Preventing Collusion Between SDN Defenders and Attackers Using a Game Theoretical Approach

Mehrdad Nojoumian and Arash Golchubian
Electrical & Computer Engineering and Computer Science
Florida Atlantic University, Boca Raton, Florida, 33341 USA
{mnojoumian,agolchub}@fau.edu

Nico Saputro and Kemal Akkaya
Department of Electrical & Computer Engineering
Florida International University, Miami, FL, 33174 USA
{nsapu002, kakkaya}@fiu.edu

***Abstract*—In this paper, a game-theoretical solution concept is utilized to tackle the collusion attack in a SDN-based framework. In our proposed setting, the defenders (i.e., switches) are incentivized not to collude with the attackers in a repeated-game setting that utilizes a reputation system. We first illustrate our model and its components. We then use a socio-rational approach to provide a new anti-collusion solution that shows cooperation with the SDN controller is always Nash Equilibrium due to the existence of a long-term utility function in our model.**

I. Introduction

Software-Defined Networking (SDN) has enjoyed tremendous popularity growth in the last few years. Everyday, more networks are migrated to use the SDN instead of traditional networking technologies because of its numerous advantages such as manageability, flexibility in network design and scalability [1]. This is partially driven by the demand for better, faster and more complex content delivery for today's advanced web applications. More importantly, the SDN has started to be used for securing the networks by integrating ideas from Moving Target Defense (MTD) and Network Function Virtualization (NFV) paradigms [2]. The MTD provides a different perspective to secure the networks, which is one of the major issues in today's highly connected cyberspace. For this reason, government organizations such as the United States Computer Emergency Response Team (CERT) as well as private security companies, such as Symantec, McAfee and Kaspersky, spend hundreds of thousands of man-hours researching and mitigating security vulnerabilities in Internet-connected devices. A recent projection by Juniper Research estimates that the annual cost of data breaches will reach to two trillion dollars per year by 2019 [3].

The MTD paradigm simply applies continuous changes to the underlying network infrastructure in order to make it harder for the attackers to launch attacks. It heavily relies on the capabilities of the SDN. Furthermore, it utilizes the SDN controller and switches to apply certain changes such as route mutation, port and address mutation, service relocation, and configuration updates [4], [5]. Using such mechanisms, the MTD can be an effective means to thwart the Distributed Denial of Service (DDoS) attack, which is very difficult to be handled by traditional techniques. In all these approaches, the SDN controller and switches are assumed to be trusted and not to be compromised by the attackers.

However, this assumption may not be true in many cases. One threat vector of importance is the exploitation of weaknesses in the security of switches [6]. If compromised, a single switch can help the attacker to control the flow of traffic on the entire network by modifying the behavior of the switch. As a result, the switch will not respond (in an expected manner) to the instructions that are sent by the SDN controller. This can be exploited by the attacker to bring the network to a crawling halt. In particular, these types of vulnerabilities can be used to disrupt or inhibit the defense mechanisms against DDoS attacks such as Crossfire attacks that target a network area for taking down the network links [7], [8].

Moreover, the attacker can use vulnerabilities within a switch to change the control flow procedures. The affected switch would be reprogrammed by the attacker to ignore route mutation orders issued by the controller. As a result, the attacker can find permanent links. This can be further exploited to redirect the switch traffic to particular destinations with the aim of aiding the attacker in pursuing the DDoS attack. In our setting, these actions are referred to as collusion of the SDN elements with the attackers.

With the aim of mitigating the effects of such attacks and fully enjoying the benefits of SDN-based MTD approaches, we propose a game-theoretical solution concept in which the defenders (switches) are incentivized not to collude with the attackers. We first illustrate our model and its components. Subsequently, we utilize a socio-rational approach [9], [10] to provide a new anti-collusion solution that shows cooperation with the SDN controller is always Nash Equilibrium.

The rest of this paper is organized as follows. Section II briefly reviews the literature of game-theoretical approaches to SDN security. Section III illustrates our game-theoretical solution with its assumptions and analysis. Finally, Section IV provides concluding remarks and future works.

II. Literature Review

A. Game-Theoretical Approaches to SDN Security

Recently, several game-theoretical approaches have been proposed for SDN security with emphasis on the SDN controller assignment [11], [12], [13] as well as the moving target defense [14]. Wang et. al. [11] proposed a novel two-phase dynamic SDN controller assignment mechanism to minimize the average response time of the control plane. The assignment

between controllers (with various capacities to serve requests) and switches (with different request demands) is considered as the stable matching problem in the first phase. The solution quality from the first phase (the mapping between switches and controllers that guarantees the worst-case response time for each switch) is then improved by leveraging the coalitional game theory. A group of switches that are assigned to a controller can be seen as a coalition and they can negotiate to change their coalitions to improve the response time.

In [12], Chen et. al. proposed a zero-sum game-theoretical solution for the controller load-balancing. In this model, controllers are the players and SDN switches are the commodities that are traded among the players who intend to maximize their profits (e.g., by load balancing). An overloaded controller selects a switch and then sends an announcement (i.e., the existing load of the selected switch) to its nearby controllers so that they can compete to be the new master controller of the selected switch.

In [13], an optimal multi-controllers placement is considered as a multi-objective optimization problem that minimizes the latency and communication overhead between switches and a controller. It also ensures the load balancing among controllers. A cooperative Nash bargaining game theory is used to find the trade-off between two conflicting objectives. These two objectives are considered in order to find a unique solution that satisfies the Pareto efficiency between both players.

Finally, in [14], Jafarian et. al. model the interaction between a defender (that proactively defends against DoS attacks through a random route mutation mechanism) and a DoS attacker as a static game of complete information. In this model, an attacker is aware of the flow properties (for instance, source and destination, size and duration, transmission starting time) and its strategy is to attack a number of routes during the flow transmission. The defender's strategy is to use a number of routes for the flow transmission. The aim of each player (that is, defender and attacker) is to determine its Nash Equilibrium strategy by taking into account the opponent strategy and the cost of its own strategy.

B. SDN-Based MTD

The static nature of existing network attributes (e.g., IP address and route to certain network hosts) enables attackers to perform the network reconnaissance without any time constraint. The network-based MTD has been intensively studied to obfuscate attackers' reconnaissance efforts by changing the network attributes randomly and periodically in order to make it harder for the attackers to collect useful information, that is, to increase the attackers' overheads significantly while minimizing the legitimate users' overheads. The emerging SDN has been employed for efficient and cost-effective network-based MTD operations [4], [5], [15], [8], [14], [16], [17]. However, the roles of the SDN switch and SDN controller are varied among the proposed SDN-based MTD approaches.

In proactive MTD-based address mutation approaches (in which the real address of a moving target network host remains untouched and a short-lived virtual address is associated to that host dynamically [4], [5], [15]), the SDN switch can be used as the address translator between these real and virtual addresses [5], [15]. On the other hand, the SDN controller has more complex tasks. In [4], while there is no additional function for the SDN switch, the SDN controller acts as a generator of the synthetic MAC and IP addresses, and also, informs the server application to create Network Address Translation (NAT) rule to map the synthetic and real addresses. In [5], the SDN controller has the following roles: coordinating mutation across the SDN switches, determining the optimal set of new virtual addresses for hosts using Satisfiability Module Theories (SMT) solver, and finally, handling the DNS updates. In [15], besides proactive and reactive IP-address randomizations, the SDN controller is responsible to learn the topology and also assign a random flow to the traffic because the IP-address randomization is still prone to traffic analysis.

Proactively modifying the traffic flow through route mutation is also performed in [14], [8]. Besides performing route mutation, the SDN controller in [14] is responsible to determine the optimal defender strategy by finding the Nash Equilibrium of the game and also the qualified routes for this strategy by using the SMT solver. In [8], route mutation is utilized to increase the attacker's cost of finding persistent links. The SDN controller is used to create traceroute profiles by monitoring the ICMP traffic, and performs route mutation in response to the identified traceroute accordingly. In [16], the SDN is employed to monitor the transport layer traffic (e.g., TCP) and generate random TCP responses and payloads for the illegitimate TCP scanning traffic to prevent operating system fingerprinting. In [17], the SDN is employed to modify the detected network scanning traffic flow to a shadow network that provides a response to this network scanning attempt.

In all of these approaches, the SDN controller and switches are assumed to be trusted and no collusion is considered between the SDN elements and the attackers. In this paper, we will drop such an assumption to propose a game-theoretical model to address collusion between the switches and attackers.

III. Our Game-Theoretical Construction

Game-theoretical paradigms are mostly used to model interaction between attackers and defenders [18], [19]. In these models, a two-player game is proposed in which attackers and defenders try to maximize the utility that they can gain. For instance, the defenders can provide value to the system and, as a result, gain utility by enabling features, shifting the attack surface, and reducing the attack surface measurement. On the other hand, the attackers can benefit if features are disabled or the attack surface measurement is increased.

In majority of existing models, an attacker and a defender play the game by selecting different actions from their action profiles in each round of the game (for instance, the defender can modify the system in order to shift the attack surface or the attacker can manipulate the system in order to disable some features). After each selection, the system moves to a new state and the players receive their rewards based on a reward function, also known as utility function.

A. Game Theory Preliminaries

A *game* consists of a set of *players*, a set of *actions* and *strategies* (that is, the way of selecting actions in different rounds of the game), and finally, a *pay-off function* which is used by the players to calculate their utilities. In *cooperative games*, the players collaborate and split the total utility among themselves. In other words, cooperation is always enforced by agreements among the players. However, in *non-cooperative games*, the players cannot reach an agreement to coordinate their behavior, that is, any cooperation must be self-enforcing. Next, some game-theoretic definitions are briefly reviewed [20] for our further technical discussions.

Definition-1: Let $A \stackrel{\text{def}}{=} A_1 \times \cdots \times A_n$ be an action profile for n players, where A_i denotes the set of possible actions for player S_i. A *game* $\Gamma = (A_i, u_i)$ for $1 \leq i \leq n$, consists of A_i and a utility function $u_i : A \mapsto \mathbb{R}$ for each player S_i. An *outcome of the game* is then a vector of actions $\vec{a} = (a_1, \ldots, a_n) \in A$.

Definition-2: The *utility function* u_i illustrates the preferences of player S_i over different outcomes. We say player S_i *prefers* outcome $\vec{a}$ over $\vec{a}'$ iff $u_i(\vec{a}) > u_i(\vec{a}')$, and he *weakly prefers* outcome $\vec{a}$ over $\vec{a}'$ if $u_i(\vec{a}) \geq u_i(\vec{a}')$.

In order to allow the players S_i to follow randomized strategies (where the strategy is the way of choosing actions), we define σ_i as a probability distribution over A_i for a player S_i. This means the player samples $a_i \in A_i$ according to σ_i. A strategy is said to be a *pure-strategy* if each σ_i assigns probability 1 to a certain action, otherwise, it is said to be a *mixed-strategy*. Let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ be the vector of players' strategies, and let $(\sigma_i', \vec{\sigma}_{-i}) \stackrel{\text{def}}{=} (\sigma_1, \ldots, \sigma_{i-1}, \sigma_i', \sigma_{i+1}, \ldots, \sigma_n)$, where player S_i replaces σ_i by σ_i' and all the other players' strategies remain unchanged. Therefore, $u_i(\vec{\sigma})$ denotes the expected utility of S_i under the strategy vector $\vec{\sigma}$. A player's goal is to maximize $u_i(\vec{\sigma})$. In the following definition, one can substitute an action $a_i \in A_i$ with its probability distribution σ_i or vice versa.

Definition-3: A vector of strategies $\vec{\sigma}$ is a *Nash Equilibrium* if, for all i and any $\sigma_i' \neq \sigma_i$, it holds that $u_i(\sigma_i', \vec{\sigma}_{-i}) \leq u_i(\vec{\sigma})$. This means no one gains any advantage by deviating from the protocol as long as the others follow the protocol.

B. Model Description

Our model is constructed upon the SDN-based MTD [8] that strives to provide route mutation defense against the link-map creation at the reconnaissance stage of the crossfire attack [7]. It is a powerful attack that degrades and cuts off network connections of selected server targets, e.g., a link-flooding DDoS that attacks links surrounding a target. Reconnaissance phase is the first and the longest step in which the attacker strives to find persistent links that can be candidates for the targeted link-flooding DDoS attacks. The persistent link is a link that always presents whenever an attacker performs the reconnaissance, as opposed to transient links. By performing the route mutation, when a suspected reconnaissance attempt is detected, an attacker is expected to receive a transient link instead of a persistent link.

Our model consists of an SDN controller that assigns a flow rule to every switch based on the selected route mutation strategy and n switches that act as defenders. On the other hand, we have a group of attackers that try to collude with switches so that (the following actions are considered as defection in our setting):

1) They do not perform the route mutation; therefore, the attacker can find persistent links.
2) They send their traffic to certain links in order to help the attacker to launch the DDoS attack.

The following two scenarios are considered for collusion: a single switch is not trusted and colludes with the attacker, or multiple switches are not trusted and form collusion with an attacker. We consider the later case as it is the general case of the first scenario. Note that game-theoretical paradigms are usually used to model *interaction* between defenders and attackers. Here, we specifically intend to model *collusion* between defenders and attackers.

In our new game-theoretical model, we first consider a 2-player game between two defenders (i.e., switches) that may/may not collude with an attacker by not performing the route mutation or sending the traffic to certain links. These two actions are part of the players' action profiles and they will be considered as defection, denoted by $\mathscr{D}$. As such, cooperative actions, denoted by $\mathscr{C}$, are considered to be performing the route mutation or sending the traffic to different links.

C. Our Solution in a Nutshell

We consider the following payoff function for two switches similar to the prisoners' dilemma, shown in Table I. Note that this model can be easily extended to a model with n switches.

TABLE I
TWO DEFENDERS WHO INTEND TO COLLUDE WITH THE ATTACKER

S_1 \ S_2	$\mathscr{C}$: Not Collude	$\mathscr{D}$: Collude
$\mathscr{C}$: Not Collude	$(0,0)$	$(0,2)$
$\mathscr{D}$: Collude	$(2,0)$	$(1,1)$

This model illustrates, if both switches collude with the attacker, they each gain, e.g., \$1 utility (i.e., attacker's \$2 budget will be shared between both switches) but if one switch colludes but the other one doesn't collude, the colluder will receive \$2 from the attacker. As a result, collusion is Nash Equilibrium meaning that switches always collude because it's in their best interest to do so. This is a realistic scenario in which an attacker with a limited budget tries to compromise components of a network by colluding with defenders.

We tackle the aforementioned problem by considering a socio-rational model [9], [10] (that is, a repeated game among rational players who have public reputation values where these values affect players' utilities overtime) in which:

1) The SDN controller selects a group of switches (a subset of switches based on their trust values using a non-uniform probability distribution) to protect the targeted system against potential attacks.
2) The attacker utilizes his budget in order to collude with switches, and consequently, compromise the system.

In our setting, if a switch colludes with the attacker, it can gain some utility in the current game (e.g., \$1), however, that switch has less chance (lower probability) to be selected by the SDN controller in the future games due to the reduction of his reputation value, see [21], [22] for a trust/reputation management system. Therefore, it would be in the best interest of switches not to collude with the attacker because a non-cooperative switch will lose his reputation, and consequently, he will lose many future games (e.g., $-$\$3).

D. Formalizing Our Solution

We utilize a trust management scheme in a repeated two-player game between "two defenders" who try to maximize their utilities through collusion with the attackers. We show that, by using proper strategies, cooperation (i.e., not-colluding with the attackers) is always Nash Equilibrium because of a long-term utility that we consider in our game-theoretical setting. We not only consider a reward function but also use a function to penalize colluders. We also consider two classes of actions, that is, *collude* as non-cooperative actions and *not collude* as cooperative actions. E.g., disabling features and increasing the attack surface are actions from the first class.

Our game is repeatedly played for an unknown number of rounds. Each network switch S_i has a public reputation value $\mathscr{R}_i$, where the initial value is zero, i.e., $\mathscr{R}_i(0) = 0$, and it is bounded as follows $-1 \leq \mathscr{R}_i(p) \leq +1$; note that $p = 0, 1, 2, \ldots$ denotes subsequent rounds of the game. Moreover, each switch's action $a_i \in \{\mathscr{C}, \mathscr{D}, \perp\}$, where $\mathscr{C}$ and $\mathscr{D}$ denote *cooperation* and *defection* respectively, and $\perp$ denotes S_i has not been chosen by the SDN controller in the current game. Finally, each switch calculates two utility functions to decide whether he should collude with the attacker or not, i.e., a long-term utility function u_i and an actual utility function u'_i. Each round of the game consists of the following steps:

1) Let Ψ be a non-uniform probability distribution over types of switches, i.e., good/non-colluding, bad/colluding and new switches. The SDN controller selects m out of n switches, where $m \leq n$, based on this probability distribution in each round of the game.
2) Each switch S_i computes his long-term utility function $u_i : A \times \mathscr{R}_i \mapsto \mathbb{R}$, and then selects an action from the action profile A, i.e., whether to collude with the attacker or not.
3) Each player S_i receives his utility $u'_i : A \mapsto \mathbb{R}$ (that is, the real utility that each switch can gain) at the end of each round of the game according to the outcome.
4) The reputation values $\mathscr{R}_i$ of all the chosen switches are publicly updated based on each switch's behavior using a reputation system. Note that the SDN controller doesn't know if a switch has colluded with the attacker at each round of the game, however, if a switch deviates from the SDN controller's instructions (e.g., he does not perform the rout mutation or sends the traffic to certain links), it will be assumed that he has colluded with the attacker.

E. Utility Assumptions

Let $u_i(\vec{a})$ denote S_i's long-term utility in outcome $\vec{a}$ by considering current and future games, let $u'_i(\vec{a})$ denote S_i's short-term utility in outcome $\vec{a}$ in the current game, let $c_i(\vec{a}) \in \{0, 1\}$ denote if S_i has colluded with the attacker in the current game, and define $\Delta(\vec{a}) = \sum_i c_i(\vec{a})$, that is, the number of switches/defenders who have colluded with the attacker. Let $\mathscr{R}_i^{\vec{a}}(p)$ denote the reputation of S_i after outcome $\vec{a}$ in period p; note that $\vec{a}$ and $\vec{a}'$ are two different outcomes of the game. The following preferences are considered in our setting:

- $c_i(\vec{a}) = c_i(\vec{a}')$ and $\mathscr{R}_i^{\vec{a}}(p) > \mathscr{R}_i^{\vec{a}'}(p) \Rightarrow u_i(\vec{a}) > u_i(\vec{a}')$.
- $c_i(\vec{a}) > c_i(\vec{a}') \Rightarrow u'_i(\vec{a}) > u'_i(\vec{a}')$.
- $c_i(\vec{a}) > c_i(\vec{a}')$ and $\Delta(\vec{a}) < \Delta(\vec{a}') \Rightarrow u'_i(\vec{a}) > u'_i(\vec{a}')$.

The first assumption states that each switch S_i prefers to sustain a high reputation value overtime despite of colluding or not colluding with the attacker as he can potentially gain a higher long-term utility. The second assumption expresses that if a switch S_i colludes with the attacker, he gains a short-term utility. Finally, the third assumptions illustrates that if a switch S_i colludes with the attacker and the total number of colluding parties in $\vec{a}$ is less than the total number of colluding parties in $\vec{a}'$, he gains a higher short-term utility in outcome $\vec{a}$.

F. Utility Function and Mathematical Analysis

The long-term utility function $u_i : A \times \mathscr{R}_i \mapsto \mathbb{R}$ calculates the utility that each switch S_i potentially gains or loses by taking into account both current and future games (based on all three utility assumptions), whereas the short-term utility function $u'_i : A \mapsto \mathbb{R}$ only calculates the current gain or loss in a given period (based on the last two utility assumptions). Note that A is the action profile.

Let ϕ_i be the reward coefficient that is defined by the SDN controller based on the reputation value of each switch S_i, and let $\delta_i(\vec{a}) = \mathscr{R}_i^{\vec{a}}(p) - \mathscr{R}_i^{\vec{a}}(p-1)$ be the difference of two consecutive reputation values. Note that $\tau_i = |\delta_i(\vec{a})| / \delta_i(\vec{a})$ is positive if the selected action in period p is $\mathscr{C}$ and it is negative, if it is $\mathscr{D}$. Also, let $\Omega > 0$ be a unit of utility, e.g., \$100. To satisfy the stated assumptions in Section III-E, we have the following equations:

$$\frac{|\delta_i(\vec{a})|}{\delta_i(\vec{a})} \times \phi_i \times \Omega \tag{1}$$

$$c_i(\vec{a}) \times \Omega \tag{2}$$

$$\frac{c_i(\vec{a})}{\Delta(\vec{a}) + 1} \times \Omega \tag{3}$$

- Eqn (1) means S_i gains or loses ϕ_i units of utility Ω in the future games due to his behavior as reflected in $\mathscr{R}_i$.
- Eqn (2) illustrates that S_i gains one unit of utility if he colludes with the attacker in the current game and he loses this opportunity, otherwise.
- Eqn (3) results in almost one unit of utility to be divided among all the colluders.

The linear combination of these terms defines the long-term utility function $u_i(\vec{a})$, however, actual utility $u_i'(\vec{a})$ only consists of the linear combination of equations (2) and (3).

$$u_i(\vec{a}) \quad = \quad \Omega\left(\frac{|\delta_i(\vec{a})|}{\delta_i(\vec{a})} \times \phi_i + c_i(\vec{a}) + \frac{c_i(\vec{a})}{\Delta(\vec{a})+1}\right).$$

Theorem-1: *In a $(2,2)$-socio-rational collusion game, $\mathscr{C}$ strictly dominates $\mathscr{D}$ when we use our utility function.*

Proof: We compute the utility of each outcome for S_i. Let S_j be the other defender.

1) If both defenders don't collude/cooperate, then δ_i is positive, $c_i = 0$, and $\Delta = 0$:

$$\big(\delta_i > 0, c_i = 0, \Delta = 0\big) \Rightarrow u_i^{(\mathscr{C},\mathscr{C})} = \Omega\,\phi_i.$$

2) If only S_i cooperates, then δ_i is positive, $c_i = 0$ since S_i has not colluded, and $\Delta = 1$ because only switch S_j has colluded with the attacker:

$$\big(\delta_i > 0, c_i = 0, \Delta = 1\big) \Rightarrow u_i^{(\mathscr{C},\mathscr{D})} = \Omega\,\phi_i.$$

3) If only S_j cooperates, then δ_i is negative, $c_i = 1$ since S_i has colluded, and $\Delta = 1$:

$$\big(\delta_i < 0, c_i = 1, \Delta = 1\big) \Rightarrow u_i^{(\mathscr{D},\mathscr{C})} = \Omega\Big(-\phi_i + 1.50\Big).$$

4) If both switches defect, then δ_i is negative, $c_i = 1$, and $\Delta = 2$ because both switches have colluded:

$$\big(\delta_i < 0, c_i = 1, \Delta = 2\big) \Rightarrow u_i^{(\mathscr{D},\mathscr{D})} = \Omega\Big(-\phi_i + 1.33\Big).$$

If reward factor $\phi_i \geq 1.5$, we will have the following payoff inequalities that proves our theorem:

$$\overbrace{u_i^{(\mathscr{C},\mathscr{C})}(\vec{a}) = u_i^{(\mathscr{C},\mathscr{D})}(\vec{a})}^{S_i \text{ cooperates}} > \overbrace{u_i^{(\mathscr{D},\mathscr{C})}(\vec{a}) > u_i^{(\mathscr{D},\mathscr{D})}(\vec{a})}^{S_i \text{ defects}} \quad \square$$

If we assume the reward factor ϕ_i is at least 1.5 (note that the minimum value of this constant is defined based on the model's parameters), the payoff matrix is as follows, Table II:

TABLE II
$(2,2)$-COLLUSION GAME BETWEEN TWO DEFENDERS IN THE SOCIO-RATIONAL MODEL

S_1 \ S_2	$\mathscr{C}$: Not Collude	$\mathscr{D}$: Collude
$\mathscr{C}$: Not Collude	$(1.5, 1.5)$	$(1.5, 0)$
$\mathscr{D}$: Collude	$(0, 1.5)$	$(-0.17, -0.17)$

As you can see, cooperation is always Nash Equilibrium. To expand our proof to a case with n switches/defenders, let $\mathscr{C}_i$ (or $\mathscr{D}_i$) denote that S_i cooperates (or defects), and let $\mathscr{C}_{-i}$ (or $\mathscr{D}_{-i}$) denote that, excluding S_i, all the other defenders cooperate (or defect), and finally, let $\mathscr{M}_{-i}$ denote that, excluding S_i, some defenders cooperate and some of them defect.

Theorem-2: *In a (n,n)-socio-rational collusion game, $\mathscr{C}$ strictly dominates $\mathscr{D}$ when we use our utility function.*

Proof: We compute the utility of each outcome in six different scenarios. Let $n > k \geq 2$.

1) If all the defenders cooperate, or S_i and $k-1$ defenders cooperate, or only S_i cooperates, as a result, δ_i is positive, $c_i = 0$, and $\Delta \in \{0, n-k, n-1\}$:

$$\big(\delta_i > 0, c_i = 0, \Delta \in \{0, n-k, n-1\}\big) \Rightarrow$$
$$u_i^{(\mathscr{C}_i,\mathscr{C}_{-i})} = u_i^{(\mathscr{C}_i,\mathscr{M}_{-i})} = u_i^{(\mathscr{C}_i,\mathscr{D}_{-i})} = \Omega\,\phi_i.$$

2) If only S_i defects, δ_i is negative, $c_i = 1$ and $\Delta = 1$:

$$\big(\delta_i < 0, c_i = 1, \Delta = 1\big) \Rightarrow u_i^{(\mathscr{D}_i,\mathscr{C}_{-i})} = \Omega\Big(-\phi_i + 1.5\Big).$$

3) If S_i as well as $k-1$ defenders defect, and the rest of them cooperate:

$$\big(\delta_i < 0, c_i = 1, \Delta = k\big) \Rightarrow u_i^{(\mathscr{D}_i,\mathscr{M}_{-i})} = \Omega\Big(-\phi_i + \frac{k+2}{k+1}\Big).$$

4) If all the defenders defect, δ_i is negative, $c_i = 1$, and $\delta = n$ because no one has cooperated:

$$\big(\delta_i < 0, c_i = 1, \Delta = n\big) \Rightarrow u_i^{(\mathscr{D}_i,\mathscr{D}_{-i})} = \Omega\Big(-\phi_i + \frac{n+2}{n+1}\Big).$$

We simply analyze these scenarios as follows. Let $*_{-i}$ be $\mathscr{C}_{-i}$ or $\mathscr{M}_{-i}$ or $\mathscr{D}_{-i}$. It is easy to show that:

$$1.5 > \frac{k+2}{k+1} > \frac{n+2}{n+1} \text{ when } n > k \geq 2.$$

Similarly, if we assume the reward factor ϕ_i is at least 1.5, cooperation (i.e., not colluding with the SDN controller) is always Nash Equilibrium. As a result, it is always in S_i's best interest to cooperate no matter what other parties do:

$$u_i^{(\mathscr{C}_i,*_{-i})}(\vec{a}) > u_i^{(\mathscr{D}_i,*_{-i})}(\vec{a}) \qquad \square$$

IV. Conclusion and Future Direction

We showed that a game-theoretical solution concept can be utilized to tackle the collusion attack in a SDN-based framework. In our proposed setting, the defenders (i.e., switches) were incentivized not to collude with the attackers in a repeated-game setting that utilizes a reputation system. We first illustrated our model and its components. We then used a socio-rational approach to provide a new anti-collusion solution that shows cooperation with the SDN controller is always Nash Equilibrium due to the existence of a long-term utility function in our model.

As our future work, we are interested in constructing new game-theoretical paradigms to model cyber deception and collusion risks in SDN-based platforms. Similarly, we will consider a SDN controller, a set of switches that act as defenders, and a group of attackers who intend to compromise different parts of the network. The defenders' goal will be the protection of the targeted network by utilizing appropriate deceptive as well as anti-collusion strategies. Subsequently, the defenders will gain utility if they select proper actions from the action profile in order to deceive the attackers.

We intend to provide new solution concepts in which the attackers are incentivized by extra utility in order to act according to the defenders' strategies (i.e., deception). We will construct a two-player game between "two attackers" and show that, in our model, selection of the deceptive action(s) is Nash Equilibrium. We will further extend this two-player game to a game with any number of attackers. In other words, in our future framework, the SDN controller and defenders deceive the attackers by choosing a certain class of actions that have a higher short-term payoff. Note that, in the presented work here, we considered a two-player game between "two defenders" who may/may not collude with the attackers.

V. Acknowledgment

We would like to thank Florida Center for Cybersecurity for sponsoring this project through the FC2 Seed Grant. We also thank the anonymous reviewers for their constructive feedback and inspiring comments.

References

[1] R. JAIN and S. Paul, "Network virtualization and software defined networking for cloud computing: a survey," *Communications Magazine, IEEE*, vol. 51, no. 11, pp. 24–31, November 2013.

[2] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in *Future Networks and Services, 2013 IEEE SDN for*, Nov 2013, pp. 1–7.

[3] J. Moar, "Cybercrime and the internet of threats," Juniper Research, Tech. Rep., 2015.

[4] D. C. MacFarland and C. A. Shue, "The SDN shuffle: Creating a moving-target defense using host-based software-defined networking," in *Proceedings of the Second ACM Workshop on Moving Target Defense*. ACM, 2015, pp. 37–41.

[5] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*. ACM, 2012, pp. 127–132.

[6] D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *2nd SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013, pp. 55–60.

[7] M. S. Kang, S. B. Lee, and V. D. Gligor, "The crossfire attack," in *Symposium on Security and Privacy (SP)*. IEEE, 2013, pp. 127–141.

[8] A. Aydeger, N. Saputro, K. Akkaya, and M. Rahman, "Mitigating crossfire attacks using SDN-based moving target defense," in *41st IEEE Conference on Local Computer Networks*, 2016, pp. 627–630.

[9] M. Nojoumian and D. R. Stinson, "Socio-rational secret sharing as a new direction in rational cryptography," in *3rd International Conference on Decision and Game Theory for Security (GameSec)*, ser. LNCS, vol. 7638. Springer, 2012, pp. 18–37.

[10] M. Nojoumian, "Generalization of socio-rational secret sharing with a new utility function," in *12th IEEE Annual International Conference on Privacy, Security and Trust, PST'14*, 2014, pp. 338–341.

[11] T. Wang, F. Liu, J. Guo, and H. Xu, "Dynamic SDN controller assignment in data center networks: Stable matching with transfers," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[12] H. Chen, G. Cheng, and Z. Wang, "A game-theoretic approach to elastic control in software-defined networking," *China Communications*, vol. 13, no. 5, pp. 103–109, May 2016.

[13] A. Ksentini, M. Bagaa, T. Taleb, and I. Balasingham, "On using bargaining game for optimal placement of SDN controllers," in *IEEE International Conference on Communications*, May 2016, pp. 1–6.

[14] J. H. Jafarian, E. Al-Shaer, and Q. Duan, *Formal Approach for Route Agility against Persistent Attackers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 237–254.

[15] A. R. Chavez, W. M. S. Stout, and S. Peisert, "Techniques for the dynamic randomization of network attributes," in *International Carnahan Conference on Security Technology*, Sept 2015, pp. 1–6.

[16] P. Kampanakis, H. Perros, and T. Beyene, "SDN-based solutions for moving target defense network protection," in *15th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks*. IEEE, 2014, pp. 1–6.

[17] L. Wang and D. Wu, *Moving Target Defense Against Network Reconnaissance with Software Defined Networking*. Springer International Publishing, 2016, pp. 203–217.

[18] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A survey of game theory as applied to network security," in *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*. IEEE, 2010, pp. 1–10.

[19] X. Liang and Y. Xiao, "Game theory for network security," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 472–486, 2013.

[20] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT press, 1994.

[21] M. Nojoumian and T. C. Lethbridge, "A new approach for the trust calculation in social networks," in *E-business and Telecommunication Networks: 3rd International Conference on E-Business, Best Papers*, ser. CCIS, vol. 9. Springer, 2008, pp. 64–77.

[22] M. Nojoumian, "Novel secret sharing and commitment schemes for cryptographic applications," Ph.D. dissertation, Department of Computer Science, University of Waterloo, Canada, 2012.