A Game-Theoretic Approach for Security Control Selection[*]

Dylan Léveillé   Jason Jaskolka

Department of Systems and Computer Engineering
Carleton University, Ottawa, ON, Canada

dylan.leveille@carleton.ca   jason.jaskolka@carleton.ca

Selecting the combination of security controls that will most effectively protect a system's assets is a difficult task. If the wrong controls are selected, the system may be left vulnerable to cyber-attacks that can impact the confidentiality, integrity and availability of critical data and services. In practical settings, it is not possible to select and implement every control possible. Instead considerations, such as budget, effectiveness, and dependencies among various controls, must be considered to choose a combination of security controls that best achieve a set of system security objectives. In this paper, we propose a game-theoretic approach for selecting effective combinations of security controls based on expected attacker profiles and a set budget. The control selection problem is set up as a two-person zero-sum one-shot game. Valid control combinations for selection are generated using an algebraic formalism to account for dependencies among selected controls. We demonstrate the proposed approach on an illustrative financial system used in government departments under four different scenarios. The results illustrate how a security analyst can use the proposed approach to guide and support decision-making in the control selection activity when developing secure systems.

1 Introduction

With computers becoming more interconnected than ever, there emerges an even greater need to secure computer systems and to effectively manage security risks. Security risks are mitigated by the implementation of a set of security controls. A *security control* refers to a safeguard or countermeasure prescribed for an information system or an organization designed to protect the confidentiality, integrity, and availability of its information and to meet a set of defined security requirements [28].

Control selection is an activity commonly found as part of a risk management process [10], a systems engineering process [28], the Risk Management Framework [13], the Cybersecurity Framework [25], or the Privacy Framework [24]. Control selection involves selecting and documenting the security controls necessary to protect the information system and organization commensurate with risk to organizational and system operations and assets, individuals, other organizations, and the nation [13].

During the control selection activity, security analysts typically select security controls from standardized security control catalogues, such as NIST SP 800-53 [15], ITSG-33 [8], ISO 27002 [11], CIS Critical Security Controls [4], and MITRE D3FEND™ [16], among others. However, selecting combinations of controls from these catalogues can be difficult for several reasons. First, these control catalogues are large, and many possible controls could be selected to mitigate the risks identified for a given system. In practical settings, it is not possible to select and implement every control possible. Considerations such as budget, effectiveness, and dependencies among various controls, must be considered to choose a combination of security controls that best achieve a set of system security objectives. Second, control selection is largely a human-oriented activity. The dynamics between security analysts (defenders) strategizing to protect critical systems and assets and achieve a set of security objectives, and attackers aiming

---

to impact critical systems and assets and violate those same security objectives must be considered when deciding on the most effective and cost-efficient combination of security controls. Although numerous optimization-based solutions are adept at accounting for various properties of the controls themselves, they fail to capture the human element that is inherently part of the control selection activity.

To address the above mentioned challenges, we propose a game-theoretic approach for security control selection. The human aspects of the control selection problem, as well as the large space of possible control combinations, and their dependencies and constraints, lends itself well to an application of game theory. Specifically, we set up a two-person zero-sum one-shot game which is played by a security analyst. The analyst selects their strategy based on an attacker profile, characterized by the expected targeted assets and security objectives. Each analyst strategy corresponds to a combination of security controls from a chosen control catalogue that are capable of achieving the security objectives. Valid control combinations are generated using an algebraic formalism (akin to product family algebra [9]) to account for dependencies among selected controls. The outcome of the game is a combination of suggested security controls that can effectively defend against the considered attacker profile. Using an illustrative governmental finance system, we demonstrate the proposed approach under four different scenarios.

The rest of this paper is organized as follows. Section 2 provides an overview of existing works on the topic of control selection and of game theory applications in cybersecurity. Section 3 presents the proposed game-theoretic approach for control selection. Section 4 provides an illustrative example demonstrating the application of the proposed approach. Section 5 discusses the benefits and potential difficulties with the proposed approach. Lastly, Section 6 concludes and briefly discusses future work.

2   Related Work

Many existing approaches to support the security control selection activity are based on setting and solving optimization problems. For example, for each considered control, Yevseyeva et al. [36] assign a probability of "survival" for each possible threat (i.e., the probability that the threat persists in the presence of the control). Probabilities are also assigned for the expected loss of successful attacks. The goal of the proposed approach is to minimize this expected loss, under constraints such as cost and system resources. Similarly, Almeida and Respício [1] also assign probabilities to controls based on their expected performance in mitigating certain vulnerabilities. For the proposed approach, the goal is to find the optimal controls for the system that will minimize an objective function accounting for both loss and cost. A different approach was proposed by Dewri et al. [5] where systems are modelled as trees, in which the leaf nodes represent possible attacks. Controls therefore mitigate one or many leaf nodes. With the attack impact, attack frequency, and cost of each control known, the optimal controls can be found by optimization. A similar tree-like approach was also proposed by Park and Huh [27]. While optimization-based approaches can account for important considerations and constraints such as cost and effectiveness, they depend heavily on assumptions about probabilities for threat likelihoods, or control success rates. Such probabilities are not likely to be accurately known in a practical setting.

Several other approaches for control selection that are not based on optimization have also been proposed. Bettaieb et al. [2] presented an approach where a machine learning model is trained with historical data from previous security assessments to make predictions using certain features of interest from a given security assessment to determine optimal controls. However, using historic data to determine how to protect a system has several limitations as every system is unique and may operate in widely different environments. In another work, Kiesling et al. [18] proposed a simulation-based approach to determine the optimal controls for a system. To do this, expected attacks are simulated on different

components of the system using different possible control combinations to find the optimal ones. This approach is noteworthy as it simply uses the properties of the controls and of the current system (such as different threats) to find the most optimal control combinations and does not depend on any probabilities.

Several works have explored the use of game theory for addressing cybersecurity challenges. For example, Nassar et al. [23] proposed a technique which focused on evaluating a system's network security with the help of a game model. Smith et al. [32] used game theory to verify the security of hardware designs. Wang et al. [35] presented a network attack-defence game to help secure a computer network. However, game theory has yet to be utilized for security control selection.

In contrast to existing work, the proposed approach aims to leverage game theory to address the shortcomings of current control selection approaches by placing a central focus on possible attacker behaviours, while also taking into account the considerations and constraints that limit the selection of certain combinations of security controls to effectively mitigate the threats to a system.

3 The Proposed Approach

In this section, we present our proposed game-theoretic approach for security control selection. An overview of the approach is shown in Figure 1. The approach consists of two main stages shown as swim lanes and six steps shown in blue. All steps are to be conducted by a security analyst. A detailed description of each step of the proposed approach is provided in the sections below.
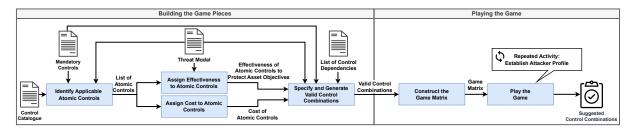

Figure 1: An overview of the proposed game-theoretic approach for security control selection

3.1 Identify Applicable Atomic Controls

The approach starts with the security analyst identifying applicable atomic controls from a given security control catalogue. In our context, the *atomic controls* are the smallest (indivisible) security controls that can be selected from a control catalogue. We say that a control is *applicable* to a system if it could provide any form of protection from the threats to the assets of the system.

We assume that a list of threats and assets are available to the security analyst in the form of a threat model. A threat model is defined as "a structured representation of all the information that affects the security of an application" [6]. Threat models typically include identified system threats and their impact on the assets within the system [6, 22]. The threat model can be obtained by applying a well-known threat modelling methodology such as STRIDE [21] or PASTA [34].

To determine the applicability of an atomic control, the security analyst must carefully consider each atomic control from the given control catalogue and decide if the control can mitigate the identified threats to the assets. Additionally, certain organizational needs or standards and regulations for the system's application domain may require that specific security controls be present in the system. The security analyst must therefore ensure that these *mandatory controls* are included as part of the set of

applicable controls identified. This is a manual process. However, it should be noted that the effort required for this activity is reasonable as security control catalogues are typically separated by control families which help guide an analyst in finding suitable controls [29]. Instructions and guidance for performing this task is well documented by ISO 27005 [12] and NIST SP 800-53B [14].

At the end of this step, the analyst will have a set of applicable atomic controls for the system. Note that the combination of suggested controls found by applying the proposed approach will be a subset of the controls gathered in this initial step.

3.2 Assign Effectiveness to Atomic Controls

For each identified atomic control, the analyst proceeds by assigning an effectiveness of the control at satisfying each *security objective* on each asset in the system. Security objectives represent the security needs of the assets on the system, such as confidentiality, integrity, and availability [3]. These objectives are normally included as part of the threat impacts described in the threat model. It is important to remember that the goal of the proposed approach is to create a game. In every game, there needs to be strategies, and payoffs defined for each strategy. Assigning the effectiveness of each atomic control therefore defines the payoffs of each atomic control in the game.

To perform this step of the approach, the atomic payoff matrix presented in Table 1 must be completed. The rows represent each atomic control that was identified in the previous step (denoted $C_1, \ldots, C_N$). The columns represent the security objectives for each asset (denoted $O_1, \ldots, O_M$). We expect the analyst to assign a value between 0 and 1 in each cell of this matrix. A value of 0 means that the atomic control is not effective at satisfying the specified objective for an asset, while a value of 1 means that the atomic control is completely effective at satisfying the specified objective for an asset. Each payoff value is therefore normalized. Provided that the rating scheme is selected and used consistently throughout the approach, the analyst is free to choose any method for assigning the effectiveness values for the atomic payoff matrix. For example, the analyst may choose to use a quantitative approach as in the Defect Detection and Prevention (DDP) risk reduction strategy developed by NASA [7], or they may alternatively choose to use a qualitative rating mapped to quantitative values as in [19].

Table 1: General form of the atomic payoff matrix

	Asset 1			Asset 2			...	Asset X		
	O_1	...	O_M	O_1	...	O_M	...	O_1	...	O_M
C_1										
$\vdots$										
C_N										

At the end of this step, the analyst will have the effectiveness of each applicable atomic control for satisfying each security objective on each asset in the system.

3.3 Assign Cost to Atomic Controls

In practical settings, cost or time constraints limit how many controls can be part of a system; if there are too many controls they may exceed a certain budget or cannot be implemented in reasonable time. In fact, without such constraints, there could technically be no limitations on the number of controls that can be selected for a system, and the best solution would be to select them all.

At the same time as assigning effectiveness, the analyst will also need to assign a cost for each identified atomic control. We expect the analyst to assign a cost from the set of real numbers $\mathbb{R}$. The units for cost could be represented as dollars, thousands of dollars, or any other form of currency as long as the same units are consistently used for all cost values. Furthermore, no units could be used if desired. Without units, costs simply represent an implementation effort.

After this step, the analyst will have the cost associated with each applicable atomic control.

3.4 Specify and Generate Valid Control Combinations

Given a set of applicable atomic controls, the analyst needs to specify and generate the set of valid control combinations that satisfies their constraints. To formally capture these constraints, we have decided to use an algebraic specification based on product family algebra [9] to specify and generate valid combinations of security controls.

Product family algebra extends the mathematical notions of semirings to describe and manipulate product families. A semiring is an algebraic structure $(S, +, \cdot, 0, 1)$ consisting of a set S with a commutative and associative binary operator $+$ and an associative binary operator $\cdot$. An element $0 \in S$ is the identity element with respect to $+$, while an element $1 \in S$ is the identity element with respect to $\cdot$. Additionally, $\cdot$ distributes over $+$ and element 0 annihilates S with respect to $\cdot$. A semiring is commutative if $\cdot$ is commutative and a semiring is idempotent if $+$ is idempotent.

For ease of presentation, we recast the vocabulary of product family engineering into the vocabulary of security controls by first defining a security control algebra to express families of security control combinations generated from a set of atomic controls.

Definition 1 (Security Control Algebra) *A security control algebra is a commutative idempotent semiring $\mathscr{C} \stackrel{\text{def}}{=} (C, \oplus, \odot, 0, 1)$ where each element of the semiring $c \in C$ is a security control family.*

In a security control algebra, the operator $\oplus$ is interpreted as a choice between two security control families and the operator $\odot$ is interpreted as a mandatory composition of two security control families[1]. The element 0 represents a non-implementable security control combination that cannot exist and the element 1 represents the empty security control combination which has no controls. A security control family is called a *security control combination* if it is indivisible with regard to the choice operator $\oplus$. Additionally, it is called a *proper security control combination* if $c \neq 0$. A security control combination is an *atomic control* if is it is indivisible with regard to the mandatory composition operator $\odot$. Optional controls are expressed as a choice between the controls and the empty security control combination 1. A list of optional controls $c_1, \ldots, c_n$ is denoted by $opt[c_1, \ldots, c_n] \stackrel{\text{def}}{=} (c_1 \oplus 1) \odot \cdots \odot (c_n \oplus 1)$.

For two security control families c_1 and c_2 in a security control algebra, the *refinement relation* ($\sqsubseteq$) is defined as $c_1 \sqsubseteq c_2 \stackrel{\text{def}}{\iff} \exists (c_3 \mid: c_1 \leq c_2 \odot c_3)$ where $\leq$ is the natural semiring order (i.e., $c_1 \leq c_2 \stackrel{\text{def}}{\iff} c_1 \oplus c_2 = c_2$). To specify constraints, such as dependencies between controls, we use the requirement relation.

Definition 2 (Requirement Relation [9]) *For elements c_1, c_2, c_3, c_4 and security control combination x in a security control algebra, the requirement relation ($\rightarrow$) is defined inductively as:*

$$c_1 \xrightarrow{x} c_2 \quad \stackrel{\text{def}}{\iff} \quad x \sqsubseteq c_1 \implies x \sqsubseteq c_2$$
$$c_1 \xrightarrow{c_3 \oplus c_4} c_2 \quad \stackrel{\text{def}}{\iff} \quad c_1 \xrightarrow{c_3} c_2 \;\wedge\; c_1 \xrightarrow{c_4} c_2$$

---

[1] When the context is clear, we omit the mandatory composition operator $\odot$ when specifying security control algebra terms.

For elements c_1, c_2 and x, the requirement relation $c_1 \xrightarrow{x} c_2$ can be read as "c_1 requires c_2 within x."

With this setting, all security control combinations can be specified algebraically by expressing the mandatory and optional controls as terms of a security control algebra along with requirement relations describing control dependencies.

The resulting specification serves as the basis for generating all possible proper security control combinations. However, not all control combinations are possible as some may exceed our defined budget. To make this determination we first define how to calculate the cost of a proper security control combination. In what follows, let $P \subseteq C$ be the set of all proper security control combinations in a security control algebra $\mathscr{C}$.

Definition 3 (Cost of a Proper Security Control Combination) *The cost of a proper security control combination Cost : $P \to \mathbb{R}$ is a function defined inductively for any proper security control combinations $a, b \in P$ in a security control algebra $\mathscr{C}$ as:*

$$
\begin{aligned}
Cost(1) &= 0 \\
Cost(a) &= G(a) \text{ if } a \text{ is atomic} \\
Cost(a \odot b) &= Cost(a) + Cost(b)
\end{aligned}
$$

where G is a function that returns the cost assigned to an atomic control (see Section 3.3).

Now that we can compute the cost of a proper security control combination, we determine the set of valid security control combinations. A *valid security control combination* is a proper security control combination that does not exceed the prescribed cost budget. The validity of a control combination is formalized in the following rule.

Definition 4 (Budget Rule) *For any $p \in P$ and budget B:*

$$
Valid(p) \iff Cost(p) \leq B
$$

After this step, the analyst will have a set of valid security control combinations that satisfy the prescribed budget. These valid security control combinations become the strategies that an analyst can select when playing the game.

3.5   Construct the Game Matrix

In this step, the analyst constructs the game matrix. The general form of the game matrix can be seen in Table 2. The rows represent the valid security control combinations found from the last step (denoted $Combo_1, \ldots, Combo_N$). The columns represent the security objectives for each asset (denoted $O_1, \ldots, O_M$). Note that the game matrix is identical in style to that of the atomic payoff matrix (see Table 1). The game matrix simply has control combinations as rows rather than atomic controls. In the game, the strategies of the security analyst will be the valid security control combinations, while the strategies of the attacker will be each security objective that could be violated on every asset.

Each outcome in a game is tied to a payoff [33]. In our game, the payoffs are represented from the perspective of the analyst and represent the effectiveness of the security control combinations towards every asset's security objectives. Just as cost was defined inductively, we can define a proper control combination's effectiveness towards an asset's security objective in a similar manner.

Table 2: General form of the game matrix

	Asset 1			Asset 2			...	Asset X		
	O_1	...	O_M	O_1	...	O_M	...	O_1	...	O_M
$Combo_1$										
$\vdots$										
$Combo_N$										

Definition 5 (Effectiveness of a Proper Security Control Combination) *The effectiveness of a proper security control combination towards an asset's security objective Eff : P → ℝ is a function defined inductively for any proper security control combinations $a, b \in P$ in a security control algebra $\mathscr{C}$ as:*

$$
\begin{aligned}
\mathit{Eff}(1) &= 0 \\
\mathit{Eff}(a) &= E(a) \text{ if a is atomic} \\
\mathit{Eff}(a \odot b) &= 1 - (1 - \mathit{Eff}(a))(1 - \mathit{Eff}(b))
\end{aligned}
$$

where E is a function that returns the effectiveness assigned to an atomic control for an asset's security objective (see Section 3.2).

With Definition 5, the payoff values in the game matrix can be calculated. Note that the calculation of the effectiveness of a security control combination is inspired from the combined effectiveness calculation as part of NASA's DDP approach [7].

After this step, the analyst will have the game matrix so that they can proceed to play the game.

3.6 Play the Game

The game is a *two-person zero-sum one-shot game*. The game is played by *two persons*: the security analyst and the attacker. The attacker may embody one or multiple entities, but acts as a unified adversary. The goal of the security analyst is to select the security control combination that will best protect the security objectives for the assets they believe will be targeted by the attacker. Only one security control combination can be selected, hence it is a *one-shot* game. On the other hand, the goal of the attacker is to attack assets and violate corresponding security objectives. An attacker could attack one or many assets and violate one or more objectives from a series of attacks. Regardless, an attacker will select which assets and objectives they will target and will commit to attacking the selected assets and objectives. The attacker will naturally prefer attacking assets which are not properly defended, i.e., those for which there are minimally effective security controls. The effectiveness values in the game matrix (payoffs) do not directly correlate to a loss to the attacker. However, it is easy to see that the higher the values, the more difficult it is for an attacker to conduct a successful attack leading to corresponding security objective violations. Therefore, what the security analyst gains in effectiveness is what the attacker loses in their ability to successfully conduct their attack; hence, it is a *zero-sum* game. Note that this game is strictly non-cooperative; the analyst and attacker are competing directly and would never want to cooperate.

Using the game matrix, the analyst must select a strategy (i.e., a valid security control combination) to play that will best protect the system assets and security objectives that they believe are most important. To do this, an analyst must establish the expected attacker profile. An *attacker profile* is an expected set of the assets and corresponding security objectives targeted by the attacker. One can imagine different classes of attackers having different capabilities, and different targets, thereby establishing different attacker profiles. In the context of a game, an attacker profile corresponds to guessing the attacker strategy

so that it can be defended. This consideration of the dynamics of the analyst and the attacker strategies in this game is what differentiates it from existing security control selection approaches.

It is impossible to know exactly which security objectives on which assets will be attacked, so assumptions must be made. One way to do this is to determine where most of the critical information flows in the system and which assets may be prone to more attacks (i.e., have more expected threats). The combination of these ideas can help localize assets that are more attractive for attacks, and therefore puts the security objectives of these assets at higher risk of violation. Another way to do this is to consider the risk to each asset and corresponding security objectives for the identified threats to the system (which we consider known to the analyst). In this case, prioritizing defence of assets and security objectives targeted by high risk threats may be a good approach. Regardless, once the attacker profile is determined, then the suggested strategy (i.e., the most effective security control combination) can be found.

Regardless of the approach taken to establish the attacker profile, it will articulate the objectives that are expected to be violated by an attacker. For this work, we establish an attacker profile by considering and prioritizing different attacker objectives. Attacker objectives correspond to a set of security objectives for some assets that are equally expected to be targeted by an attacker. Within an attacker profile, several attacker objectives may be prioritized according to their perceived likelihood of being targeted by the attacker to obtain a priority order for the objectives. For example, the security analyst could establish an attacker profile in which the attacker has two ordered attacker objectives: (1) to target the confidentiality of two specific assets equally, and (2) to target the integrity of two other assets equally. The security analyst may consider as many attacker objectives as they desire when developing an attacker profile. The suggested analyst strategies for an attacker profile will be those which maximize the *total effectiveness* across each attacker objectives (i.e., the sum of the effectiveness returned by Definition 5 for the security objectives in the attacker objectives is maximized in the priority order). To better understand this concept, an example of the strategies found by playing the game with an attacker profile with two ordered attacker objectives is visualized in Figure 2.
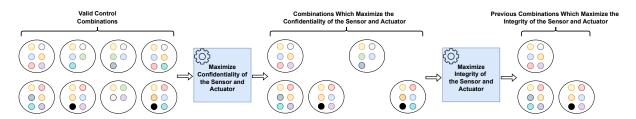

Figure 2: Finding the suggested controls for an attacker profile with multiple ordered attacker objectives

In this example, there are initially eight valid security control combinations. Each security control combination has a unique set of controls (denoted by the different coloured dots in the figure). Only two assets exist in this system; a Sensor and an Actuator. The attacker profile has two ordered attacker objectives: (1) the confidentiality of the Sensor and the Actuator and then (2) the integrity of the Sensor and the Actuator. From all valid control combinations, the control combinations which maximize the first set of attacker objectives is found, yielding five different combinations. From these five combinations, the combinations which maximize the second set of attacker objectives is found, yielding three control combinations. As there are no more ordered attacker objectives, the resulting control combinations are all considered equally valid, and represent the suggested strategies. Note that since the suggested strategies are derived through a series of maximization problems, it may be possible for more than one strategy to be the most effective for a given attacker profile.

At the end of this step, the analyst will obtain at least one strategy that best protects against the considered attacker profile and that corresponds to the suggested security control combinations to be implemented in the system. It is important to remember that this approach is a game. Therefore, as with any game, it is recommended that the game be re-constructed with different maximum budget values and re-played with different attacker profiles (as illustrated in Figure 1). This can help gauge and compare the control combinations that should be used for the system under different constraints and goals.

4 Illustrative Example

In this section, we demonstrate how the approach presented in Section 3 could be applied to support the control selection activity for an illustrative example system. Suppose a security analyst needs to select a combination of cost-effective security controls to protect a financial system used by the Canadian government called *Firebird*. An overview of the system architecture is shown in Figure 3. *Firebird* allows financial analysts to enter data about financial transactions and view those transactions through a user interface. Many identical interfaces may exist. The interfaces communicate over a 5G channel to a central processing system to process the commands from the analyst. A database stores the financial transactions data used by the central processing system. Both the processing system and database are located in an internal government network. The security analyst will apply the proposed game-theoretic approach for security control selection for the *Firebird* system as described in the following sections.
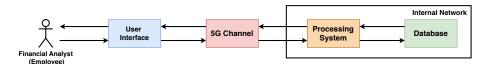

Figure 3: An overview of the *Firebird* system architecture

4.1 Identify Applicable Atomic Controls

Considering the *Firebird* system architecture shown in Figure 3, there are four primary assets: the user interface, the 5G channel, the processing system, and the database. For simplicity and brevity, suppose that the analyst is primarily focused on addressing threats to the user interface and the database and threats to the 5G channel and processing system are being handled by another analyst. Also, it has been pre-determined by the security analyst's government department that they are primarily concerned with the confidentiality (C), integrity (I), and availability (A) security objectives for the system assets.

Because *Firebird* is a Canadian government system, the analyst selects controls from the ITSG-33 control catalogue[2]. To comply with departmental requirements, it was decided by the security analyst's government department that the input validation control (i.e., *SI-10* in ITSG-33) must be present in the system. This is because improper input validation in any system can result in potentially severe consequences [17, 30, 31].

The analyst is provided with the fragment of the threat model consisting of the assets, threats, and violated security objectives for the system as shown the first three columns of Table 3. By consulting the control catalogue, the analyst decides which of the atomic controls are relevant in protecting the system by referring to the identified threats in the threat model. The applicable atomic controls found for each threat can also be seen as part of Table 3. Notice that the mandatory control (*SI-10: Input Validation*) is

---

[2]ITSG-33 is the standard control catalogue to assist security practitioners in their efforts to protect information systems in compliance with applicable Government of Canada legislation, policies, directives, and standards [8].

included in the gathered set of atomic controls; the other controls therefore represent optional controls that may or may not be included as part of the suggested controls of this approach.

Table 3: Threat model and applicable atomic controls for *Firebird*

Assets	Threats	Security Objectives Violated	Applicable Atomic Controls
User Interface	• Commands received from unknown sources	• Confidentiality • Integrity	• *AC-4: Information Flow Enforcement*
	• Improper/malicious commands entered	• Confidentiality • Integrity	• *SI-10: Input Validation*
	• Employee freely accesses and changes features provided in the interface	• Confidentiality • Integrity	• *AC-3: Access Enforcement* • *AC-6: Least Privilege*
Database	• SQL injection from an improper analyst input changes or retrieves data	• Confidentiality • Integrity	• *AC-4: Information Flow Enforcement* • *SI-10: Input Validation*
	• Employee freely inspects data in the database	• Confidentiality	• *AC-6: Least Privilege*

4.2   Assign Effectiveness to Atomic Controls

The analyst must now assign the effectiveness values for each identified applicable atomic control at mitigating the threats and protecting the security objectives listed in Table 3. The analyst has elected to assign qualitative ratings for the effectiveness of each atomic control that are mapped to quantitative values. The considered ratings and corresponding values are adopted and adapted from the metrics in the Common Vulnerability Scoring System (CVSS) [20] and include: *None* (0.0), *Low* (0.2), *Medium* (0.5), *High* (0.8), and *Very High* (0.9). No rating was assigned to the value of 1 as it is unrealistic to expect a single control to fully protect a security objective.

  With these metrics, the analyst develops the atomic payoff matrix, as illustrated in Table 4. Note that none of the identified controls protect asset availability; this is fine as no threats towards availability were identified in the threat model (see Table 3). Therefore, protecting availability is not required.

Table 4: Atomic payoff matrix for *Firebird*

	Database			User Interface		
	C	*I*	*A*	*C*	*I*	*A*
SI-10: Input Validation	Medium	Very High	None	Medium	High	None
AC-3: Access Enforcement	None	None	None	Medium	High	None
AC-4: Information Flow Enforcement	Medium	Medium	None	Medium	Low	None
AC-6: Least Privilege	High	None	None	Medium	Low	None

4.3   Assign Cost to Atomic Controls

The analyst also assigns a cost for each identified atomic control as shown in Table 5. No units were used for each cost as it was decided that cost could best be represented as a unit of effort for this particular system. Additionally, the analyst's department has allocated a total budget (expressed as effort) of $B = 15$.

Table 5: Atomic control costs for *Firebird*

Control	Cost
SI-10: Input Validation	5
AC-3: Access Enforcement	6
AC-4: Information Flow Enforcement	4
AC-6: Least Privilege	3

4.4 Specify and Generate Valid Control Combinations

Next, the analyst must determine the valid security control combinations that could be considered for the system. To do this, they use security control algebra to specify the security control family from the mandatory and optional atomic controls identified in the previous steps. Recall that *SI-10: Input Validation* is a mandatory control and that *AC-3: Access Enforcement*, *AC-4: Information Flow Enforcement* and *AC-6: Least Privilege* are optional controls.

Suppose the security analyst has determined that to implement any access enforcement policy (such as Role-Based Access Control) a least privilege approach to protecting the data in the system must first be implemented. Therefore there is a dependency between *AC-3: Access Enforcement* and *AC-6: Least Privilege*. The analyst must consider this dependency in the specification of the security control family.

Denoting the security control family as F, the security control family for this example is specified as the following security control algebra term and requirement relation.

$$F = SI\text{-}10 \odot opt[AC\text{-}3, AC\text{-}4, AC\text{-}6] \qquad \text{such that} \qquad AC\text{-}3 \xrightarrow{F} AC\text{-}6$$

The possible security control combinations are generated by expanding the specification of the security control family F subject to the requirement relation. The possible security control combinations for F along with their costs calculated using Definition 3 are shown in Table 6. Note that the security control combinations *SI-10 AC-3* and *SI-10 AC-3 AC-4* are not part of the security control family F because they do not respect the specified requirement relation.

The analyst now determines the validity of the possible security control combinations according to the Budget Rule (Definition 4). Recall that the total budget B is 15. Therefore, applying the Budget Rule for each security control combination, it is easy to see that all control combinations, except for *Combo 6*, satisfy the rule and are therefore valid. As a result, *Combo 6* is no longer considered.

Table 6: Security control combination costs for *Firebird*

ID	Security Control Combination	Cost
Combo 1	SI-10	5
Combo 2	SI-10 AC-4	9
Combo 3	SI-10 AC-6	8
Combo 4	SI-10 AC-3 AC-6	14
Combo 5	SI-10 AC-4 AC-6	12
Combo 6	SI-10 AC-3 AC-4 AC-6	18

4.5 Construct the Game Matrix

Now that the analyst knows all of the valid security control combinations, the game matrix can be constructed. The payoff of each valid security control combination for each asset's security objectives is found by applying Definition 5. The resulting game matrix is shown in Table 7.

4.6 Play the Game

With the game matrix constructed, the security analyst can now find a suggested security combination to protect the security objectives of the considered assets for the system. To do this, the security analyst can play the game considering different attacker profiles captured by the scenarios described below. Table 8 presents the total effectiveness of each strategy in the game for each of the attacker objectives used in each scenario. Noteworthy effectiveness values are highlighted in bold. For strategies that have been excluded for specific attacker objectives, the corresponding effectiveness is noted as "N/A".

Table 7: Game matrix for *Firebird*

| | Database | | | User Interface | | |
	C	I	A	C	I	A
Combo 1	0.5	0.9	0.0	0.5	0.8	0.0
Combo 2	0.75	0.95	0.0	0.75	0.84	0.0
Combo 3	0.9	0.9	0.0	0.75	0.84	0.0
Combo 4	0.9	0.9	0.0	0.875	0.968	0.0
Combo 5	0.95	0.95	0.0	0.875	0.872	0.0

Table 8: Total effectiveness of game strategies against different attacker objectives for *Firebird*

| | Scenario 1 | Scenario 2 | Scenario 3 | | Scenario 4 | |
	AO1.1	*AO2.1*	*AO3.1*	*AO3.2*	*AO4.1*	*AO4.2*
Combo 1	1.0	2.7	0.5	N/A	0.9	N/A
Combo 2	1.50	3.29	0.75	N/A	**0.95**	1.59
Combo 3	1.65	3.39	0.75	N/A	0.9	N/A
Combo 4	1.775	3.643	**0.875**	**0.968**	0.9	N/A
Combo 5	**1.825**	**3.647**	**0.875**	0.872	**0.95**	**1.822**

Scenario 1: This scenario considers an attacker profile where the attacker equally targets the confidentiality of the database and the confidentiality of the user interface (*AO1.1*). By playing the game against this attacker, the suggested security control combination to implement is *Combo 5* because it has the greatest total effectiveness (1.825) for defending against the attacker objectives. Given that all identified threats impact the confidentiality of both assets, the suggested combination includes the optional controls which maximize confidentiality across both assets. While *AC-3* does not provide any protection to the confidentiality of the database, both *AC-4* and *AC-6* protect confidentiality across both assets. Given that *SI-10* is mandatory, *Combo 5* is the logical choice. Note that by disregarding *AC-3*, the threat related to "employee freely accesses and changes features provided in the interface" on the user interface is mitigated only through *AC-6*. Since *AC-6* is not as effective as *AC-3* for protecting integrity, the user interface's integrity is at higher risk of being violated. However, this is an acceptable risk given that the expected behaviour of the attacker is not interested in violating any integrity objectives.

Scenario 2: This scenario considers an attacker profile where the attacker equally targets all of the objectives (confidentiality, integrity, and availability) of each asset (database and user interface) (*AO2.1*). By playing the game against this attacker, the suggested security control combination to implement is *Combo 5* because it has the greatest total effectiveness (3.647) for defending against the attacker objectives. Given that this attacker profile aims to violate all security objectives on all assets, the suggested combination includes the optional controls which maximize the confidentiality and integrity across both assets. *AC-4* stands out in this regard, as it effectively safeguards all security objectives unlike *AC-3* and *AC-6*. While *AC-3* is not effective towards any of the database security objectives, *AC-6* at least offers protection towards the confidentiality of the database. Given that *SI-10* is mandatory, *Combo 5* is again the logical choice. Note that by disregarding *AC-4*, the same (acceptable) risk is imposed on the system as in Scenario 1. Also note that an assumed attacker profile targeting all objectives leads to a strategy that best balances the security objectives across all assets.

Scenario 3: This scenario considers an attacker profile where the attacker has two ordered attacker objectives to target: the confidentiality of the user interface (*AO3.1*) and then the integrity of the user

interface (*AO3.2*). By playing the game against this attacker, the suggested security control is determined by first considering how to best defend against the highest priority attacker objectives. This leaves *Combo 4* and *Combo 5* since they each have the greatest total effectiveness (0.875) for protecting the confidentiality of the user interface. Given that all controls provide the same effectiveness for the confidentiality of the user interface, any valid combination which maximizes this security objective is ideal. We now consider the next highest priority attacker objective from these possible security control combinations. Now, the suggested security control combination to implement is *Combo 4* because it has a greater total effectiveness (0.968 versus 0.872 for *Combo 5*) for protecting the integrity of the user interface. Given that *Combo 4* and *Combo 5* differ by only one control, and that *AC-3* is more effective at protecting the integrity of the user interface than *AC-4*, it follows that *Combo 4* is the logical choice. Note that by disregarding *AC-4*, the threat related to "commands received from unknown sources" on the user interface is not addressed. However, this is an acceptable risk given the expected attacker profile.

Scenario 4: This scenario considers an attacker profile where the attacker has two ordered attacker objectives to target: the integrity of the database (*AO4.1*) and then equally the confidentiality of the database and the integrity of the user interface (*AO4.2*). By playing the game against this attacker, the suggested security control is determined by first considering how to best defend against the highest priority attacker objective. This leaves *Combo 2* and *Combo 5* since they each have the greatest total effectiveness (0.95) for protecting the integrity of the database. These combinations are logical as *AC-4* is the only optional control that protects the integrity of the database. The next highest priority attacker objective must then be considered for these possible security control combinations. Now, the suggested security control combination to implement is *Combo 5* because it has a greater total effectiveness (1.822 versus 1.59 for *Combo 2*) for protecting confidentiality of the database and the integrity of the user interface. Given that *Combo 5* additionally contains *AC-6* which protects against the second set of attacker objectives, it follows that *Combo 5* is the logical choice. Again, as *Combo 5* disregards *AC-4*, the same risk is imposed on the system as in Scenario 1 and Scenario 2. However, this is an acceptable risk given the expected attacker profile.

This illustrative example and the corresponding scenarios highlight the fact that security control selection can indeed be seen as a game, and that the suggested controls to use depends greatly on the expected attacker profiles. The proposed approach considers all the constraints and considerations required to perform control selection, and in contrast to existing approaches, also takes into consideration the expected attacker behaviours; an important factor to this problem that helps justify controls of interest and the risk to leave certain threats unaddressed.

5   Discussion

Approaching security control selection as a game emphasizes the human element in deciding how to most effectively protect a system's assets under various considerations such as budgetary constraints. Selecting controls for a system is indeed a human-centric problem as the large number of potential controls to use from a control catalogue can be overwhelming and could lead to many mistakes in the chosen combination of security controls selected for the system. To expand on this point, selecting security controls exclusively on technical considerations while overlooking attacker behaviours is a fundamentally flawed approach in addressing this issue, as ultimately, it is humans which are conducting attacks. Given that the proposed approach emphasizes the need to reflect on potential attacker behaviours, it prioritizes

the human-centric aspect to find its solutions. Additionally, viewing this problem as a game captures the opposing dynamics of the attacker and analyst, aligning with the real-world motivations of both actors.

Unfortunately, a limitation with many existing game-theoretic approaches from addressing cybersecurity challenges is their heavy reliance on assumptions. Specifically, game theory depends on assumptions on the players, such as the players knowing every strategy available to them, knowing the probabilities of every move, and knowing the payoff functions [26]. Compared to existing works, the proposed approach can be used practically as it does not rely on assigning probabilities for the likelihood of attacks succeeding, and instead focuses on general assumptions about which security objectives could be violated by the attacker. Security analysts cannot realistically predict exact probabilities of attack, but they can make informed assumptions regarding which system components might be more attractive to attackers. These considerations ensure that the proposed approach is systematic, repeatable, and realistic, thereby minimizing the influence of human bias on the results of the game and eliminating many of the required assumptions with existing game-theoretic approaches.

While the proposed approach may seem limited by the security analyst's certainty in the effectiveness values for each atomic control, a sensitivity analysis was performed on the effectiveness metrics used in Section 4 and revealed that the results of the approach were not sensitive to these values. The full details of this analysis are omitted due to space limitations. In any case, to allow the analyst to express their uncertainty, it is possible to allow them to provide multiple values when assigning the effectiveness for atomic controls. The modifications to the game under these conditions is left for future work. The proposed approach also currently requires manual effort on the part of the security analyst. While some tasks are unavoidably manual (e.g.,, selecting applicable atomic controls and assigning effectiveness and costs to those controls), the rest of the approach can be supported with automated tools. Lastly, as atomic controls are considered indivisible components, we assume costs can be independently assigned to each atomic control. From a business perspective, this assumption may not always hold as the aggregation of certain controls could result in lower total costs to implement some control combinations. We argue that this limitation is unlikely to occur unless the controls are provided by third party vendors. In such cases, the cost function outlined in Definition 3 can be adapted to combine costs in a different manner.

6   Conclusions and Future Work

Ensuring effective security controls are selected for a system can greatly impact its security. In this work, a game-theoretic approach to security control selection is proposed in which a game is played by a security analyst to determine security controls which best mitigate expected attacker profiles. To create the game, the controls which are believed to secure the system must first be gathered. Following this, the effectiveness and cost of each of these controls is determined. After every possible control combination is generated, the effectiveness and cost of each combination is calculated and the game matrix can be constructed. The game can be played with many different expected attacker profiles and will suggest unique sets of controls for each. The suggested controls can help make a security analyst feel more confident in their decision to implement some controls over others.

In future work, we aim to extend the approach to consider more than one effectiveness value for each control to account for uncertainties in the effectiveness values assigned by the analyst. This would result in the game being played with more than one game matrix. The suggested controls from these different matrices could be compared to guide a security analyst in selecting the controls for the system. Additionally, to support the calculation of the game outcomes for large systems with many controls, we aim to develop software tools to automate aspects of the approach.

References

[1] Luís Almeida & Ana Respício (2018): *Decision support for selecting information security controls*. Journal *of Decision Systems* 27, pp. 173–180, doi:10.1080/12460125.2018.1468177.

[2] Seifeddine Bettaieb, Seung Yeob Shin, Mehrdad Sabetzadeh, Lionel C. Briand, Michael Garceau & Antoine Meyers (2020): *Using machine learning to assist with the selection of security controls during security assessment*. *Empirical Software Engineering* 25(4), pp. 2550–2582, doi:10.1007/s10664-020-09814-x.

[3] Jennifer Cawthra, Michael Ekstrom, Lauren Lusty, Julian Sexton & John Sweetnam (2020): *Data Integrity: Detecting and Responding to Ransomware and Other Destructive Events*. Special Publication (NIST SP) 1800-26, National Institute of Standards and Technology, doi:10.6028/NIST.SP.1800-26.

[4] Center for Information Security (2021): *CIS Critical Security Controls – Version 8*. `https://www.cisecurity.org/controls/v8` [Accessed: 2024-06-21].

[5] Rinku Dewri, Nayot Poolsappasit, Indrajit Ray & Darrell Whitley (2007): *Optimal security hardening using multi-objective optimization on attack tree models of networks*. In: *Proceedings of the 14th ACM Conference on Computer and Communications Security*, Association for Computing Machinery, New York, NY, USA, pp. 204–213, doi:10.1145/1315245.1315272.

[6] Victoria Drake: *Threat Modeling*. `https://owasp.org/www-community/Threat_Modeling` [Accessed: 2023-12-11].

[7] Martin S. Feather, Steven L. Cornford, Kenneth A. Hicks & Kenneth R. Johnson: (2005): *Applications of tool support for risk-informed requirements reasoning*. `https://www.researchgate.net/publication/220403935_Applications_of_tool_support_for_risk-informed_requirements_reasoning` [Accessed: 2024-06-21].

[8] Government of Canada (2014): *IT Security Risk Management: A Lifecycle Approach – Security Control Catalogue*. `https://www.cisecurity.org/controls/v8` [Accessed: 2024-06-21].

[9] Peter Höfner, Ridha Khedri & Bernhard Möller (2011): *An Algebra of Product Families*. *Software and Systems Modeling* 10(2), pp. 161–182, doi:10.1007/s10270-009-0127-2.

[10] International Organization for Standardization (2018): *ISO/IEC 31000:2018 Risk Management – Guidelines*. `https://www.iso.org/standard/65694.html` [Accessed: 2024-06-21].

[11] International Organization for Standardization (2022): *ISO/IEC 27002:2022 Information security, cybersecurity and privacy protection – Information security controls*. `https://www.iso.org/standard/75652.html` [Accessed: 2024-06-21].

[12] International Organization for Standardization (2022): *ISO/IEC 27005:2022 Information security, cybersecurity and privacy protection – Guidance on managing information security risks*. `https://www.iso.org/standard/80585.html` [Accessed: 2023-12-11].

[13] Joint Task Force Interagency Working Group (2018): *Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy*. Special Publication (NIST SP) 800-37 Revision 2, National Institute of Standards and Technology, doi:10.6028/NIST.SP.800-37r2.

[14] Joint Task Force Interagency Working Group (2020): *Control Baselines for Information Systems and Organizations*. Special Publication (NIST SP) 800-53B, National Institute of Standards and Technology, doi:10.6028/nist.sp.800-53b.

[15] Joint Task Force Interagency Working Group (2020): *Security and Privacy Controls for Information Systems and Organizations*. Special Publication (NIST SP) 800-53 Revision 5, National Institute of Standards and Technology, doi:10.6028/NIST.SP.800-53r5.

[16] Peter Kaloroumakis & Michael Smith (2020): *Toward a Knowledge Graph of Cybersecurity Countermeasures*. `https://apps.dtic.mil/sti/citations/AD1156977` [Accessed: 2024-06-21].

[17] Osamah Ibrahim Khalaf, Munsif Sokiyna, Youseef Alotaibi, Abdulmajeed Alsufyani & Saleh Alghamdi (2021): *Web Attack Detection Using the Input Validation Method: DPDA Theory*. *Computers, Materials & Continua* 68(3), doi:10.32604/cmc.2021.016099.

[18] Elmar Kiesling, Andreas Ekelhart, Bernhard Grill, Christine Strauss & Christian Stummer (2016): *Selecting security control portfolios: a multi-objective simulation-optimization approach*. EURO Journal on Decision Processes 4(1-2), pp. 85–117, doi:10.1007/s40070-016-0055-7.

[19] Qixu Liu & Yuqing Zhang (2011): *VRSS: A New System for Rating and Scoring Vulnerabilities*. Computer Communications 34, pp. 264–273, doi:10.1016/j.comcom.2010.04.006.

[20] Peter Mell, Karen Scarfone & Sasha Romanosky (2007): *The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agency Systems*. NIST Interagency Report 7435, National Institute of Standards and Technology, doi:10.6028/NIST.IR.7435.

[21] Microsoft (2022): *Microsoft Threat Modeling Tool – Threats*. `https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats` [Accessed: 2024-06-21].

[22] Murugiah Souppaya and Karen Scarfone (2016): *Guide to Data-Centric System Threat Modeling*. `https://csrc.nist.gov/pubs/sp/800/154/ipd` [Accessed: 2024-06-21].

[23] Mohamed Nassar, Joseph Khoury, Abdelkarim Erradi & Elias Bou-Harb (2021): *Game Theoretical Model for Cybersecurity Risk Assessment of Industrial Control Systems*. In: *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–7, doi:10.1109/NTMS49979.2021.9432668.

[24] National Institute of Standards and Technology (2020): *The NIST Privacy Framework: A Tool for Improving Privacy through Enterprise Risk Management*. Cybersecurity White Papers (CSWP) 10, National Institute of Standards and Technology, doi:10.6028/nist.cswp.10.

[25] National Institute of Standards and Technology (2024): *The NIST Cybersecurity Framework (CSF) 2.0*. Cybersecurity White Papers (CSWP) 29, National Institute of Standards and Technology, doi:10.6028/NIST.CSWP.29.

[26] Guillermo Owen (2015): *Game Theory*. In James D. Wright, editor: *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)*, second edition edition, Elsevier, Oxford, pp. 573–581, doi:10.1016/B978-0-08-097086-8.43045-X.

[27] Jun Young Park & Eui Nam Huh (2020): *A cost-optimization scheme using security vulnerability measurement for efficient security enhancement*. Journal of Information Processing Systems 16(1), pp. 61–82, doi:10.3745/JIPS.02.0128.

[28] Ron Ross, Victoria Pillitteri, Richard Graubart, Deborah Bodeau & Rosalie Mcquaid (2021): *Developing Cyber-Resilient Systems: A Systems Security Engineering Approach*. Special Publication (NIST SP) 800-160, Volume 2 Revision 1, National Institute of Standards and Technology, doi:10.6028/NIST.SP.800-160v2r1.

[29] Quentin Rouland, Stojanche Gjorcheski & Jason Jaskolka (2023): *Eliciting a Security Architecture Requirements Baseline from Standards and Regulations*. In: *2023 IEEE 31st International Requirements Engineering Conference Workshops*, REW, Hannover, Germany, pp. 224–229, doi:10.1109/rew57809.2023.00045.

[30] Theodoor Scholte, Davide Balzarotti & Engin Kirda (2012): *Have things changed now? An empirical study on input validation vulnerabilities in web applications*. Computers & Security 31(3), pp. 344–356, doi:10.1016/j.cose.2011.12.013.

[31] Theodoor Scholte, William Robertson, Davide Balzarotti & Engin Kirda (2012): *Preventing Input Validation Vulnerabilities in Web Applications through Automated Type Analysis*. In: *2012 IEEE 36th Annual Computer Software and Applications Conference*, pp. 233–243, doi:10.1109/COMPSAC.2012.34.

[32] Andrew M. Smith, Jackson R. Mayo, Vivian Kammler, Robert C. Armstrong & Yevgeniy Vorobeychik (2017): *Using computational game theory to guide verification and security in hardware designs*. In: *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 110–115, doi:10.1109/HST.2017.7951808.

[33] Philip D. Straffin (1993): *Game Theory and Strategy*, second edition. The Mathematical Association of America.

[34] Tony UcedaVélez & Marco M. Morana (2015): *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*, first edition. John Wiley & Sons, doi:10.1002/9781118988374.

[35] Baoyi Wang, Jianqiang Cai, Shaomin Zhang & Jun Li (2010): *A network security assessment model based on attack-defense game theory*. In: *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, 3, pp. V3–639–V3–643, doi:10.1109/ICCASM.2010.5620536.

[36] Iryna Yevseyeva, Vitor Basto-Fernandes, Michael Emmerich & Aad Van Moorsel (2015): *Selecting Optimal Subset of Security Controls*. *Procedia Computer Science* 64, pp. 1035–1042, doi:10.1016/j.procs.2015.08.625.