A Bayesian Game Decision-Making Model for Uncertain Adversary Types

Mahsa Emami-Taba
Software Technologies Applied Research (STAR) Group
University of Waterloo
Waterloo, Canada
msemamit@uwaterloo.ca

Ladan Tahvildari
Software Technologies Applied Research (STAR) Group
University of Waterloo
Waterloo, Canada
ltahvild@uwaterloo.ca

ABSTRACT

Adaptive application security involves making decisions under uncertainties such as the time, the power, or the damage of potential attacks. One of the uncertainties that has been largely ignored in the literature is the intention of the adversaries. The majority of research focuses on characteristics of *attacks* (e.g., their request arrival rates), whereas characteristics of *attackers/adversaries* (e.g., their intentions and strategies) are neglected. In today's sophisticated attacks, in order to confuse defense systems, adversaries may initiate an attack that exhibits a scenario similar to another attack but has an entirely different malicious goal (e.g., to break down the server or to harm a specific user in the system). In such cases, incorporating uncertainty about the type of adversaries into the decision model helps to choose a proper countermeasure for protecting the software system efficiently. In this paper, we present a Bayesian game model that captures the uncertainty about an adversary's motivation for sending malicious requests. Our game-theoretic model formalizes possible intentions of adversaries along with the security preferences of the software system. In such a novel design, the equilibrium of the modeled game balances the gain from achieving security goals with the loss incurred by mitigating the attack. We provide an extensive analysis of the proposed game-theoretic model in the presence and absence of uncertainty about the adversary type. Moreover, we present a case study to show how such uncertainty can be addressed using the proposed technique in a real-world scenario.

Keywords

Self-Protecting Software, Adversary-Type Uncertainty, Game Theory, Decision-Making Model

1. INTRODUCTION

Preserving the security goals of software systems when confronted with application-layer attacks requires not only detecting the attack, but also responding in a timely and appropriate manner. A manual intrusion response introduces a delay between notification and response, which could be exploited by the adversary to significantly increase damage to the system [23]. Protecting software systems against today's sophisticated attacks calls for intelligent decision-making techniques. However, the decision-making process needs to deal with various unknown features such as uncertainty associated with the satisficement of Non-Functional Requirements (NFRs) given a set of decisions [6] or uncertainty in alert notification [23].

Consider a scenario where malicious requests are sent to the system at a higher-than-normal rate. The intention of the adversary could be either to break the system down by overloading it, or to access as much confidential information as possible in a brief amount of time before being detected by the Intrusion Detection System (IDS). In both attacks, similar malicious behaviors are detected by the IDS, but the proper mitigation for each of the attacks is completely different from the other. Hence, the defense mechanism is required to trigger a response action in a timely manner while taking into account the uncertainty about the type of adversary. In order for the defense mechanism to initiate a proper response action, the uncertainty about the intention of the adversary for intruding in the system should be fused into the decision model. Incorporating the uncertainty about the adversary type into the network protection is studied in *network security*, e.g., network protection [12], Mobile Ad hoc Networks (MANETs) [14], and wireless ad hoc networks [16]. However, to the best of our knowledge, modeling and analyzing such an uncertainty has not been investigated in the field of *software security*.

In this paper, we explore the incorporation of such uncertainty into the decision model of Self-Protecting Software (SPS) systems. We propose a game-theoretic approach that can be easily employed in any SPS system according to its preferences in security goals and the cost of response actions while fusing the cost and benefit of initiating attacks by various types of adversaries. The decision model is based on a Bayesian game model which acknowledges the interactions between the SPS system and the adversary. The decision model considers the uncertainty about the type of adversary by incorporating the probability for each type of adversary. We analyze the achievable Nash equilibrium for both pure and mix strategies for the SPS system and the rational adversary.

© 2016 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

The organization of this paper is as follows: In Section 2, we introduce our proposed game and the notations used in our model. In Sections 3 and 4, we define two normal-form games with complete information about the type of adversary. Following that, in Section 5, we describe a Bayesian game model to formulate our problem with uncertainty about the type of adversary. In Section 6, we present our experimental case study, followed by related work in Section 7. We conclude the paper in Section 8.

2. THE GAME SPECIFICATION

We address the uncertainty about the type of adversary by incorporating such information, along with the benefits and costs of attacks and countermeasures, into the decision model. We consider a two-player normal-form game in which, one player is the adversary, denoted by "a". The other player is the defense system in the SPS, denoted by "d". In this game, the adversary aims at targeting one of the two security goals of the system: G_1, and G_2. For instance, the aim of the adversary could be breaking down the server in order to discount the *availability* goal (G_1) of the SPS system or to access sensitive information and break the *confidentiality* goal (G_2). Each targeted security goal calls for a different type of adversary. For example, the availability goal can be targeted by Denial of Service (DoS) attackers whereas the confidentiality goal is targeted by insider attackers.

Today's sophisticated and well-planned attack scenarios have motivated us to consider attack scenarios in which the adversary adapts more than one pure strategy to mislead the defense system. In this case, a traditional defense system may select a countermeasure that works to the benefit of the adversary. The benefits of game theory in providing a holistic decision making in adaptive security is discussed in [9]. In our game specification, the SPS system protects two security goals: G_1 and G_2. Accordingly, two pure strategies for the adversary are considered: *Attack 1* (AT1) and *Attack 2* (AT2), which target G_1 and G_2, respectively. However, an intelligent adversary may mix these two strategies, aiming at confusing the SPS system. For example, a DoS attacker can mix two pure strategies: (i) *Heavy Load*: sending malicious requests that introduce high workload to the system, and (ii) *High Sensitive*: sending malicious requests that target files with low workload, yet sensitive data. The former strategy is prone to fast detection by the IDS, whereas the latter strategy results in misdiagnosing the attack and applying a countermeasure that is effective for an insider attacker instead of a DoS attack. Therefore, the defense system will fail to treat the adversary as a DoS attacker.

In our game specification, the defense system has two pure strategies to protect the software system: *Countermeasure 1* (CM1) and *Countermeasure 2* (CM2), which protect G_1 and G_2, respectively. For example, two possible countermeasures are (i) *Issue Puzzle*: issuing a puzzle such as CAPTCHA [22] to determine whether or not the original of the request is from a real user or a botnet [13], and (ii) *Drop Request*: providing no respond to the incoming request. These two countermeasures protect the availability and confidentiality goals, respectively. In many situations, the defense system can only select and apply one of the available countermeasures to protect the software system due to the fact that: (i) most of the countermeasures are mutually exclusive, (ii) the combined cost of applying countermeasures exceeds the value of the software protected, or (iii) the combination of countermeasures degrades the level of service to an unacceptable level.

Table 1: Summary of Notations Used

Notation	Definition
a	Adversary (Player)
d	Defense System (Player)
ATi	Attack i (Action)
CMi	Countermeasure i (Action)
C_x	Cost of applying action x
Ti	Adversary type i
M_{ATj}^{Ti}	Measure of benefit to the Ti type adversary to achieve its malicious goal via ATj
Gi	Security goal i
S_{Gi}	Measure of benefit to the defense system to satisfy security goal Gi
p	Probability with which the defense system plays CM1
$1-p$	Probability with which the defense system plays CM2
q_i	Probability with which the type i adversary plays AT1
$1-q_i$	Probability with which the type i adversary plays AT2
μ	Probability of an adversary of type 1
$1-\mu$	Probability of an adversary of type 2
$E_{Player}(Action)$	Expected payoff of a *Player* to play an *Action*

In our modeled games, two players choose strategies simultaneously under the assumption that both have common knowledge about the cost and benefits of the game. Table 1 summarizes the notations used in our game models: C_x denotes the cost of actions for the players (the adversary and the defense system), M_{ATj}^{Ti} represents the gain by the malicious user (adversary), and S_{Gi} denotes the gain by the SPS system. For both players, all possible strategies incur some cost, which can be interpreted as the operational cost to conduct the strategy. The benefits of goals G_1 and G_2 are represented by S_{G1} and S_{G2} and are defined by the stakeholders. In the defined game model, the decision factors are abstract enough to adapt to the required application. For example, the values of S_{G1} and S_{G2} can be dependent on the type of application and the preferences over security goals in the system. In Table 1, it is reasonable to assume: (i) $S_{G1}, S_{G2} > C_{CM1}$, (ii) $S_{G1}, S_{G2} > C_{CM2}$, (iii) $M_{AT1}^{T1}, M_{AT2}^{T1} > C_{AT1}$, and (iv) $M_{AT1}^{T1}, M_{AT2}^{T1} > C_{AT2}$, since otherwise the defense system does not have the incentive to defend the system and the adversary does not have the incentive to attack. The cost of response actions (C_{CM1} and C_{CM2}) can be defined as a function of resource consumption with respect to activities to enable and process the response action. The cost of AT1 (C_{AT1}) and AT2 (C_{AT2}) can be defined as a function of resource consumption to initiate the attack.

In our modeled games, the payoffs of players represents the cost and benefit of actions, using their impact on quality goals. Incorporating such information into modeling the interdependencies of strategies between the SPS system and the adversary is proposed in [10]. Here, we formulate this information in order to incorporate the type of the adver-

sary. The following two sections describe two normal-form games [19] with complete information about the type of adversary. In each game, we model the payoffs of players and analyze the Nash equilibrium solution of the game along with a numerical example.

3. TYPE 1 ADVERSARY

In this complete information game, we consider the scenario in which the intention of the adversary is to target the G_1 security goal (a type 1 adversary). Hence, the adversary initiates AT1 along with AT2 to mislead the other player (the defense system). Table 2 lists the payoffs. Each cell in Table 2 has two payoffs that correspond with player d and player a accordingly. The notations that represent the gain and cost of players are defined in Table 1.

Table 2: Strategic Form of Type 1 Adversary vs. SPS

$d \setminus a$	AT1	AT2
CM1	$S_{G1} - C_{CM1}$, $-M^{T1}_{AT1} - C_{AT1}$	$-S_{G2} - C_{CM1}$, $-M^{T1}_{AT2} - C_{AT2}$
CM2	$-S_{G1} - C_{CM2}$, $M^{T1}_{AT1} - C_{AT1}$	$S_{G2} - C_{CM2}$, $M^{T1}_{AT2} - C_{AT2}$

For the SPS system, satisfaction of the two security goals G_1 and G_2 are denoted by S_{G1} and S_{G2}. The cost of applying CM1 is denoted by C_{CM1}. In the case of pure strategy AT1, applying CM1 can successfully address the attack and G_1 security goal is protected with this strategy. Therefore, the payoff of the SPS system is $S_{G1} - C_{CM1}$. Applying CM2 costs C_{CM2} and CM2 does not properly satisfy the security goal of G_1. For example, dropping the request will not stop the adversary from sending malicious requests. The continuation of such requests eventually results in threatening G_1 security goal (such as the availability goal). Hence, the payoff for CM2 in case of AT1 is $-S_{G1} - C_{CM2}$.

When an adversary wanting to target G_1 security goal issues AT2 in order to stay hidden from the IDS, the strategy CM1 puts security goal G_2 at risk, as represented by $-S_{G2}$. In this case, the response action of selecting CM1 results in a payoff of $-S_{G2} - C_{CM1}$, whereas selection of CM2 results in protection of security goal G_2 and the payoff of $S_{G2} - C_{CM2}$.

Now we turn to the adversary payoffs. The adversary incurs the cost of C_{AT1} by issuing AT1 and gains M^{T1}_{AT1} if the malicious intention to threat the G_1 security goal is satisfied. If the adversary fails, the gain of the adversary is $-M^{T1}_{AT1}$. Here, it can be seen that the success of an attack depends on the strategy selected by the SPS system. Only when SPS chooses CM2 will the adversary succeed ($M^{T1}_{AT1} - C_{AT1}$).

The adversary could also choose AT2 to misguide the SPS system and encourage it to change its strategy to CM2. The gain in this case is M^{T1}_{AT2}, which is less than M^{T1}_{AT1} because the adversary reaches its malicious goal quicker by issuing AT1 instead of AT2. Hence, we have: $M^{T1}_{AT1} > M^{T1}_{AT2}$. AT2 also introduces extra effort for the adversary that is intended to target G_1 security goal. This effort is denoted by C_{AT2}.

3.1 Nash Equilibrium Analysis

The well-known concept of Nash equilibrium [19] defines a set of actions for players such that none have any incentive to deviate from their chosen action. Assuming that the defense system always takes the pure strategy CM1, then the adversary's best response is to select AT2 when $-M^{T1}_{AT1} - C_{AT1} \leq -M^{T1}_{AT2} - C_{AT2}$. However, this is not an equilibrium, as the pure strategy of AT2 by the adversary motivates the rational defense system to change its strategy to CM2. By switching to CM2, the adversary is inclined to change its strategy to AT1 when $M^{T1}_{AT1} - C_{AT1} \geq M^{T1}_{AT2} - C_{AT2}$. Since we have $M^{T1}_{AT1} > M^{T1}_{AT2}$, this can be interpreted thus: if the defense system applies CM2 then the adversary issues AT1 when $C_{AT1} \leq M^{T1}_{AT1} - M^{T1}_{AT2} + C_{AT2}$.

The above finding is interesting in the decision-making of SPS systems. It suggests that in selecting a response action, the decision-making engine needs to incorporate the cost of *initiating an attack* for adversaries rather than considering the cost of *applying a countermeasure* for the defense system. This is contrary to most approaches for SPS systems, where the decision-making engine considers only the operational cost of providing a response action. The application of such a decision-making engine can introduce a perspective system (e.g., by focusing on increasing the cost of mounting attacks for adversaries) rather than simply deploying a response action.

3.2 Mixed Strategy Equilibrium Analysis

In the previous subsection, we found that there are two pure strategies: (i) (CM1, AT1) when $-M^{T1}_{AT1} - C_{AT1} > -M^{T1}_{AT2} - C_{AT2}$, and (ii) (CM2, AT2) when $M^{T1}_{AT1} - C_{AT1} < M^{T1}_{AT2} - C_{AT2}$. Hence, there is no pure strategy Nash equilibrium when $-M^{T1}_{AT1} - C_{AT1} \leq -M^{T1}_{AT2} - C_{AT2}$ and $M^{T1}_{AT1} - C_{AT1} \geq M^{T1}_{AT2} - C_{AT2}$. Now, we check for Nash equilibrium when the adversary plays a mixed strategy. A *mix strategy* is randomizing over the set of available actions according to some probability distribution [20]. The expected payoffs of the defense system d are as follows (The notations that represent the probabilities (q_1, $1-q_1$, p, and $1-p$) are defined in Table 1)).

$$E_d(CM1) = q_1(S_{G1} - C_{CM1}) + (1-q_1)(-S_{G2} - C_{CM1}), \quad (1)$$

$$E_d(CM2) = q_1(-S_{G1} - C_{CM2}) + (1-q_1)(S_{G2} - C_{CM2}). \quad (2)$$

To make CM1 and CM2 indifferent to the defense system, i.e., $E_d(CM1) = E_d(CM2)$, the adversary's equilibrium strategy is to play AT1 with $q_1 = \frac{S_{G2} - C_{CM2} + S_{G2} + C_{CM1}}{2S_{G1} + 2S_{G2}}$. Similarly, the expected payoffs of the adversary are

$$E_a(AT1) = p(-M^{T1}_{AT1} - C_{AT1}) + (1-p)(M^{T1}_{AT1} - C_{AT1}), \quad (3)$$

$$E_a(AT2) = p(-M^{T1}_{AT2} - C_{AT2}) + (1-p)(M^{T1}_{AT2} - C_{AT2}). \quad (4)$$

By a similar calculation it can be shown that to make AT1 and AT2 indifferent to the adversary, i.e., $E_a(AT1) = E_a(AT2)$, the defense system's equilibrium strategy is to play CM1 with probability $p = \frac{M^{T1}_{AT2} - C_{AT2} - M^{T1}_{AT1} + C_{AT1}}{2M^{T1}_{AT2} - 2M^{T1}_{AT1}}$. It is noteworthy that deploying a countermeasure according to probability p is based on the given circumstances and has a risk of applying the wrong countermeasure.

3.3 Case Based Analysis

Using the defined game model, in the following, we consider two possible cases: (i) an SPS system with a much

Table 3: Payoffs and Numerical Examples of Type 1 Adversary

Table 3.a: Payoffs of Type 1 Adversary in G_1 Preferred SPS

$d \setminus a$	AT1	AT2
CM1	U^{++}, U^{--}	U^{-}, U^{-}
CM2	U^{--}, U^{++}	U^{+}, U^{+}

Table 3.b: Numerical Example of Type 1 Adversary in G_1 Preferred SPS

$d \setminus a$	Heavy Load	High Sensitivity
Issue Puzzle	$85, -100$	$-15, -90$
Drop Request	$-95, 85$	$5, 50$

Table 3.c: Payoffs of Type 1 Adversary in G_2 Preferred SPS

$d \setminus a$	AT1	AT2
CM1	U^{+}, U^{--}	U^{--}, U^{-}
CM2	U^{-}, U^{++}	U^{++}, U^{+}

Table 3.d: Numerical Example of Type 1 Adversary in G_2 Preferred SPS

$d \setminus a$	Heavy Load	High Sensitivity
Issue Puzzle	$5, -100$	$-95, -90$
Drop Request	$-15, 85$	$85, 50$

stronger preference to protect G_1 security goal than G_2 security goal, and (ii) an SPS system with a much stronger preference to protect G_2 security goal than G_1 security goal. We provide a numerical example and analyze potential responses for each case.

- **Case 1: Type 1 Adversary vs. SPS System with G_1 Preference**

  Assuming the benefit to protect G_1 is much higher than that for G_2, we have: $S_{G1} >> S_{G2}$. The payoffs for the defense system d and the adversary a are as shown in Table 3.a. Here, U represents the outcome related to Table 2 while shows the the magnitude of the gain/lost in the payoff value.

  Table 3.b exemplifies a type 1 adversary versus an SPS system with a high preference for G_1 security goal. The calculated mix strategy Nash equilibrium is: $p = 0.78$, $1 - p = 0.22$, $q_1 = 0.10$, and $1 - q_1 = 0.90$.

  The resulting Nash equilibrium illustrates that in the mix strategy the response action CM1 has a much higher probability to be selected than the response action CM2. The rational adversary is motivated to attack with the probability of 0.10, due to the observation of mix strategy by the defense system d, which selects CM1 with the probability of 0.90. Hence, the rational adversary is discouraged from attacking the SPS system.

- **Case 2: Type 1 Adversary vs. SPS System with G_2 Preference**

  The second case considers a low preference of G_1 security goal and high preference of G_2 security goal ($S_{G1} << S_{G2}$). Payoffs for both players are shown in Table 3.c. Table 3.d provides a numerical example of such a scenario. We get a mix strategy Nash equilibrium for the game with $p = 0.78$, $1 - p = 0.22$, $q_1 = 0.90$, and $1 - q_1 = 0.10$. Accordingly, the rational response action for the defense system against AT1 is to choose CM1 with a much higher probability than CM2.

Note that the mix strategy response action selections (p & $1 - p$) in both cases are the same, because the defense system selects a response action by considering the goal of the adversary and the cost of the attack. In both cases, the intention behind the attacks and their cost are the same. Hence, the defense system takes the same strategy in protecting security goals. Comparing the two cases reveals that in case 1 the rational adversary chooses AT1 with less probability. This is motivated by the possibility of the defense system benefiting greatly id CM1 is chosen.

4. TYPE 2 ADVERSARY

In the case of the adversary targeting the G_2 security goal, we model a normal form game in Table 4. The notations that represent the gain and cost of players are defined in Table 1. The payoffs for the defense system are based on the gain/lost of G_1 and G_2 security goals, and the cost of response actions accordingly. The payoffs for the player d in Table 4 are the same as in Table 2. However, in this scenario, the payoffs for the adversary differ from those in the game modeled in the previous section, due to the changed intention of the adversary.

Table 4: Strategic Form of Type 2 Adversary vs. SPS

$d \setminus a$	AT1	AT2
CM1	$S_{G1} - C_{CM1}$, $M^{T2}_{AT1} - C_{AT1}$	$-S_{G2} - C_{CM1}$, $M^{T2}_{AT2} - C_{AT2}$
CM2	$-S_{G1} - C_{CM2}$, $-M^{T2}_{AT1} - C_{AT1}$	$S_{G2} - C_{CM2}$, $-M^{T2}_{AT2} - C_{AT2}$

In this game, the adversary gains/loses a value of M^{T2}_{AT2} if AT2 is successful/unsuccessful. Meanwhile the adversary may choose to issue AT1 in order to misguide the defense system. In this case, the gain/loss of applying this action is represented by M^{T2}_{AT1}. In the modeled game, the gain/loss of AT2 is higher than in the AT1 attack since the intention of the adversary is to break the G_2 security goal. Hence, we have $M^{T2}_{AT1} < M^{T2}_{AT2}$. As in the previous scenario, to motivate the rational adversary, we assume that $M^{T2}_{AT1}, M^{T2}_{AT2} > C_{AT1}, C_{AT2}$.

4.1 Nash Equilibrium Analysis

Assuming that the defense system takes the pure strategy CM1, then the best response for the adversary is AT2 when $M^{T2}_{AT1} - C_{AT1} \leq M^{T2}_{AT2} - C_{AT2}$. However, if the adversary plays AT2, then CM1 will not be the best response for the

Table 5: Payoffs and Numerical Examples of Type 2 Adversary

Table 5.a: Payoffs of Type 2 Adversary in G_1 Preferred SPS

$d \setminus a$	AT1	AT2
CM1	U^{++}, U^{+}	U^{-}, U^{++}
CM2	U^{--}, U^{-}	U^{+}, U^{--}

Table 5.b: Numerical Example of Type 2 Adversary in G_1 Preferred SPS

$d \setminus a$	Heavy Load	High Sensitivity
Issue Puzzle	$85, 50$	$-15, 85$
Drop Request	$-95, -90$	$5, -100$

Table 5.c: Payoffs of Type 2 Adversary in G_2 Preferred SPS

$d \setminus a$	AT1	AT2
CM1	U^{+}, U^{+}	U^{--}, U^{++}
CM2	U^{-}, U^{-}	U^{++}, U^{--}

Table 5.d: Numerical Example of Type 2 Adversary in G_2 Preferred SPS

$d \setminus a$	Heavy Load	High Sensitivity
Issue Puzzle	$5, 50$	$-95, 85$
Drop Request	$-15, -90$	$85, -100$

defense system, which will play CM2 instead. Hence, the adversary is motivated to trigger AT1 when $-M^{T2}_{AT1} - C_{AT1} \geq -M^{T2}_{AT2} - C_{AT2}$. Accordingly, pure strategy Nash equilibrium exists when: (i) $M^{T2}_{AT1} - C_{AT1} > M^{T2}_{AT2} - C_{AT2}$, or (ii) $-M^{T2}_{AT1} - C_{AT1} < -M^{T2}_{AT2} - C_{AT2}$.

4.2 Mixed Strategy Equilibrium Analysis

We check for Nash equilibrium when the adversary plays mix strategy AT1 with probability q_2 and AT2 with probability $1-q_2$. The defense system's expected payoffs ($E_d(CM1)$ and $E_d(CM2)$) are the same as the expected payoffs defined in Equations (1) and (2). The adversary's equilibrium strategy is to play AT1 with $q_2 = \frac{S_{G2}-C_{CM2}+S_{G2}+C_{CM1}}{2S_{G1}+2S_{G2}}$, similar to the case of the type 1 adversary in the previous section. This similarity suggests that despite the type of adversary, the probability of an attack is highly dependent on the security preference of the targeted system as well as the cost of applying a countermeasure. The expected payoffs of the adversary are

$$E_a(AT1) = p(M^{T2}_{AT1} - C_{AT1}) + (1-p)(-M^{T2}_{AT1} - C_{AT1}), \quad (5)$$

$$E_a(AT2) = p(M^{T2}_{AT2} - C_{AT2}) + (1-p)(-M^{T2}_{AT2} - C_{AT2}). \quad (6)$$

The defense system's equilibrium strategy is to make AT1 and AT2 indifferent to the adversary ($E_a(AT1) = E_a(AT2)$). Therefore, the defense system must choose CM1 with probability $p = \frac{M^{T2}_{AT1}+C_{AT1}-M^{T2}_{AT2}-C_{AT2}}{2M^{T2}_{AT1}-2M^{T2}_{AT2}}$. As we have pointed out in Section 3.2, applying a countermeasure using probability, has the risk of applying the wrong countermeasure. However, considering the costs and benefits of actions in our analysis aims at decreasing such risk.

4.3 Case Based Analysis

In this section, we introduce two cases for the proposed game theoretic model and analyze the potential responses of the two players.

- **Case 1: Type 2 Adversary vs. SPS System with G_1 Preference**

  Assuming that the defender benefits much more by keeping the G_1 security goal than the G_2 security goal, we have: $S_{G1} >> S_{G2}$. Given this assumption, the payoffs for both the defense system and the adversary are as shown in Table 5.a. Similar to Section 3.3, U represents the outcome related to Table 4 while shows the the magnitude of the gain/lost in the payoff value.

  Table 5.b exemplifies an insider attack on an SPS system with a high preference for G_1 security goal. The equilibrium solution of the game is $p = 0.22$, $1 - p = 0.78$, $q_2 = 0.10$, and $1 - q_2 = 0.90$. The adversary will rationally choose to initiate AT2 with the probability of 0.10. Therefore, the defense system's best response to the adversary's mix strategy is to select CM2 with the probability of 0.78.

- **Case 2: Type 2 Adversary vs. SPS System with G_2 Preference**

  Assume that the defender benefits much more from protecting security goal G_2 than from protecting security goal G_1. Hence, we have: $S_{G2} >> S_{G1}$. The payoffs are shown in Table 5.c. An example of payoffs for both players are shown in Table 5.d. In this case, we get a mix strategy equilibrium for the game when $p = 0.22$, $1 - p = 0.78$, $q_2 = 0.90$, and $1 - q_2 = 0.10$. Therefore, the rational action for the defense system is to choose CM1 only if it believes $q_2 > 0.90$. Otherwise, CM2 is the best response.

In the above two cases with different goal preferences, the probability of response action CM2 is identical in both cases, because the intention of the adversary in both cases is the same. In case 2, the probability of AT2 is lower than in case 1 due to its higher risk of benefiting the defense system.

In the two game models discussed in Sections 3 and 4, the defense system is able to make an intelligent decision that considers both (i) system security goals, and (ii) adversary intentions in targeting the SPS system. The type of adversary (its costs and benefits) is considered to be known when deciding on the response action. However, this is not always the case. Next, we discuss a game theoretic model that incorporates such uncertainty into the response action selection.

5. ADVERSARY-TYPE UNCERTAINTY

In this section, we consider scenarios in which the defender is uncertain about the intention of the adversary in targeting

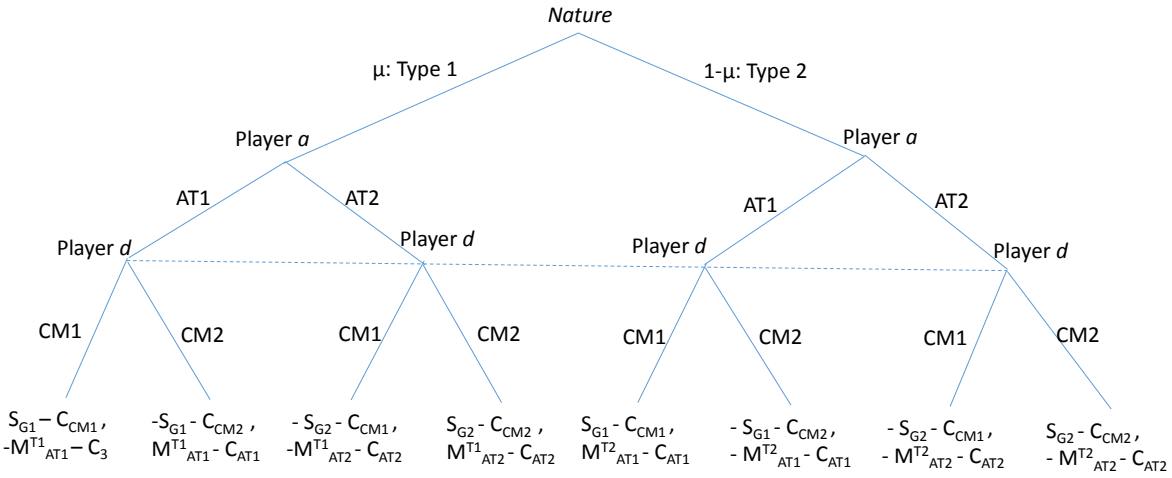

Figure 1: Extensive Form of the Modeled Bayesian Game

the SPS system. This situation could be interpreted as that of responding to two types of adversaries. We consider a two-player static Bayesian game.

Figure 1 illustrates the extensive form of the Bayesian game. Here, node *Nature* determines the type of adversary. The objective of both players is to maximize their expected payoffs. This implies that we assume both players to be rational. This assumption is a generic assumption for a well-defined adversary-defender game [16]. In the defined game, the adversary plays a Bayesian strategy in order to minimize his chances of being detected, and the defender plays a Bayesian strategy in order to maximize his chance of responding to attacks without introducing high cost to the SPS system.

5.1 Bayesian Nash Equilibrium (BNE) Analysis

We analyze BNE based on the assumption that the type of adversary is unknown to the defense system. We assume that μ is a common prior, meaning that the adversary a knows the defender's belief of μ. Obviously, the adversary has private information behind its intention to target G_1 or G_2. In the following, we analyze the four possible pure-strategy BNEs that could exist.

1. If the adversary plays the pure strategy pair (**AT1 if type 1, AT1 if type 2**), then the expected payoff of the defense system playing the pure strategy CM1 is

$$E_d(CM1) = \mu(S_{G1} - C_{CM1}) + (1-\mu)(S_{G1} - C_{CM1}). \quad (7)$$

The expected payoff of the defense system playing pure strategy CM2 is

$$E_d(CM2) = \mu(-S_{G1} - C_{CM2}) + (1-\mu)(-S_{G1} - C_{CM2}). \quad (8)$$

So if $E_d(CM1) > E_d(CM2)$ or if $S_{G1} > \frac{C_{CM1} - C_{CM2}}{2}$ (which is always true since the assumption is $S_{G1} > C_{CM1}, C_{CM2}$), then the best response of the defense system is to play CM1. However, if the defense system plays CM1, then AT1 is not the best response when $-M^{T1}_{AT1} - C_{AT1} \leq -M^{T1}_{AT2} - C_{AT2}$ or $M^{T2}_{AT1} - C_{AT1} \leq M^{T2}_{AT2} - C_{AT2}$, and the adversary will move on to play AT2 instead. Therefore, in this case, (AT1 if type 1, AT1 if type 2, CM1) is not a BNE. However, if $-M^{T1}_{AT1} - C_{AT1} > -M^{T1}_{AT2} - C_{AT2}$ and $M^{T2}_{AT1} - C_{AT1} > M^{T2}_{AT2} - C_{AT2}$, the best response for the defender is CM1, and thus (AT1 if type 1, AT1 if type 2, CM1) is a pure-strategy BNE.

2. If the adversary plays the pure strategy pair (**AT1 if type 1, AT2 if type 2**), then the expected payoff for the defense system playing the pure strategy CM1 is

$$E_d(CM1) = \mu(S_{G1} - C_{CM1}) + (1-\mu)(-S_{G2} - C_{CM1}). \quad (9)$$

Similarly, the expected payoff of the defense system playing the pure strategy CM2 is

$$E_d(CM2) = \mu(-S_{G1} - C_{CM2}) + (1-\mu)(S_{G2} - C_{CM2}). \quad (10)$$

So if $E_d(CM1) > E_d(CM2)$ or if $\mu > \frac{2S_{G2} - C_{CM2} + C_{CM1}}{2S_{G1} + 2S_{G2}}$, then the best response of the defense system is to play CM1. However, if the defense system plays CM1, then AT1 is not the best response for targeting G_1 when $-M^{T1}_{AT1} - C_{AT1} \leq -M^{T1}_{AT2} - C_{AT2}$, and it switches to AT2. If $M^{T2}_{AT1} - C_{AT1} \leq M^{T2}_{AT2} - C_{AT2}$, then AT2 is not the best response for targeting G_2, and the adversary moves on to play AT1. Hence, (AT1 if type 1, AT2 if type 2, CM1, μ) is not a pure-strategy BNE.

If $\mu < \frac{2S_{G2} - C_{CM2} + C_{CM1}}{2S_{G1} + 2S_{G2}}$, then the best response of the defense system is to play CM2. However, if $M^{T1}_{AT2} - C_{AT2} \geq M^{T1}_{AT1} - C_{AT1}$ then a type 1 adversary switches to AT2, and if $-M^{T1}_{AT1} - C_{AT1} \geq -M^{T2}_{AT2} - C_{AT2}$, then the type 2 adversary switches to AT1. Hence, (AT1 if type 1, AT2 if type 2, CM2, μ) is not a pure-strategy BNE.

3. If the adversary plays the pure strategy pair (**AT2 if type 1, AT1 if type 2**), then the expected payoff of the defense system playing the pure strategy CM1 is

Table 6: Case Based Analysis of Adversary-Type Uncertainty

	Type 1 AT1 (q_1)	**Type 1 AT2** ($1-q_1$)	**Type 2 AT1** (q_2)	**Type 2 AT2** ($1-q_2$)	**CM1** (p)	**CM2** ($1-p$)
Case 1	0.20	0.80	0.00	1.00	0.78	0.22
Case 2	1.00	0.00	0.80	0.20	0.22	0.78

$$E_d(CM1) = \mu(-S_{G2} - C_{CM1}) + (1-\mu)(S_{G1} - C_{CM1}). \quad (11)$$

The expected payoff of the defense system playing pure strategy CM2 is

$$E_d(CM2) = \mu(S_{G2} - C_{CM2}) + (1-\mu)(-S_{G1} - C_{CM2}). \quad (12)$$

So if $E_d(CM1) > E_d(CM2)$, or if $\mu < \frac{2S_{G1}+C_{CM2}-C_{CM1}}{2S_{G1}+2S_{G2}}$, then the best response by the defense system is to play CM1. As with the previous case, we can show that (AT2 if type 1, AT1 if type 2, CM1, μ) and (AT2 if type 1, AT1 if type 2, CM2, μ) are not pure-strategy BNEs.

4. If an adversary plays the pure strategy pair (**AT2 if type 1, AT2 if type 2**), then the expected payoff of the defense system playing the pure strategy CM1 is

$$E_d(CM1) = \mu(-S_{G2} - C_{CM1}) + (1-\mu)(-S_{G2} - C_{CM1}). \quad (13)$$

The expected payoff of the defense system playing pure strategy CM2 is

$$E_d(CM2) = \mu(S_{G2} - C_{CM2}) + (1-\mu)(S_{G2} - C_{CM2}). \quad (14)$$

So if $E_d(CM2) > E_d(CM1)$, or if $S_{G2} > \frac{C_{CM2}-C_{CM1}}{2}$ (which is always true since the assumption is that $S_{G2} > C_{CM1}, C_{CM2}$), then the best response of the defense system is to play CM2. However, if the defense system plays CM2, then AT2 is not the best attack strategy when $M^{T1}_{AT2} - C_{AT2} \leq M^{T1}_{AT1} - C_{AT1}$ and $-M^{T2}_{AT2} - C_{AT2} \leq -M^{T2}_{AT1} - C_{AT1}$. Hence, the adversary moves on to play AT1 instead. Therefore, (AT2 if type 1, AT2 if type 2, CM2) is not a BNE. However, if $M^{T1}_{AT2} - C_{AT2} > M^{T1}_{AT1} - C_{AT1}$ and $-M^{T2}_{AT2} - C_{AT2} > -M^{T2}_{AT1} - C_{AT1}$, then the best response for the defender is CM2, and thus (AT2 if type 1, AT2 if type 2, CM2) is a pure-strategy BNE.

We previously showed that pure-strategy BNE exists when (i) AT1 if type 1, AT1 if type 2, CM1, $-M^{T1}_{AT1} - C_{AT1} > -M^{T1}_{AT2} - C_{AT2}$ and $M^{T2}_{AT1} - C_{AT1} > M^{T2}_{AT2} - C_{AT2}$, and (ii) AT2 if type 1, AT2 if type 2, CM2, $M^{T1}_{AT2} - C_{AT2} > M^{T1}_{AT1} - C_{AT1}$ and $-M^{T2}_{AT2} - C_{AT2} > -M^{T2}_{AT1} - C_{AT1}$. Although there exist a number of BNEs for particular pure strategies that meet certain criteria, here, we seek to find a mixed-strategy BNE for cases that do not result in a pure-strategy BNE. For this we must introduce two new belief probabilities. Setting the expected value of playing strategies CM1 and CM2 equal to each other we get

$$\begin{aligned} &\mu q_1(S_{G1} - C_{CM1}) + \mu(1-q_1)(-S_{G2} - C_{CM1}) + \\ &(1-\mu)q_2(S_{G1} - C_{CM1}) + (1-\mu)(1-q_2)(-S_{G2} - C_{CM1}) = \\ &\mu q_1(-S_{G1} - C_{CM2}) + \mu(1-q_1)(S_{G2} - C_{CM2}) + \\ &(1-\mu)q_2(-S_{G1} - C_{CM2}) + (1-\mu)(1-q_2)(S_{G2} - C_{CM2}). \end{aligned} \quad (15)$$

Thus, the strategy pair (q_1 if type 1 adversary, q_2 if type 2 adversary, μ) is a mixed-strategy BNE, if Equation 15 is satisfied. Consequently, neither of the players can improve their payoffs by changing strategies.

5.2 Case Based Analysis

Let us describe how the proposed Bayesian game model analyzes potential responses given uncertainty regarding adversarial intentions. Here, we consider a scenario where the probability of targeting G_1 and G_2 (a type 1 adversary and a type 2 adversary) are the same in the SPS system.

We analyzed the response of the SPS system in two cases in which the preference of the SPS system is to protect either: (i) the G_1 security goal or (ii) the G_2 security goal. These two cases are exemplified using the two sets of numerical inputs introduced in Sections 3 and 4. Consequently, cases 1 and 2 are evaluated. We solved these two games using GAMBIT software [2]. Table 6 shows the probability of each strategy for both players a and d. In the rest of this subsection, we discuss in detail the response of the SPS system in each case.

- **Case 1: Uncertain Adversary Type vs. SPS System with G_1 Preference**

  We exemplify a type 1 adversary in an SPS system where G_1 is preferred over G_2. The numerical example of this case is shown in Figure 2. This game does not admit any NE solution in pure strategies. However, a unique NE is numerically computed in mixed strategies and shown in Table 6. At the NE, the type 1 adversary issues AT1 with the probability of 0.20. A reason for this low probability is the discouraging effect of the SPS system's capability to correctly respond to the attack. The NE strategy of the defender is CM1, with the probability of 0.78, and CM2 with the probability of 0.22.

- **Case 2: Uncertain Adversary Type vs. SPS System with G_2 Preference**

  Here, we consider a scenario where the preference of the SPS system is to protect the G_2 security goal in the SPS system. The inputs are the same as in case 2 of the game model defined in Sections 3 and 4. At the NE, the probability of CM1 is 0.22, while the probability of CM2 is 0.78. The outcome of the game is based on consideration of G_2 preference in the SPS system.

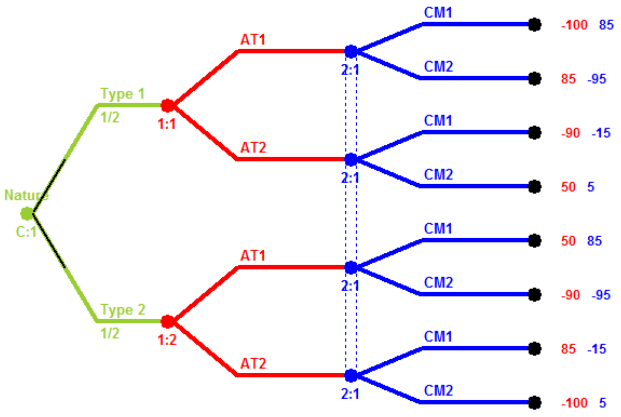

Figure 2: An Example of a Bayesian Security Game and Its Numerical Solution Obtained Using GAMBIT Software [2]

In this section, we used Bayesian game technique to model and analyze the uncertainty about the type of adversary targeting the SPS system. In addition, we provided a detailed analysis for various possible security goal preferences. In the next section, we describe the case study employing the proposed game-theoretic approach in selecting a proper countermeasure.

6. PERFORMANCE EVALUATION

As a proof of concept for the proposed approach, we performed a case study with adversary-type uncertainty. The experiments are executed on a Voice over IP (VoIP) telephony system. The motivation for choosing such a system is the increasing number of application-layer flooding attacks in these systems. We study the case of application-layer attacks in which adversaries trigger various types of flooding attacks in an attempt to avoid detection. In this experimental evaluation, the following research questions are particularly taken into account:

• **RQ1 - Impact of adversary-type uncertainty**: What is the impact of considering uncertainty about adversary's type in the performance of the action selection in the SPS system?

• **RQ2 - Impact of security goal preference**: What is the impact of considering the preferences of various security goals in the performance of the action selection in the SPS system?

To answer the above questions, we conducted two types of flooding attacks while the SPS system is uncertain about the type of adversary. We analyzed and compared the impact of various mitigation approaches, including our proposed Bayesian game model.

Session Initiation Protocol (SIP) is a core protocol for real-time communication networks. VoIP communications rely on SIP protocol. Message flooding attacks exploit the common vulnerability of servers: their limited processing capability. Flooding attacks can be achieved with different SIP messages (REGISTER, INVITE, OPTIONS, etc.) [7]. In our experiments, we consider two types of flooding attacks: (i) REGISTER, and (ii) INVITE.

Protection of SIP servers against flooding attacks requires an online detection and mitigation technique to recognize such malicious requests and to drop them. Most research on the detection and mitigation of SIP flood attacks focuses on mining network-layer data [7]. However, this case study aims at incorporating the adversary-type uncertainty and the preferences of the SPS system when providing a response action.

6.1 Implementations

OpenJSIP [1] is an open source SIP service implemented in Java. The current version of the project, which we use in our experiments, is v0.0.4. OpenJSIP provides three services: *Proxy*, *Registrar*, and *Location Service*. The project is based on JainSIP [3], which is the standardized Java interface to the SIP for desktop and server applications.

To make a phone call, a user first needs to send a *registration request*, which will be directed to the Registrar server. If the request is handled successfully, then the user can make a *call request* to the Proxy server. In attempting to evolve OpenJSIP to an SPS system, our proposed decision-making approach is developed and integrated into OpenJSIP. The implemented adaptation manager monitors the requests and provides the necessary adaptation action upon receiving malicious requests. In the designed game model, the payoff values of actions are populated based on stakeholders' preferences. The payoff tables provide enough information for the decision-making engine to select proper actions.

Client traffic is generated using SIPp tool [4]. Three traffic generators are exploited, all running on the same machine with 3.00 GHz CPU and 4.00 GB memory. Two of the traffic generators represent legitimate users that try to call each other by sending REGISTER and INVITE requests to the Registrar and Proxy servers. The third traffic generator represents adversaries that send malicious traffic while the system is uncertain about the type of adversary. Two possible attacks studied in our experiments are (i) flooding the Registrar server with malicious REGISTER requests (AT1), and (ii) flooding the Proxy server with malicious INVITE requests (AT2).

6.2 Attack Scenario

The case study security goals are to sustain the availability of the *Registrar* and *Proxy* servers during an attack. Hence, the objective is to increase the number of successful registration and call requests that are initiated by legitimate users. Accordingly, legitimate users are successfully provided with their expected service. In the experiments, the security goals of the SPS system are (i) availability of the Registrar server (G_1), and (ii) availability of the Proxy server (G_2). Consequently, two type of adversaries threaten the SPS system: (i) a type 1 adversary which targets the availability of the Registrar server, and (ii) a type 2 adversary which targets the availability of the Proxy server. The SPS system can take two countermeasures: (i) dropping the incoming REGISTER requests (CM1), and (ii) dropping the incoming INVITE requests (CM2).

Two attack scenarios have been designed. In both, the adversary changes its strategy in order to confuse the SPS system by sending both malicious REGISTER and INVITE requests. In the first attack scenario, the intention of the adversary is to target the availability of the Registrar server (a type 1 adversary). In the second attack scenario, the intention of the adversary is to target the availability of the Proxy server (a type 2 adversary).

In both attack scenarios, two traffic generators represent

Table 7: Type 1 Adversary: Targeting the Availability of Registrar Server (G_1)

	No Adaptation	Protecting G_1	Protecting G_2	Known Adversary-Type	Adversary-Type Uncertainty Case 1	Adversary-Type Uncertainty Case 2
% Suc. Reg.	80	97	78	95	**95**	86
% Suc. Call	49	70	48	88	**83**	85

Table 8: Type 2 Adversary: Targeting the Availability of Proxy Server (G_2)

	No Adaptation	Protecting G_1	Protecting G_2	Known Adversary-Type	Adversary-Type Uncertainty Case 1	Adversary-Type Uncertainty Case 2
% Suc. Reg.	44	52	100	92	86	**99**
% Suc. Call	27	30	100	93	61	**95**

legitimate users try to call each other. First, both caller and callee send a REGISTER request to register their IDs at the Registrar server. Then, each caller tries to call a predefined callee by sending an INVITE request to the Proxy server. The server looks for the callee, and if the callee is available, it informs the caller to establish the call. The caller holds the line for 1 second and then tries to terminate the call by sending a BYE message to the server. Both sides terminate the call when they receive an ACK from the server. The clients keep generating traffic for a total of 1 minute.

Prior to starting the test, we ran a stress test to discover the saturation point of the server on the running computer. We tested the system with increasing traffic and monitored at which point the system failed to respond to client requests. The observed limit was around 40 Calls Per Second (CPS). This rate was used to flood the Registrar and Proxy servers with malicious requests while the adversary changed its strategy by changing the type of flooding attack. The attack starts after 5 seconds, when server and both clients start to run. The attack continues for the rest of the minute or until the system breaks down. The experimental setup satisfactorily illustrates the performance of the proposed approach when facing uncertainty about the type of adversary. The following subsection presents the outcome of the experiments.

6.3 Obtained Results

The experiments in this section evaluate the mitigation of attacks in the case of adversary-type uncertainty. Six approaches are considered for action selection by the defense system:

1. **No Adaptation**: selects no mitigation action during the attack.

2. **Protecting** G_1: selects countermeasures solely to protect G_1, which means dropping malicious REGISTER requests without considering the uncertainty about the strategy and intention of adversaries.

3. **Protecting** G_2: selects countermeasures solely to protect G_2, which means dropping malicious INVITE requests without considering the uncertainty about the strategy and intention of adversaries.

4. **Known Adversary-Type**: selects countermeasures assuming that the intention behind the attack is known to the system and it is considered in the action selection. (Game model in Section 3 and Section 4)

5. **Adversary-Type Uncertainty Case 1**: selects countermeasure using Bayesian game model while the SPS system has higher preference for G_1.

6. **Adversary-Type Uncertainty Case 2**: selects countermeasure using Bayesian game model while the SPS system has higher preference for G_2.

In the following, we aim at providing the relevant answers for each of the two research questions introduced in the beginning of this section. To make this possible, we need to measure the effectiveness of all six approaches. An appropriate method is to assess the percentage of successful registrations and successful calls for legitimate users during type 1 and type 2 adversary attacks.

Table 7 summarizes the result of the first attack scenario, whose intention was to target the availability of the Registrar server. In this case, *Protecting* G_1, *Known Adversary Type*, and *Adversary-Type Uncertainty Case 1* are among the successful approaches. However, *Protecting* G_1 performs best solely in terms of successful registrations. *Known Adversary Type* technique is based on the assumption that the SPS system is aware of the intention and strategy of adversaries, while in *Adversary-Type Uncertainty Case 1*, the SPS system is uncertain about the type of adversary. The results are comparable in both techniques in percentage of successful registrations (addressing *RQ1*). In terms of successful calls, *Adversary-Type Uncertainty Case 1* has less value compared to *Adversary-Type Uncertainty Case 2* because in the former the security goal G_1 had higher priority. Hence, the SPS system selected countermeasures that protected this goal (addressing *RQ2*).

Table 8 summarizes the results of the second attack scenario, whose intention was to target the availability of the Proxy server. As with the previous attack scenario, it appears that there is no significant difference between *Protecting* G_2, *Known Adversary Type*, and *Adversary-Type Uncertainty Case 2* techniques. The response to *RQ1*, based on the results, is that our proposed approach to address uncertainty obtains results comparable to no uncertainty in the type of adversary (*Known Adversary Type*). *Protecting* G_2 is the most effective approach as it aims at protecting the Proxy server, which is reflected during countermeasure se-

lection. However, this approach assumes that the intention of the adversary is known to the SPS system.

Comparison of *Adversary-Type Uncertainty Case 1* and *Adversary-Type Uncertainty Case 2* approaches, indicates that including the preferences of the security goals is more satisfactory when the adversary targets the security goal that is of higher preference. If the adversary targets the availability of the Registrar server (G_1), then the Case 1 uncertainty approach results in more-successful registrations and calls. If the adversary targets the availability of the Proxy server (G_2), then the Case 2 obtains more effective results than Case 1 (addressing *RQ2*).

Our case study illustrates that the proposed Bayesian game theoretic approach effectively addresses uncertainty about adversary type by selecting a countermeasure that maximizes the expected payoff of the SPS system considering the strategy of the adversary and the security goal preference of the SPS system.

7. RELATED WORK

The increasing number of dynamic application-layer attacks [8] calls for new approaches to adaptive application security. One important component in SPS systems is the decision-making process responsible for selecting response actions. Despite the central role of the decision-making process in SPS systems, uncertainty about the type of adversary has not been directly modeled for these systems. Consequently, it is not incorporated in driving the action-selection process.

Capturing intentions of attackers and the alternative ways they can be realized through an attack is studied in the field of requirement engineering. Alexander [5] advocates using misuse and use cases together to conduct threat analysis during requirements analysis. Van Lamsweerde [21] provides constructive guidance in early elaboration of security concerns. In his paper, a requirement engineering method for elaborating an building an intentional anti-model yielding vulnerabilities and capabilities required for achieving the anti-goals. Although security requirement engineering provides models that elicit attackers' goals and intentions at the requirement engineering stage, run-time action selection in an SPS system requires the support of run-time models.

Only a few cybersecurity approaches consider adversary-type uncertainty in their decision models. For instance, a recent work by Garnaev et al. [12] studied the incorporation of adversary-type uncertainty in network security protection. They discuss the dilemma a network defender faces due to limited resources: either to focus on increasing defense of the most valuable nodes or to defend all the nodes while reducing the level of defense of the most valuable ones. They suggested a Bayesian game in which the probability of the adversary's type can be considered as a sub-scale in the scale of threat levels. While the authors in [12] address the uncertainty in network security, their modeled matrix game is specific to a network of nodes in a communication network and cannot be applied in software security.

Another work, by Li and Wu [14], describes a dynamic Bayesian game between regular and malicious nodes in Mobile Ad hoc Networks (MANETs). In their modeled game, the regular node forms beliefs and measures uncertainty to evaluate the type of opponent. The game is studied to choose the probability of cooperating with the opponent or to report the malicious node. Additionally, a malicious node evaluates the risk of being caught and uses its flee strategy to avoid punishment. The game modeled by Li and Wu [14] is specific to MANETs and cannot be applied in the field of software security. In wireless ad hoc networks, the uncertainty of a defender about his type of opponent (regular or malicious) is formulated in both static and dynamic Bayesian game contexts [16]. The modeled game is specific to IDS implementation in a network layer and cannot be employed in the field of software security, and more specifically, for addressing security goals at the application level. Liu et al. [15] developed a game-theoretic formalization that can capture the inherent interdependency between an adversary's intent, objectives, and strategies and a defender's objectives and strategies in such a way that adversarial objectives and strategies can be automatically inferred. However, in their game model the security goal preferences of the system are not considered in the decision model.

Recent literature in security games incorporates payoff uncertainty [17] and adversarial uncertainty [18] using Stackelberg security games. Their modeled security game is general and not specific to cyber security. Fielder et. al. [11] address the challenge of making better security decisions by the aid of a game-theoretic model that captures essential characteristics of resource allocation decision making (i.e. system administrators' time) to prevent data loss and defend system and network assets of an organization.

As discussed above, the state-of-the-art either focuses on incorporating the uncertainty about adversary types in (i) specific network security scenarios or (ii) general security games. However, the focus of this article is on software security and fusing the uncertainty about adversary types into the decision-making model of an SPS system. More specifically, the benefits and costs of protecting/mitigating security goals in a software system are incorporated in the proposed decision model.

8. CONCLUSION

This article aims at providing a decision-making model that addresses adversary-type uncertainty. For this purpose, this work deals with how to keep security goals in the SPS system and to involve the intention and strategy of an adversary in the decision model. We used Bayesian game theoretic technique to analyze the battle between the adversary and defense system. The defense system forms beliefs and measures uncertainty to evaluate the type of opponent and selects a countermeasure according to the adversarial strategy. The adversary keeps evaluating the cost and benefit of initiating an attack and selects a strategy aiming to minimize punishment. We provide Bayesian Nash Equilibrium (BNE) analysis of the game, along with a case study. The experiments conducted on a Java-based VoIP telephony system, reveal that the proposed Bayesian game model efficiently selects proper countermeasures in the case of adversary-type uncertainty. The selected countermeasures satisfy the preferred security goals while considering the benefit and the cost of applying countermeasures. The focus in this article is selecting a single countermeasure. However, the problem can be generalized to plan a course of actions against a planned attack. Therefore, of interest for future research is an extension of this model to counter multiphase, well-planned, carefully hidden attacks.

9. REFERENCES

[1] Open Java SIP. [Online.] Available: https://code.google.com/p/openjsip/, 2009.

[2] Gambit Game Theory Analysis Software and Tools. [Online.] Available: http://gambit.sourceforge.net/, 2016.

[3] JAIN SIP: The Standardized Java Interface to the Session Initiation Protocol. [Online.] Available: https://jsip.java.net/, 2016.

[4] SIPp: Traffic Generator for the SIP Protocol. [Online.] Available: http://sipp.sourceforge.net/, 2016.

[5] I. Alexander. Misuse cases: Use cases with hostile intent. *IEEE Software*, 20(1):58–66, 2003.

[6] N. Bencomo, A. Belaggoun, and V. Issarny. Dynamic decision networks for decision-making in self-adaptive systems: a case study. In *Proceedings of International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 113–122, 2013.

[7] S. Ehlert, D. Geneiatakis, and T. Magedanz. Survey of network security systems to counter SIP-based denial-of-service attacks. *Computers & Security*, 29(2):225–243, 2010.

[8] A. Elkhodary and J. Whittle. A survey of approaches to adaptive application security. In *Proceedings of International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 20–26, 2007.

[9] M. Emami-Taba, M. Amoui, and L. Tahvildari. On the road to holistic decision making in adaptive security. *Technology Innovation Management Review*, 3(8):59–64, 2013.

[10] M. Emami-Taba, M. Amoui, and L. Tahvildari. Mitigating dynamic attacks using multi-agent game-theoretic techniques. In *Proceedings of IBM Center for Advanced Studies Conference*, pages 375–378, 2014.

[11] A. Fielder, E. Panaousis, P. Malacaria, C. Hankin, and F. Smeraldi. Game theory meets information security management. In *Proceedings of Information Security Conference*, pages 15–29, 2014.

[12] A. Garnaev, M. Baykal-Gursoy, and H. V. Poor. Incorporating attack-type uncertainty into network protection. *IEEE Transactions on Information Forensics and Security*, 9(8):1278–1287, 2014.

[13] S. Kandula, D. Katabi, M. Jacob, and A. Berger. Botz-4-sale: Surviving organized DDoS attacks that mimic flash crowds. In *Proceedings of Symposium on Networked Systems Design & Implementation*, pages 287–300, 2005.

[14] F. Li and J. Wu. Hit and run: a bayesian game between malicious and regular nodes in MANETs. In *Proceedings of Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 432–440, 2008.

[15] P. Liu, W. Zang, and M. Yu. Incentive-based modeling and inference of attacker intent, objectives, and strategies. *ACM Transactions on Information and System Security*, 8(1):78–118, 2005.

[16] Y. Liu, C. Comaniciu, and H. Man. A bayesian game approach for intrusion detection in wireless ad hoc networks. In *Proceedings of Workshop on GameTheory for Communications and Networks*, 2006.

[17] T. H. Nguyen, F. M. Delle Fave, D. Kar, A. S. Lakshminarayanan, A. Yadav, M. Tambe, N. Agmon, A. J. Plumptre, M. Driciru, F. Wanyama, and A. Rwetsiba. Making the most of our regrets: Regret-based solutions to handle payoff uncertainty and elicitation in green security games. In *Proceedings of Conference on Decision and Game Theory for Security*, pages 170–191, 2015.

[18] T. H. Nguyen, A. Sinha, and M. Tambe. Conquering adversary behavioral uncertainty in security games: An efficient modeling robust based algorithm. In *Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, pages 4242–4243, 2016.

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

[20] Y. Shoham and K. Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[21] A. Van Lamsweerde. Elaborating security requirements by construction of intentional anti-models. In *Proceedings of International Conference on Software Engineering*, pages 148–157, 2004.

[22] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: Using hard AI problems for security. In *Advances in Cryptology*, volume 2656 of *Lecture Notes in Computer Science*, pages 294–311. 2003.

[23] S. A. Zonouz, H. Khurana, W. H. Sanders, and T. M. Yardley. RRE: a game-theoretic intrusion response and recovery engine. *IEEE Transactions on Parallel and Distributed Systems*, 25(2):395–406, 2014.