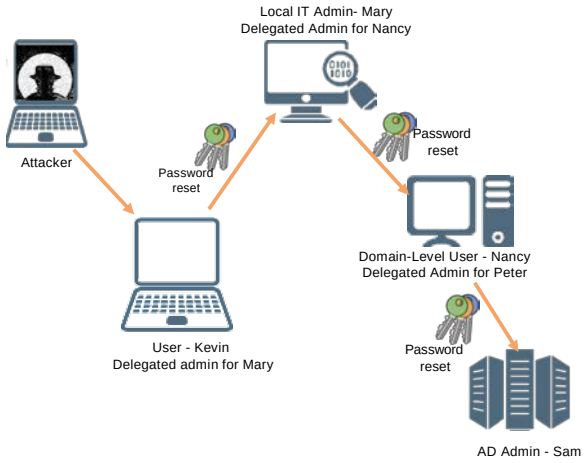

Fig. 2. Privilege escalation using a password reset attack

Active Directory (AD) is one of the key IT assets of any organization as it contains information of all the IT resources (hosts, servers, printers, etc.) and users. It makes user management easier by providing a single repository for all the users and IT resources. It provides this service using Kerberos Authentication and Single Sign-On (SSO). Kerberos provides a user with one set of credentials and grants them access across a range of resources and services with that same set of credentials. Attackers are primarily after privilege access to AD so that they can identify the identity and location of golden assets, for instance enterprise CEO/CFO accounts, customer databases, source code and build servers information, etc. They primarily use active directory privilege escalation based on the

identification and exploitation of unauthorized access grants (Defenses, 2016). Attackers use the Reset Password feature to reset the password of a targeted user. The reset password feature allows anyone who has this permission to instantly reset that account's password, and login as that account. This is a desired feature as the IT environment needs to be maintained and managed by multiple admins, and sometimes the delegated admin needs to reset the password of the admin account for user and resource management. This feature is exploited by the attacker for privilege escalation. Typically, an attacker gets a domain user-level access via a social engineering-based phishing attack. Then they identify all the users having higher-level access. Using the password reset analysis on the delegated admin they escalate the access to the next level. They keep on doing this analysis until they get the AD Admin credentials. We illustrate this attack in Figure 2. The attacker first compromises Kevin's account who is delegated admin for Mary. Mary is a local IT Admin. Using the password reset method, attacker resets Mary's password and compromises Mary's account. In the next step, the attacker compromises Peter's account using Mary's account. In the end, the attacker gets AD admin access using Peter's account, which is delegated admin for Sam. In the real world, it just takes less than 5 mins for the attacker to do this kind of privilege escalation if he knows the attack path, i.e. (Kevin $\longrightarrow$ Nancy $\longrightarrow$ Mary $\longrightarrow$ Peter $\longrightarrow$ Sam). There are several tools available online that can provide attack graphs for active directory(Robbins, 2016). Fig- ure 3 shows the attack scenario in the presence of deception. Deception could be of various types including decoy users showing fake access at var- ious levels e.g., local admin, domain-level, and admin-level. Deception is also provided using various types of breadcrumbs over the endpoints, and these breadcrumbs point to decoys. Lures could be provided to attract the attacker towards decoy users and make him believe that it is real. Now when the attacker tries to do privilege escalation using the password reset method, he may bump into one of the breadcrumbs, and that will increase the Blue teams probability of detecting the attacker presence.

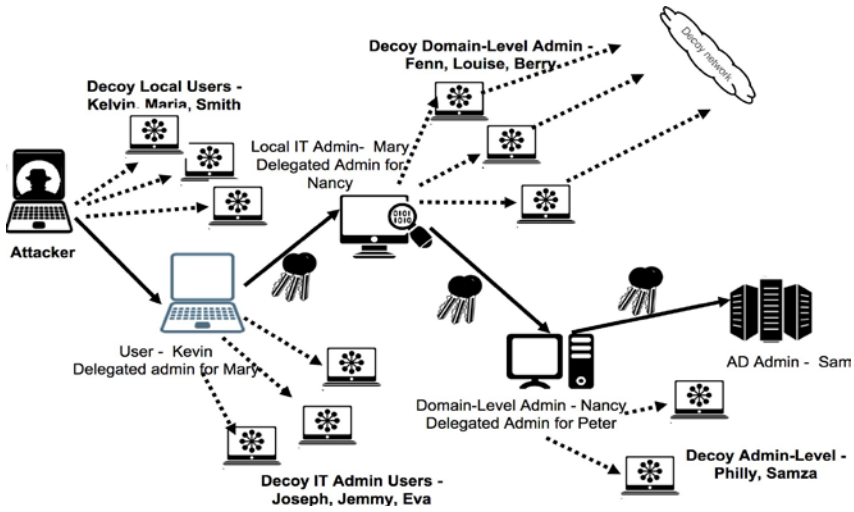

Fig. 3. Privilege escalation with deception deployed in between. The attacker is forced to explore a larger network in this case.

In this paper, we consider the scenario where deception in the form of fake users and fake credentials are setup in the AD user network. Using this form of deception we model an attacker-defender game in section 4.

3 Related Work

In this section, we contextualize our research with the existing works in the literature.

The use of deception is an age-old one, especially in the area of warfare (Dewar, 1989), counter-terrorism, homeland security (Zhuang and Bier, 2011) and so on. Though we can find many descriptive and analytical literature concerning deception in a physical and virtual warfare scenario, as far as our knowledge, there is no such analytical model that characterizes the efficiency of the method of deception in the context of cyber attack and defense. Use of deception for active security defense is gaining popularity (Lin et al., 2008; Zhang et al., 2003). In this work, we employ deception to model attacker-defender interaction. We model the use of deception in the cyber network as an attacker-defender game with

exponential learning for the attacker and with private information for the defender.

Concept of deception in the context of attacker-defender games have been attempted before (Xu and Zhuang, 2016; Zhuang et al., 2010). Zhuang et al. (2010) showed that in a multiple period game, deception could serve as a cost-effective defense strategy for the defender. Xu and Zhuang (2016) assumed that the attacker's learning is costly, and the cost of learning has a significant effect on the equilibrium payoff of both the attacker and defender. Both of these works are immensely important in the area of deception based defense. However, none of the literature that is currently available deal with the network structure of the security system. Moreover, there is no research available in the domain of cybersecurity, which deals with the efficiency of deploying deception based defense system from a game theoretic perspective and considering deception as a strategic decision. We approach the problem of determining the effect of deception based defense system from a game theoretic perspective. The model of the game is explained in detail in the next section.

4 Formulation

We model the event of cyber attack and defense as an attacker-defender game that is played on a network of users/processes. We assume that in the network of users/processes, there is a process that is of very high value, and the attacker is trying to reach that node in the network. Each node in the network is characterized by several parameters. Each node is initially characterized by a unique ID, that is a proxy of its location in the network. The 'visibility' of a node is defined as the binary variable which decides whether a node is visible to the attacker at a time t or not. The parameter 'operability' denotes whether a node that is visible to the attacker is operable by the attacker or not. Here, by operable, we denote the event of usability of the process in the node by the attacker. If the node is operable at time t, the attacker gets a positive payoff from the node. The operability of any node also decreases exponentially with time. Every

node houses a user or a process of the cyber network. Hence, every node has a standalone value attached to it. This value is assumed to be the value that the attacker of the network will gain, without the knowledge of the position of the node in the network, if the node is compromised/hacked by the attacker. The position of a node in the network is also another important parameter to understand the importance of the node. If a node is compromised, its position or connectedness defines how vulnerable the node is for the network. Thus, the degree centrality of the node is also an important parameter that we will consider in this model. The properties of a node are mathematically formulated in Section 4.1.

In the context of active directory, nodes represent different users with different credentials. Each user is unique, visible only to a unique set of users. The operability is in the formability to reset the password.

The game is being played between two players: an attacker and a defender. The attacker enters the network through one node and at each time point jumps to another node based on the perceived payoff of the destination node. It continues moving through the network till it can find the node with the highest standalone value, that we call as treasure node in this paper.

We assume that the attacker has a limited budget, and it stops playing the game when the budget gets exhausted. The defender tries to stop the attacker from reaching the treasure node through mainly two actions. Firstly, it deploys deception at some nodes in the network. We explain the model of deception in Section 2. Secondly, the defender emits a signal about the importance of a node, which may or may not be the real importance of the node. This part of the model is explained in Section 4.3.

If the attacker reaches a deception node and does not recognize the deception, it incurs a negative payoff. Also, with every incident of deception, the attacker's ability to recognize a deception increases exponentially. We also include another parameter called risk for the attacker in the game. The risk of the attacker of being detected increases exponentially with

the time spent in the network. We assume that if the attacker does not move to another node at some time point, the defender wins the game. The expected payoff at any time t in the game is formulated in Section 4.5.

4.1 Characterization of a node in ANG

Consider a node i in the network. The key identifiers for node i are presented below:

1. Location: (x^i, y^i)
2. Visibility at time t: $v^i(t) \in \{0, 1\}$
3. Operability at time t: $r^i_t \in \{0, 1\}$, $P(r^i_t = 0) = 1 - e^{-t}$
4. Standalone value: c^i
5. Degree Centrality: C^i

At any time t, if there is a direct edge between node i and node j, then

$$v^i(t) = 1 \Rightarrow v^j(t + 1) = 1$$

4.2 Modeling Deception

Deception in this game is treated as a simple action that will (by any means) stop the attacker from reaping the benefits of an attacked node. Deception can be of many types in a real network. For simplicity, we have considered it to be a binary variable $D \in \{0, 1\}$. In the context of active directory attack, deception exists in the form of fake users with fake credentials.

The key properties of deception in this game are:

1. Probability that an attacker will recognize the deception is $P(D)$. $P(D)$ is independent of the node where the attacker is in.
2. We follow the standard exponential learning equation to model the increase in recognition probability as the attacker faces more deception. We also assume that the attacker learns from both success and failure

to recognize earlier deceptions. Probability that the attacker will recognize a deception after facing n earlier deceptions is $P(D_n) = 1 - e^{-\frac{n}{2}}$ (Leibowitz et al., 2010).

3. Deploying deception also incurs a cost c_D for the defender.

4.3 Cost Signal

Cost Shading The defender will also send a signal about the value or importance of each node that is visible to the attacker. The cost signal can be the true importance of the node or can be a shaded cost. To define the cost signal, we first define a metric that quantifies the 'importance' of a node in a network.

If a node i has a value c^i and a degree centrality C^i, then both the connectedness and the value contribute in determining the importance of the node. The value becomes essential in a standalone manner, which signifies the value of the information lost in the process of losing the control of the node, while the connectedness means the vulnerability in terms of visibility of the rest of the network, once the specific node is compromised. We define the importance of a node by formulating it as a Cobb-Douglas utility/production function

$$I^i = C^i c^{i\theta}$$

where θ signifies the relative importance (more commonly referred to as elasticity in Cobb-Douglas production functions) of the value of the node in relation to the location of the node in contributing to the overall importance of the node. θ is a property of the network and remains constant in a network.

We define the cost signal c_s^i of node i as

$$c_s^i = f(I^i)$$

Where the function f assures the following:

1. The most connected nodes look less attractive to the attacker

2. The most precious nodes look less attractive to the attacker

The exact nature of the function f will be derived after defining the payoff structure in Section 4.6.

Cost of cost shading: It is only logical to assume that the more important the node is, the more effort is needed to hide it from the attacker. Hence, the cost of cost shading can be seen as a function of the difference between the shaded cost and the actual importance. We formulate the cost of cost shading as

$$c_c^i = g(\varsigma^i - I^i)$$

For a basic simple model; we assume g to be a constant multiplier of 0.05 in the simulation. Thus,

$$c_c^i = 0.05(\varsigma^i - I^i)$$

4.4 Strategies

Strategy of the Defender is to choose a deception level ($D = 0$, or $D = 1$), and a cost signal of node i, c_s^i Thus, the action space of the Defender is $A_D = \{D = 0, D = 1\} \times \{f\}$

The strategy of the Attacker is to choose the next node to attack, once he is on node i. Symbolically, $A_{i_t \to j_{t+1}}$ denotes the action of the attacker of attacking node j from node i at time t. Another available action for the attacker would be not to attack any more nodes. We denote that action by $A_{i_t \to i_{t+1}}$.

The important properties of movement of an attacker are listed as below:

1. Every movement of the attacker between i and j at time t are also associated with a time-dependent risk R_t for the attacker. Let us formulate $R_t = \alpha + e^{\beta t}$ where α is the initial risk and β is the amplification/dampening coefficient.

2. $A_{i_t \to i_{t+1}}$ can happen when the attacker has no gain in moving forward. This denotes two possible situations in this game. Firstly, the attacker can be stalled in the middle of a network, which would essentially mean that the attack attempt has failed. Second, the attacker may choose $A_{i_t \to i_{t+1}}$ when he got the most important node in the network, i.e., when the attack is successful.
3. Let us denote the set of all nodes in the network as N. So, when an equilibrium of this game consists of $A_{i_t \to i_{t+1}}$, if $c^i = \max(c^j) \; \forall j \in N$, the attacker wins, else, the defender wins.

4.5 Expected Payoff Structure

1. **Case 1: D = 0**
   - No deception. Attacker gets the value of the node minus the risk, if the node is operable. Else, only loses the risk amount.

$$u_A(A_{i_t \to j_{t+1}}, D_{D_j=0, c_s^j}) = P(r_{t+1}^j = 1)(c_s^j + (I^i - c_s^i) - R_{t+1}) + P(r_{t+1}^j = 0)(-R_{t+1})$$

   - No deception. Defender loses the node value if the node is operable, and also loses the cost of cost shading.

$$u_D(A_{i_t \to j_{t+1}}, D_{D_j=0, c_s^j}) = P(r_{t+1}^j = 1)[-c^j] - c^j{}_c$$

2. **Case 2: D = 1**
   - Deception deployed. If deception is recognized by the attacker, he gets the same payoff as no deception. Else, loses the risk amount.

$$u_A(A_{i_t \to j_{t+1}}, D_{D_j=1, c_s^j}) = P(D)[u_A(A_{i_t \to j_{t+1}}, D_{D_j=0, c_s^j})] + (1 - P(D))[-R_{t+1}]$$

   - Deception deployed. If deception is recognized and the node is operable, she loses the value of the node. Also she loses the deception cost and the cost of cost shading.

$$u_D(A_{i_t \to j_{t+1}}, D_{D_j=1, c_s^j}) = P(D)[P(r_{t+1}^j = 1)[-c^j]] - c_D - c^j{}_c$$

4.6 Expected Payoff for the Attacker & the structure of *f*

When an attacker chooses its next node to attack, it does so by evaluating the expected payoff for each of the available nodes, and choosing the best one. Hence, the expected payoff for the attacker by choosing any node *j* that is connected to node *i* will be $u_A(i \longrightarrow j) = \frac{1}{2}[u_A(A_{i_t \to j_{t+1}}, D_{D_j=0, c^j_s}) + u_A(A_{i_t \to j_{t+1}}, D_{D_j=1, c^j_s})]$, as described in section 4.5.

Upon simplifying, we can rewrite

$$u_A(i \longrightarrow j) = A.(f(I^j) + (I^i - c^i_s)) - 2R_{t+1},\ A = \{(P(r^j_{t+1} = 1) + P(r^j_{t+1} = 1).P(D))\}$$

As we want $u_A(i \longrightarrow j)$ to be decreasing in I^j in order to make the crucial nodes look unattractive to the attacker, we define

$$f(I^j) = I_{max} - I^j$$

Where I_{max} is the importance value of the maximum important node in the network.

4.7 Equilibrium Conditions

Looking at it as a 2 × 2 game : In order to develop the condition of a Nash Equilibrium in this game, we can assume that the defender has two pure strategies to pursue, *i.e.* D=0 and D=1, and the attacker also has two pure strategies to pursue which are 'select node *j*' and 'do not select node *j*'. If we can find the condition for a Nash Equilibrium consisting 'do not select node *j*' as the attackers strategy, for all nodes directly connected to node *i*, *i.e.* the originating node, then we can say that such a condition will be sufficient for the attacker to stop movement at node *i*, which will, in turn, denote the win of the defender in this game. Attacker's strategy of not selecting node *j* will essentially mean that another node *k* will be selected. So, from the expected payoff structure at Section 4.5, in order to have a Nash Equilibrium consisting of 'do not select node *j*', the sufficient conditions are:

$$u_A(A_{i_t \to j_{t+1}}, D_{D_j=0, c^j_s}) \leq u_A(A_{i_t \to k_{t+1}}, D_{D_j=0, c^k_s})$$

and

$$u_A(A_{i_t \to j_{t+1}}, D_{D_j=1, c^j_s}) \leq u_A(A_{i_t \to k_{t+1}}, D_{D_j=1, c^k_s})$$

Both of these conditions will be satisfied if and only if

$$c^j_s \leq c^k_s \quad \forall k \leftrightarrow i \ \& \ k \neq j \tag{1}$$

In order to satisfy the condition 4.7, it is imperative that

$$c^j_s = c^k_s, \quad \forall k \leftrightarrow i \ \& \ k \neq j \tag{2}$$

Also, as we want the attacker to stay in node i, the further condition will be

$$c^j_s \leq I^i, \quad \forall j \leftrightarrow i \tag{3}$$

As $c^j_s = I_{max} - I^j$, the combined condition for having a Nash Equilibrium consisting of the attacker's strategy that forces the attacker to stay at a node i, rather than moving to any node j will be

$$c^j_s = c^k_s, \quad \forall k, j \leftrightarrow i \ \& \ k \neq j \tag{4}$$

$$I_{max} - I^j \leq I^i, \quad \forall j \leftrightarrow i \tag{5}$$

As it is evident from Equations 4 and 5, the conditions only depend on the 'arrangement' of the nodes according to their importance, in a specific order. Hence, it is the property of the network that can force this equilibrium without any other condition on any other parameter of the game.

Looking at it as a $2 \times n$ game Instead of looking at the attacker's choice as a binary one, we can also look at it as a bouquet of nodes to choose one from, as its next step in the network. Let us assume there are m nodes, $j_1, j_2, \ldots, j_m$ connected to node i at time t. Without loss of generality, we can always assume an order of their importance value as

$$I^{j_1} \geq I^{j_2} \geq \ldots \geq I^{j_m}$$

Hence, as

$$c_s^{j_p} = I_{max} - I^{j_p} \quad \forall p \in 1, 2, ..., m$$

we can see that

$$c_s^{j_m} \geq c_s^{j_{m-1}} \geq \ldots \geq c^{j_1}{}_s$$

It follows that

$$u_A(A_{i_t \to j_m t+1}, D_{D_{j_m}=0, c_s^{j_m}}) \geq u_A(A_{i_t \to j_p t+1}, D_{D_{i_p}=0, c_s^{j_p}}) \quad \forall j_p \leftrightarrow i \quad (6)$$

$$u_A(A_{i_t \to j_m t+1}, D_{D_{j_m}=1, c_s^{j_m}}) \geq u_A(A_{i_t \to j_p t+1}, D_{D_{i_p}=1, c_s^{j_p}}) \quad \forall j_p \leftrightarrow i \quad (7)$$

Thus, we prove from Equations 6 & 7 that with the choice of f in Section 4.6, the attacker's strategy of choosing the node with the lowest importance forms the Nash Equilibrium of this game. This equilibrium does not stop the attacker from moving forward in the network but definitely prevents it from accessing the most critical node in the immediate neighborhood.

It can be said that, theoretically, careful construction of the network can force the attacker to stop at a node pre-decided by the defender. Nonetheless, for any network, the choice of f is critical, and the choice of f used in this game assures that the attacker will choose the least vital node to attack at every step in its movement, regardless of the network structure.

5 Simulation

We set up a simulation framework to demonstrate our formulation for Attacker-Defender game. The framework generates a random graph with n number of nodes and an edge probability ξ. This graph represents the abstract network graph (ANG), where each node in the graph represents a host. Some of these hosts can be deception with fake users and a suite of fake services.

5.1 Deceptions deployment strategies

Whereas, in the real world, deceptions are deployed on a network based on some strategies suitable to it; in this simulation, we deploy deception after the generation of a network graph. The framework depicts the real world scenario as network graphs are generated randomly, and the selection of nodes for deception is dependent on the strategy and budget of the defender. Here, we experimented with the following two strategies:

1. Deploy deception on a fixed number of nodes, randomly. Here, we kept it as a percentage of total nodes in the ANG. In real life, it can depend on the budget and the cost of deception deployment.
2. Deploy deception strategically based on the node's connectivity in the network. It is more sensible to deploy deception on nodes with higher connectivity in the network as it increases the attacker's chance to stumble upon these nodes. In this simulation; for n nodes, we choose deception nodes based on following criteria:

$$D_i = \begin{cases} 0, \text{ if } I^i \leq \frac{1}{n} \sum_i I^i \\ 1, \text{ otherwise} \end{cases}$$

There can be many such strategies based on the network properties and nodes individual properties. We will discuss these strategies in our future work.

5.2 Choice of next step by the Attacker

The attacker chooses the next node for attack based on their perception of payoff. So, at every node n, at time t, the attacker evaluates the payoff for all j directly connected to i as $u_A(i \longrightarrow j) = \frac{1}{2}[u_A(A_{i_t \to j_{t+1}}, D_{D_j=0,c^j_s}) + u_A(A_{i_t \to j_{t+1}}, D_{D_j=1,c^j_s})]$, as described in section 4.6. Node j with maximum $u_A(i \longrightarrow j)$ is then chosen by the attacker.

In the real world, the attacker would not revisit a node, unless stalled on some node. The simulation captures this scenario by keeping track of visited nodes. In this case, if the best available node has already been

visited, it would select the second best available unvisited node in the network.

5.3 Computing exact payoff

Payoff at each node at time t should be calculated as follows:

1. When $D_i = 0$, and $r_t^i = 0$

$$u_A = -R_t, \quad u_D = -c_c^i$$

2. When $D_i = 0$, and $r_t^i = 1$

$$u_A = I^i - R_t, \quad u_D = -I^i - c_c^i$$

3. When $D_i = 1$, Deception is recognized, and $r_t^i = 0$

$$u_A = -R_t, \quad u_D = -c_D - c^i{}_c$$

4. When $D_i = 1$, Deception is recognized, and $r_t^i = 1$

$$u_A = I^i - R_t, \; u_D = -I^i - c_D - c^i{}_c$$

5. When $D_i = 1$, Deception is not recognized, and $r_t^i = 0$

$$u_A = -R_t, \quad u_D = -c_D - c^i{}_c$$

6. When $D_i = 1$, Deception is not recognized, and $r_t^i = 1$

$$u_A = -R_t, \quad u_D = -c_D - c^i{}_c$$

Algorithm 1 Simulation steps (n, ξ, θ, S, T)

1: Generate the network for given n, ξ and θ; assign i for each node
2: Assign standalone value c^i for each node
3: Calculate degree centrality C^i for each node
4: Assign importance value I^i for each node
5: Calculate cost of cost shading c_c^i for each node
6: First node = node with the lowest importance value
7: Treasure node = node with the highest standalone value
8: Deploy deceptions based on the strategy S
9: $t = 1$
10: **while** $t \leq T$ **do**
11: Calculate the payoff for both the players according to section 5.3
12: **if** the cumulative negative payoff received by the attacker exceeds the attacker's budget **then**
13: Stop and declare **defender(D)** as winner.
14: **else**
15: Choose the next node to attack by following section 5.2
16: **end if**
17: **if** the cumulative payoff of Defender exceeds the available budget or the selected node is the treasure node **then**
18: Stop, declare **attacker(A)** as winner.
19: **end if**
20: **end while**
21: Attacker could not find the treasure in T moves, declare **defender** as winner.

5.4 Experiments

In the game setup, the attacker is declared a winner if either of the following conditions is met:

1. Attacker has found the treasure node i.e., has gained access over admin of active directory server
2. Defender has run out of the budget to deploy further deceptions, which means the attacker can explore the network as much time as it requires to find the treasure node

While defender wins if either the attacker runs out of the budget or has no gain in exploring the network further.

The attacker has two constraints to adhere:

1. He/she cannot stay indefinitely: They have only limited time inside the enterprise network to avoid detection. Hence, several moves are limited, assuming each move takes some time. In this simulation, this time is consistent across all the steps.
2. As the stay continues, the risk of detection increases: Each time the attacker stumbles upon a deception node his chances for detection increases.

To observe the effect of network properties on the game, we simulated the game on various networks with a different number of nodes, edge probability ξ and θ. We generated a network for a given number of nodes and a given ξ. The game simulation was done on different values of θ and with different strategies of deception deployment. Results of the simulation are presented in section 6.

For the simulations, the graph is generated, and the deceptions are fixed based on the property of the graph. If there are no deceptions, then the attacker will get a higher cumulative payoff at the end of the game. We experiment with and without scenario to verify this hypothesis.

For the next set of experiments, we fix network properties ξ and θ, except for the size of the network. We vary the limit on several moves an attacker make to find the treasure node. Budgets of attacker and defender were also kept fixed. We, then, experimented with different budgets for attacker and defender on networks of various sizes. Deception deployment is kept constant for each network.

6 Results and Discussion

Table 1 shows results of the simulation with varying degree of network size, varying probability of an edge between two nodes, and varying θ. If the network of users is not large enough, then the inclusion of deception does not increase defenders winning chance. If the users are not that connected, then the attacker wins almost every time. From the results, it

is evident that θ is a vital parameter to predict the winner of the game. θ represents the relational importance of the value of the node against the significance of the location of the node. Observing the results across different number nodes, one can see network size and property has as an effect on deciding the winner. However, if θ is greater than 1, the attacker wins in all the simulation runs.

Table 1. Simulation runs for varying number of users, strategies, θ, probability of an edge between two users

		Network Size (Total Number of users)											
		10			20			30			40		
	Probability of an edge between any two users (ξ)	0.4	0.6	0.8	0.4	0.6	0.8	0.4	0.6	0.8	0.4	0.6	0.8
θ <1	D = strategy	A	A	D	D	A	A	D	A	D	D	D	A
	D = random	A	A	D	D	A	A	D	A	D	D	D	A
θ = 1	D = strategy	A	A	D	D	D	D	D	D	D	D	D	D
	D = random	A	A	D	D	D	D	D	D	D	D	D	D
θ > 1	D = strategy	A	A	A	A	A	A	A	A	A	A	A	A
	D = random	A	A	A	A	A	A	A	A	A	A	A	A

Table 2 presents the most likely winner of the game when the available moves of the attacker are constrained. This experiment was done to categorize Advance Persistent Threat (APT) kind of attack versus Malware attacks. Mostly in the APT scenario, the number of lateral movements attacker makes is minimalistic to decrease the risk of detection. Again, the critical observation is if the network is not large enough, the even with a small number of moves, the attacker can win. Even though not included in the result tables, we saw that if the attacker has unbounded steps and the network is extensive, he/she fails due to budget exhaustion.

We saw that network parameters have a significant effect on deciding the winner. Table 3 shows how the possibility of winning changes with the change of the available budget of the attacker. In certain scenarios, attackers have an infinite budget in terms of the tools and resources available. In such scenarios, i.e., when the attacker is significantly more 'wealthy' than the defender, the available budget can overshadow the effect of the network size. Also, when the number of moves available to the attacker (before being detected) is more, the probability of the attacker winning the game increases.

Table 2. simulation runs for a varying number of moves permitted for the attacker to go unnoticed for a fixed centrality measure. Here the attacker's and defender's budget is fixed.

	Number of users			
Number of Moves by Attacker	10	20	30	40
5	D	D	D	D
8	A	D	D	D
10	A	D	D	D
As many users	A	A	D	D

Table 3. Simulation runs for varying budget scenarios. The damage attacker can inflict depends on his/her budget.

	Number of user			
Scenario	10	20	30	40
Attacker Budget >>> Defenders budget	A	A	A	A
Attacker Budget <<< Defenders budget	D	D	D	D
Attacker Budget = Defenders budget	A	A	D	D
Attacker Budget ◻ Defenders budget	A	A	D	D

Figure 4 and Table 4 shows the (cumulative) attacker payoff at the end of the game. The results suggest, perhaps for the first time, that having

deception makes the attacker work more to achieve his/her target. This is an important finding.

Table 4. Simulation run to track the cumulative attacker payoff in scenarios where there is no deception and when there is deception

	Number of users			
Deception scenario	10	20	30	40
No Deception	-1016.6	-12832371.7	-65789324.7	-69988662.1
With Deception	-1247.1	-13921355.4	-67189128.2	-69988671.9

Generally, when the graph is static, and deception does not provide a different ‘fake worldview’ to the attacker but only prevents the attacker from gaining access to the node where deception is deployed, even with exponential learning capability of the attacker, one can successfully prevent an attack by creating a large network.

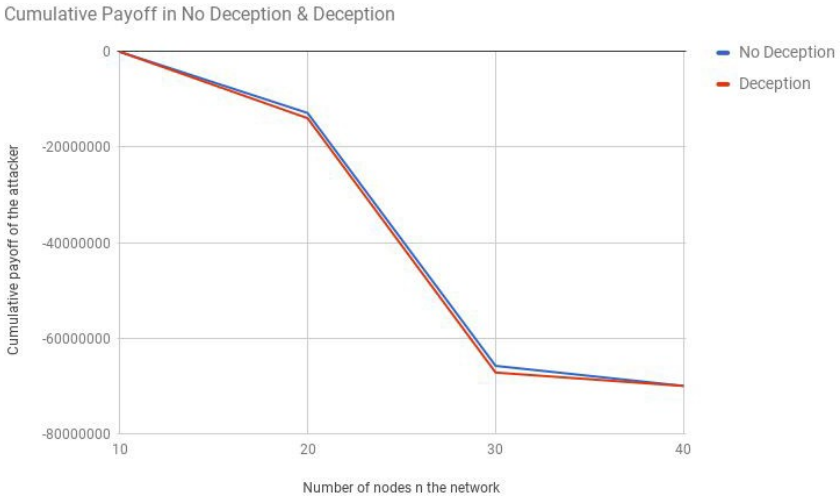

Fig. 4. Simulation run to track the cumulative attacker payoff in scenarios where there is no deception and when there is deception

7 Conclusions & Future Works

In this paper, we presented a novel model for attacker-defender games using deception-based security. To the best of our knowledge, our work is the first attempt to construct a game theoretic model of deception-based security. Novelty consisted of formulating the game in the form of an abstract network graph by strategically deploying the deceptions to exhaust the attacker's resources. Deception in the form of fakes users, and fake credentials were planted to defend an active directory privilege attack, and game simulations were performed. Our results suggest that the attackers can be slowed down in achieving his objectives using deceptions.

Moreover, results suggest that in achieving his/her target, the attacker has to expend more energy.

We analyzed the simulations over a static network. In reality, this network is dynamic. This needs to be further investigated along with network characteristics that affect the outcome of the game. Also, a majority of the networks are star networks, whereas we have simulated our game in a random network setting. This needs to be extended for various other network architecture.

Bibliography

Almeshekah, M. H. (2015). *Using deception to enhance security: A Taxonomy, Model, and Novel Uses*. PhD thesis, Purdue University.

Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)*, 3(3):186–205.

Chadwick, D. (2005). Threat modelling for active directory. In *Communications and Multimedia Security*, pages 173–182. Springer.

Defenses, P. (2016). Privileged access is the new holy-grail for malicious perpetrators. Online; accessed 15-June-2017.

Dewar, M. (1989). *The art of deception in warfare*. Sterling.

inc, A. T. (2017). *Deception 2.0 for Dummies*. John Wiley & Sons.

Leibowitz, N., Baum, B., Enden, G., and Karniel, A. (2010). The exponential learning equation as a function of successful trials results in sigmoid performance. *Journal of Mathematical Psychology*, 54(3):338–340.

Leversage, D. J. and Byres, E. J. (2008). Estimating a system's mean time-to-compromise. *IEEE Security & Privacy*, 6(1).

Lin, K., Kyaw, L., et al. (2008). Hybrid honeypot system for network security.

Metcalf, S. (2016). Attack methods for gaining domain admin rights in active directory. Online; accessed 15-June-2017.

Mitnick, K. and Simon, W. L. (2011). *The art of deception: Controlling the human element of security*. John Wiley & Sons.

Robbins, A. (2016). Bloodhound. Online; accessed 15-June-2017.

Xu, J. and Zhuang, J. (2016). Modeling costly learning and counterlearning in a defender-attacker game with private defender information. *Annals of Operations Research*, 236(1):271–289.

Yuill, J., Denning, D. E., and Feer, F. (2006). Using deception to hide things from hackers: Processes, principles, and techniques. Technical report, DTIC Document.

Zaliva, V. (2008). Firewall policy modeling, analysis and simulation: a survey. *Source-Forge, Tech. Rep*.

Zhang, F., Zhou, S., Qin, Z., and Liu, J. (2003). Honeypot: a supplemented active defense system for network security. In *Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on*, pages 231–235. IEEE.

Zhuang, J. and Bier, V. M. (2011). Secrecy and deception at equilibrium, with applications to anti-terrorism resource allocation. *Defence and Peace Economics*, 22(1):43–61.

Zhuang, J., Bier, V. M., and Alagoz, O. (2010). Modeling secrecy and deception in a multiple-period attacker–defender signaling game. *European Journal of Operational Research*, 203(2):409–418.

Research Office
Indian Institute of Management Kozhikode
IIMK Campus P. O.,
Kozhikode, Kerala, India,
PIN - 673 570
Phone: +91-495-2809237/ 238
Email: research@iimk.ac.in
Web: https://iimk.ac.in/faculty/publicationmenu.php