Optimal Offline Experimentation for Games

Abstract

Many business situations can be called “games” because outcomes depend on multiple decision makers with differing objectives. Yet, in many cases the payoffs for all combinations of player options are not available, but the ability to experiment offline is available. For example, war gaming exercises, test marketing, cyber range activities, and many types of simulations can all be viewed as offline gaming-related experimentation. We address the decision problem of planning and analyzing offline experimentation for games with an initial procedure seeking to minimize the errors in payoff estimates. Then, we provide a sequential algorithm with reduced selections from option combinations that are irrelevant to evaluating candidate Nash, correlated, cumulative prospect theory or other equilibria. We also provide an efficient formula to estimate the chance that a given Nash equilibria exists, provide convergence guarantees relating to general equilibria, and provide a stopping criterion called the estimated expected value of perfect offline information (EEVPOI). The EEVPOI is based on bounded gains in expected utility from further offline experimentation. An example of using a simulation model to illustrate all the proposed methods is provided based on a cyber security Capture the Flag (CTF) game. The example demonstrates that the proposed methods enable substantial reductions in both the number of test runs (half) compared with a full factorial and the computational time for the stopping criterion.

1. Introduction

In many realistic situations, the individual decision maker is not in complete control of all factor settings that influence outcomes. Instead, multiple decision makers select options and receive rewards that depend on the selections made by all players (Nash 1951). In many of these situations, it can be helpful to estimate the rewards for all possible player action combinations, perhaps focusing on the combinations most likely to be played. Here, we seek efficient experimental methods and stopping criteria to estimate mean rewards or utilities to support decision making where the starting point is access to low consequence experimentation, e.g., engagement simulations or experimental wargames.

One well-studied set of action combinations of potential interest is Nash equilibria which are setting options such that no player could benefit through individual adjustments. The relevance of Nash equilibria is rationalized by Expected Utility Theory (EUT, von Neumann and Morganstern 2007). Many other explanations for the relevance of Nash equilibria have been provided in the literature. A common view is the "self-enforcing" agreement relating to possible communications between players before play (Brandenberger and Dekkel 1987). In part because of the possibility of these agreements, wargame or other game designers can use the structure of the equilibria to suggest system improvements or other incentives to make the equilibria more desirable, i.e., "mechanism" design (Conitzer and Sandholm 2002, De Clippel, Saran, and Serrano 2018).

Correlated equilibria (Aumann 1987) and Cumulative prospect theory (CPT) equilibria are generalizations of Nash equilibria (Tversky and Kahneman 1992, Keskin 2016, Phade and Anantharam 2019). More general equilibria have motivations that include the subjective nature of probabilities and the irrationality of decision makers. Selten and Chmura (2008) study multiple types of equilibria and demonstrate that some make more accurate predictions of human behavior than Nash equilibria. The purpose of this article is to provide

algorithms for pre-experiments "offline" to support a variety of equilibria estimates and related mechanism design objectives.

Player rewards are often not known with certainty by all the players. This uncertainty may be an intrinsic property of the game requiring strategies for mitigation (e.g., see Harsanyi 1967). More commonly perhaps, it may be possible to learn the rewards and treat them afterwards as known constants. It is possible that some apparent violations of EUT motivating relevant generalizations such at Cumulative prospect theory (CPT, Keskin 2016 and Phade and Anantharam 2019) might relate to parametric uncertainty rather than irrationality. Also, much research addresses how players can learn to reduce the uncertainty by repeatedly playing the real game (e.g., see Foster et al. 2013, Chapman et al. 2016). For these problems, Nash equilibria are sometimes not considered to be relevant. Instead, learning the so-called no-regret decision options (analogous to Nash equilibria) is an important objective. Yet, what if the parameters are unknown and the game is not repeated?

To overview, we start pregame preparations with unknown payoff matrices but also with an ability to experiment offline, e.g., we have a simulation model. We play the game offline many times choosing actions for each player following our experimental planning and analysis methods. In each run, we observe the payoffs for all players. Then, we use metamodels to predict all the mean payoffs and determine whether offline experimental stopping conditions are met. Our goal is to predict the information needed to support decision making, e.g., the Nash or Cumulative prospect theory equilibria. With offline experimentation complete (at least temporarily), the decision maker can either re-design the related systems (e.g., mechanism design of a combat aircraft) and/or play the actual game (i.e., play the real game).

Therefore, this paper is focused on a new problem. The decision maker can reduce parametric uncertainty using so-called "offline" experimentation for variance reduction similar to risk reduction (Delquié 2012). For example, offline testing might involve a

simulation or a low consequence pre-game period. Even though the experiments are offline, they are not free. For example, creating test ranges and performing sets of war games might cost billions of dollars. Consider that one player may have ten or more options as may the opponent. Then, offline experimentation would need to support the estimation of literally hundreds of payoff parameters from simulation or real-world tests which may have replication or other errors.

The original examples of games in the Management Science literature include the design of advertising strategies and military tactics (Shubik 1955, 2002). In the context of modern online advertising, the internal working of the ad placement algorithms and the decisions of potential customers create an opaque game system. Through relatively inexpensive experimentation on test markets, decision makers can develop analytical models, predict estimated rewards, and enumerate equilibria to facilitate large scale campaigns. Similarly, in military (or cyber security) contexts simulation models with inputs from red and blue teams can be tested with replication to produce the inputs for game theoretic studies, leading to further insights into system vulnerabilities and strategic policy selections.

This paper makes several important contributions:

1. A new class of experimental design problems is introduced that support the enumeration of equilibria relevant to predicting behavior or mechanism design. The empirical exploration of offline systems can leverage the need to explore in detail option combinations that relate to settings that decision makers are likely to select.

2. A tractable and scalable sequential algorithm for offline experimentation to support single-period games is provided. Empirical models are developed and used to predict the expected rewards and the initial batch of experiments minimizes the prediction

variance over the player option combinations of interest. The algorithm converges to the true equilibria (Nash, correlated, Cumulative prospect theory or other) with probability 1.

3. To clarify the convergence properties, an analytical formula for the probability that specific Nash equilibria exist as a function of the available inputs is included. Monte Carlo can be applied for other equilibria. The probability relates to an integral over a multivariate normal distribution with inputs that depend on the data and often realistic assumptions.

4. Demonstration of the novel methods for a simulation-based cyber security Capture the Flag (CTF) red team/blue team game is given. By planning and executing experiments involving inputs from more than one decision maker, advice is provided both for participants in upcoming games as well as insights for game or system designers.

The rest of this paper is organized as follows. Section 2 introduces the notation and reviews studies of deterministic games, gaming under uncertainty, and relevant empirical modeling and experimental design methods. Section 3 combines the previous results to provide an initial heuristic procedure and an augmentation algorithm for equilibria estimation. Section 4 characterizes the experimental methods with respect to chances for identifying the true equilibria both for finite samples and asymptotically. Section 5 describes an application in cyber security. Section 6 offers conclusions and opportunities for future work. Note that some of the details about the cyber security case study are omitted because of space limitation, but these appear in the appendix.

2. Literature Review

This research combines methods and results from the game theory and experimental design literatures. We begin by introducing the notation. Then, we describe the literature and concepts relating to the basic game formulation, empirical modeling methods, and relevant experimental design results.

2.1. Notation

Our notation contains elements from both the game theory and experimental design literatures. Let m and n be the number of real game options for player A and player B respectively. We use A_{ij} and B_{ij} as the reward, if player A selects option i and player B selects option j, for player A and player B respectively. Also, $\hat{A}_{ij}$ and $\hat{B}_{ij}$ are the associated estimated quantities. Vectors $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ with dimensions mn are the vectorized mean values of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively with joint $(2mn) \times (2mn)$ covariance matrix $\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{AB}}$. Let q be the number of Nash equilibria (either true or estimated depending on the context). The number and location of the equilibria are uncertain because of our uncertainty about the rewards or, equivalently, the payoffs. The real game decision variables are $\boldsymbol{w}_A$ and $\boldsymbol{w}_B$ which represent probability over the m and n options for player A and player B respectively. The decision variable $\boldsymbol{w}_C$ would apply to a third player. The scalars α^i and β^i represent optimal payoff values that players A and B achieve at Nash equilibria i, and all the candidate equilibria are (w_A^i, w_B^i) for $i = 1, \dots, q$ or simply (w_A^0, w_B^0) for a specific candidate under consideration. Equilibria are "pure" if the vector has probability 1 on a single action or "mixed" otherwise. The vectors $\boldsymbol{e}$ and $\boldsymbol{l}$ have all entries equal to 1 and dimensions m and n respectively (and $\boldsymbol{o}$ is for Player 3). For three players, the tensors A, B, C are payoff cubes.

Here, each player action represents a combination of factor level settings. Also, we use regression models to predict the mean rewards for all combinations of player actions. Let K represent the number of regression model terms and N denote the number of experimental runs. The initial number is N_0, and M_A and M_B are the number of decision factors for players A and B respectively. The assumption parameters $\boldsymbol{\beta}_A$ and $\boldsymbol{\beta}_B$ relate model coefficients for predicting the player A and player B reward matrices respectively. The corresponding estimated quantities are $\widehat{\boldsymbol{\beta}}_A$ and $\widehat{\boldsymbol{\beta}}_B$. The standard deviations of repeated experimental outputs are σ_A and σ_B under the simplest equal variance assumption considered.

In the context of either linear regression or Gaussian Stochastic Regression (GSR) models, the random errors are N dimensional vectors $\boldsymbol{\varepsilon}_A$ and $\boldsymbol{\varepsilon}_B$ for players A and B respectively. For GSR, the correlation function between points is ϕ and the covariance matrix is $\boldsymbol{C}$. Because of experimental uncertainty, the existence of a candidate equilibrium (as defined in Section 3) is uncertain with probability $p_N(\boldsymbol{x}_0, \boldsymbol{y}_0)$.

Decision factors represent dimensions along which specific player options are available. For example, in a cyber security Capture the Flag example, player 1 has choices relating to whether to try to exploit the firewall or pivot immediately to the internal machine. We say that the factor is "firewall-pivot" and the levels are "first" and "never" which means that options or player policies are referred to as combination of factor levels. In this example, we are implying that the player policies are designed offline before the real game begins and then followed. Our analysis activity is intended to help the players design these policies. We believe that the decision factor decomposition of the strategy space is relevant for many real-world situations in which key policies are effectively set in a single round, e.g., the combinations of chess openings and team pre-set strategies for cyber Capture the Flag (CTF).

The experimental design decision variables include $x_{A,k,l}$ and $x_{B,k,r}$ for the setting selection for experimental run indexed by k, decision factors l and r and players A and B respectively. The vectors $\boldsymbol{x}_{A,k}$ and $\boldsymbol{x}_{B,k}$ are M_A and M_B dimensional vectors of settings for run k for players A and B respectively, with $(\boldsymbol{x}'_{A,k} | \boldsymbol{x}'_{B,k})$ in the region of interest set S_e. Corresponding values relate to the decision options in the game, which may be assumed to represent a discretization of the factor levels: $\tilde{x}_{A,i,l}$ and $\tilde{x}_{B,j,r}$ are the game settings for Player A option i and decision factor l for and Player B option j and decision factor r. These are $\tilde{\boldsymbol{x}}_{A,i}$ and $\tilde{\boldsymbol{x}}_{B,i}$ indexed to run i in vector form. The parameters $r_{i,j}$ weight the option combinations by subjective importance, all set to 1 by default. The vector $\mathbf{f}(\boldsymbol{x}_A, \boldsymbol{x}_B)$ is K dimensional and

includes the model terms (e.g., 1 and $x_{A,2}x_{B,1}$). The design matrix $\boldsymbol{X}$ is $N \times K$ corresponding to the model terms and experimental runs (for coefficient estimation). The vectors $\boldsymbol{Y}_A$ and $\boldsymbol{Y}_B$ are N dimensional response values of players A and B at the experimental points. These responses could be simulation game scores or the income from test markets.

The design matrix $\widetilde{\boldsymbol{X}}$ is $mn \times K$ based on the real game available options (for reward matrix estimation). Intermediate matrices for calculating equilibrium probabilities are $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, which are $m \times (mn)$ and $n \times (mn)$ respectively. Also, $\boldsymbol{T}(m,n)$ is an $(mn) \times (mn)$ permutation matrix. A key intermediate random vector, $\boldsymbol{Z}$, has dimension $(m+n)$.

In the context of sequential augmentation experimentation, the set of irrelevant option combinations for establishing whether all candidates are equilibria is $S_{irrelevant}$ and the associated random search parameter is $p_{irrelevant}$.

2.2. Bimatrix and Multiplayer Games

In the standard single period (bimatrix) game, player A sets the probability vector $\boldsymbol{w}_A$ and player B sets the probability vector $\boldsymbol{w}_B$. The standard formulation assumes that the reward or payoff matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are known. We preliminarily entertain this (often unrealistic) assumption for the sake of reviewing a seminal contribution of Nash (Nash 1951). With known $\boldsymbol{A}$ and $\boldsymbol{B}$, the rewards received for the players are derived using the joint formulation:

$$\begin{aligned} &\max_{\boldsymbol{w}_A} \boldsymbol{w}_A{}'\boldsymbol{A}\boldsymbol{w}_B \\ &\text{s.t } \boldsymbol{e}'\boldsymbol{w}_A - \boldsymbol{1} = 0; \boldsymbol{w}_A \geq 0, \\ &\max_{\boldsymbol{w}_B} \boldsymbol{w}_A{}'\boldsymbol{B}\boldsymbol{w}_B \\ &\quad \text{s.t } \boldsymbol{l}'\boldsymbol{w}_B - 1 = 0; \boldsymbol{w}_B \geq 0. \end{aligned} \tag{1}$$

The payoff values may, in general, represent mean profits or mean utilities. Here, we propose regression-based prediction of payoffs **A** and **B** from offline experiments and the assumption that these matrices represent mean utilities.

The joint formulation in Equation (1) leads, without loss of generality, to Nash equilibria (w_A^i , w_B^i) for $i = 1, \dots, q$. Selections not among these equilibria indicate (potentially) irrationality. Each Nash equilibrium satisfies the well-known property that player A cannot do better in the first optimization than w_A^i if player B does w_B^i and player B cannot do better in the second optimization than w_B^i if player A does w_A^i. Players can benefit by knowing the equilibria because they can select among them to maximize their game rewards. Game designers or system owners can benefit from knowing them because they may want to design incentives for players to change their behaviors.

Generalizations to multiplayer games have been explored extensively including generalizations of Nash equilibria (e.g., Phade and Anatharam 2019). Yet, the numbers of rewards needed to be estimated and the notational complexity grow with the number of players. For example, consider the extension of Nash Equilibria from a bimatrix game to a 3-player game. The payoff cubes are A, B, C and "$\otimes$" is the Kronecker product. Nash equilibria satisfy (Lee and Baldick 2003):

$$
\begin{aligned}
&A \otimes [w_A^i, w_B^i, w_C^i] \geq A \otimes [w_A, w_B^i, w_C^i] \text{ for all } w_A \in R^{N_1}, w_A \geq 0,\ {w_A}'e = 1,\\
&B \otimes [w_A^i, w_B^i, w_C^i] \geq B \otimes [w_A^i, w_B, w_C^i] \text{ for all } w_B \in R^{N_2}, w_B \geq 0, {w_B}'l = 1, \text{and}\\
&C \otimes [w_A^i, w_B^i, w_C^i] \geq C \otimes [w_A^i, w_B^i, w_C]] \text{ for all } w_C \in R^{N_3}, w_C \geq 0, {w_C}'o = 1.
\end{aligned} \tag{2}
$$

2.3. Equilibrium Conditions

Even with known $\boldsymbol{A}$ and $\boldsymbol{B}$, the general problems of finding the number of equilibria q and the actual equilibria (w_A^i, w_B^i) are NP-hard in terms of the numbers of options m and n (Chen and Deng 2006; Daskalakis et al. 2009; Conitzer and Sandholm 2008). However, state-of-the-art solution methods can practically enumerate equilibria for problems in which both players have hundreds of options (Savani and von Stengel 2015). Also, necessary and

sufficient conditions for the equilibria (Mangasarian and Stone 1964) relate to the existence of scalar α^0 and β^0 satisfying:

$$\begin{aligned}
&\boldsymbol{w}_A{}^{0'}\boldsymbol{A}\,\boldsymbol{w}_B{}^0 - \alpha^0 = 0,\\
&\boldsymbol{w}_A{}^{0'}\boldsymbol{B}\,\boldsymbol{w}_B{}^0 - \beta^0 = 0,\\
&\boldsymbol{A}\boldsymbol{w}_B{}^0 - \alpha^0\boldsymbol{e} \le \boldsymbol{0},\\
&\boldsymbol{B}'\boldsymbol{w}_A{}^0 - \beta^0\boldsymbol{l} \le \boldsymbol{0},\\
&\boldsymbol{e}'\boldsymbol{w}_A{}^0 - 1 = 0;\ \boldsymbol{w}_A{}^0 \ge \boldsymbol{0}, \text{ and}\\
&\boldsymbol{l}'\boldsymbol{w}_B{}^0 - 1 = 0;\ \boldsymbol{w}_B{}^0 \ge \boldsymbol{0}.
\end{aligned} \tag{3}$$

More general multi-player conditions like those in Equation (3) are available (Phade and Anatharam 2019). The key features of all correlated equilibria only involve the rows and columns associated with nonzero values of $\boldsymbol{w}_A$ and $\boldsymbol{w}_B$.

2.4. Empirical Prediction of Payoff Matrices

Key to our approach is the use of planned experiments and empirical regression models to predict simultaneously all the mean parameters in both payoff matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Whereas the decision variables are weights or probabilities (i.e., $\boldsymbol{w}_A$,$\boldsymbol{w}_B$) for real games, the empirical model building decision variables are the factor level settings (i.e., $\boldsymbol{x}_{A,i}, \boldsymbol{x}_{B,i}$) for offline experimental games. The experiments can be offline or not "real" in the sense that they do not require playing the game, e.g., one can experiment on a simulation model of the game as we illustrate for our cyber security planning example in Section 5. The experiments could also be relatively low consequence pre-experiments, e.g., involving test markets.

Consider that system options are potentially combinations of factor levels, i.e., option combination or run i is represented by the settings $(\boldsymbol{x}_{A,i}, \boldsymbol{x}_{B,i})$. The standard linear model functional form is $\mathbf{f}'(\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1})$. The "design" matrix (e.g., see Goos and Jones 2011) is:

$$\boldsymbol{X} = \begin{pmatrix} \mathbf{f}'(\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1}) \\ \vdots \\ \mathbf{f}'(\boldsymbol{x}_{A,N}, \boldsymbol{x}_{B,N}) \end{pmatrix}. \tag{4}$$

Gaussian process regression is a generalization of ordinary linear models (e.g., see Gorodetsky and Marzouk 2016). The multivariate expressions of the rewards, $\boldsymbol{Y}_A$ and $\boldsymbol{Y}_B$, derive from model coefficients, $\boldsymbol{\beta}_A$ and $\boldsymbol{\beta}_B$, and random errors, $\boldsymbol{\varepsilon}_A$ and $\boldsymbol{\varepsilon}_B$:

$$\begin{pmatrix}\boldsymbol{Y}_A\\ \boldsymbol{Y}_B\end{pmatrix} = \begin{pmatrix}\boldsymbol{X}\boldsymbol{\beta}_A\\ \boldsymbol{X}\boldsymbol{\beta}_B\end{pmatrix} + \begin{pmatrix}\boldsymbol{\varepsilon}_A\\ \boldsymbol{\varepsilon}_B\end{pmatrix}, \tag{5}$$

where the random errors could derive from simulation lack of repeatability, e.g., Monte Carlo random errors in cyber-attack simulations. A common assumption is that the random errors follow a multivariate normal (MN) distribution with variance covariance matrix, $\boldsymbol{\Sigma}_{\boldsymbol{AB}}$:

$$\begin{pmatrix}\boldsymbol{\varepsilon}_A\\ \boldsymbol{\varepsilon}_B\end{pmatrix} \sim MN[\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{AB}}]. \tag{6}$$

Here, we consider both the standard $\boldsymbol{\Sigma}_{\boldsymbol{AB}} = \sigma^2\boldsymbol{I}$ linear model regression assumption and a more general Gaussian stochastic regression (GSR) assumption in terms of scalar variance parameter, τ, variance, σ, directional parameters, θ_k, and variance-covariance matrix, $\boldsymbol{C}$. This gives:

$$\boldsymbol{\Sigma}_{\boldsymbol{AB}} = \boldsymbol{I}\tau^2 + \begin{bmatrix}\boldsymbol{C} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{C}\end{bmatrix}$$

$$\text{where } \boldsymbol{C} = \begin{bmatrix}\phi[(\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1}), (\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1})] & \cdots & \phi[(\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1}), (\boldsymbol{x}_{A,N}, \boldsymbol{x}_{B,N})]\\ \vdots & \ddots & \vdots\\ \phi[(\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1}), (\boldsymbol{x}_{A,N}, \boldsymbol{x}_{B,N})] & \cdots & \phi[(\boldsymbol{x}_{A,N}, \boldsymbol{x}_{B,N}), (\boldsymbol{x}_{A,N}, \boldsymbol{x}_{B,N})]\end{bmatrix}$$

$$\text{and } \phi[(\boldsymbol{x}_{A,i}, \boldsymbol{x}_{B,i}), (\boldsymbol{x}_{A,j}, \boldsymbol{x}_{B,j})] = \sigma^2 \exp\left[\sum_{k=1}^{n}\left(\frac{(\boldsymbol{x}_{A,i}, \boldsymbol{x}_{B,i}) - (\boldsymbol{x}_{A,j}, \boldsymbol{x}_{B,j})}{\theta_k}\right)^2\right]. \tag{7}$$

In our case study, we consider only categorical factors so continuous variable GSR is not relevant. Yet, we believe that the flexibility of GSR is critical for problems involving continuous factors and specific results apply to the more general GSR assumptions as we clarify. Using the standard linear model case assumption ($\boldsymbol{C} = \boldsymbol{0}$), the least squares coefficient estimates are: $\widehat{\boldsymbol{\beta}}_A = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}_A$ and $\widehat{\boldsymbol{\beta}}_B = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}_B$.

Then, the payoff matrix estimates, $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$, can be predicted using the regression in their vectorization forms, $vec(\widehat{\boldsymbol{A}})$ and $vec(\widehat{\boldsymbol{B}})$ and a full factorial design matrix $\widetilde{\boldsymbol{X}}$:

$$vec(\widehat{\boldsymbol{A}}) = \begin{pmatrix} \hat{A}_{1,1} \\ \vdots \\ \hat{A}_{n,1} \\ \vdots \\ \hat{A}_{i,j} \\ \vdots \\ \hat{A}_{n,m} \end{pmatrix} = \widetilde{\boldsymbol{X}}\widehat{\boldsymbol{\beta}}_A \text{ , } vec(\widehat{\boldsymbol{B}}) = \widetilde{\boldsymbol{X}}\widehat{\boldsymbol{\beta}}_B \text{, and where } \widetilde{\boldsymbol{X}} = \begin{pmatrix} \mathbf{f}'(\widetilde{\boldsymbol{x}}_{A,1}, \widetilde{\boldsymbol{x}}_{B,1}) \\ \vdots \\ \mathbf{f}'(\widetilde{\boldsymbol{x}}_{A,n}, \widetilde{\boldsymbol{x}}_{B,1}) \\ \vdots \\ \mathbf{f}'(\widetilde{\boldsymbol{x}}_{A,i}, \widetilde{\boldsymbol{x}}_{B,j}) \\ \vdots \\ \mathbf{f}'(\widetilde{\boldsymbol{x}}_{A,n}, \widetilde{\boldsymbol{x}}_{B,m}) \end{pmatrix}. \quad (8)$$

2.5. Experimental Design

It is well known that the accuracy of the empirical model greatly depends on the experimental design points used in its construction. In our game context, the experimental runs are pairings of level settings chosen by both players: $(\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1}), \dots, (\boldsymbol{x}_{A,N}, \boldsymbol{x}_{B,N})$. The accuracy also depends on the discrete points which form the options for the game: $(\widetilde{\boldsymbol{x}}_{A,1}, \widetilde{\boldsymbol{x}}_{B,1}), \dots, (\widetilde{\boldsymbol{x}}_{A,m}, \widetilde{\boldsymbol{x}}_{B,n})$. Typically, perhaps, there may be many more combinations of player options than experimental budgets can afford, i.e., $mn \gg N$. This makes the use of optimization particularly critical to permit prediction of the payoff matrices and thus accurate estimation of the Nash equilibria.

The accuracy of a linear model also depends on the terms included or the so-called functional form implied by the vector functions $\mathbf{f}_A{}'(\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1})$and $\mathbf{f}_B{}'(\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1})$ for deriving estimated payoff matrices $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ respectively. A concise and relevant functional form includes only the first order terms and Player A and Player B interactions:

$$\mathbf{f}_A{}'(\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1}) = \mathbf{f}_B{}'(\boldsymbol{x}_{A,1}, \boldsymbol{x}_{B,1}) =$$
$$(1 \quad x_{A,1} \quad \cdots \quad x_{A,M_A} \quad x_{B,1} \quad \cdots \quad x_{B,M_B} \quad x_{A,1}x_{B,1} \quad \cdots \quad x_{A,M_A}x_{B,M_B}) \quad (9)$$

For continuous variables, more detailed and accurate models may also be of interest including adding quadratic terms, e.g., $x_{A,1}^2$. The standard regression model with parameters $\widehat{\boldsymbol{\beta}}_A$ and $\widehat{\boldsymbol{\beta}}_B$ to predict a generic mean reward is:

$$\hat{y}_A = \mathbf{f}'(\tilde{\boldsymbol{x}}_A, \tilde{\boldsymbol{x}}_B)\widehat{\boldsymbol{\beta}}_A \text{ and } \hat{y}_B = \mathbf{f}'(\tilde{\boldsymbol{x}}_A, \tilde{\boldsymbol{x}}_B)\widehat{\boldsymbol{\beta}}_B. \quad (10)$$

These models have prediction variances of the form:

$$var[\hat{y}_A(\boldsymbol{x}_A, \boldsymbol{x}_B)] = \sigma_A^2 \mathbf{f}'(\boldsymbol{x}_A, \boldsymbol{x}_B)(\boldsymbol{X}'\boldsymbol{X})^{-1}\mathbf{f}(\boldsymbol{x}_A, \boldsymbol{x}_B) \text{ and}$$
$$var[\hat{y}_B(\boldsymbol{x}_A, \boldsymbol{x}_B)] = \sigma_B^2 \mathbf{f}'(\boldsymbol{x}_A, \boldsymbol{x}_B)(\boldsymbol{X}'\boldsymbol{X})^{-1}\mathbf{f}(\boldsymbol{x}_A, \boldsymbol{x}_B). \quad (11)$$

A natural objective to generate the initial experimental points is to minimize the average prediction errors over the lattice of player options. The standard prediction variance formula yields the following experimental design formulation. Gorodetsky and Marzouk (2016) provide a formulation relevant to Gaussian Process Regression. Here, we focus on an initial design with the linear model terms because our application has only categorical variables.

Assume that the relevant variance is σ_A^2 (could be σ_B^2) and $r_{i,j}$ is the weight for player A, game option i and player B, game option j. In our example, we assume that all option combinations are equally of interest ($r_{i,j} = 1$) at the start of experimentation but more general assumptions could be important. Therefore, Weighted Prediction Variance (WPV) reduces to the Average Prediction Variance (APV) and the optimization is over points in the experimental set $\boldsymbol{S}_e$. Then, the WPV formulation including the relevant option combinations for a fixed number of runs, N, is:

$$\underset{(\boldsymbol{x'}_{A,1}, \boldsymbol{x'}_{B,1}), \ldots \in \boldsymbol{S}_e}{\text{Minimize:}} \frac{\sigma_A^2}{mn} \sum_{j=1}^{n} \sum_{i=1}^{m} r_{i,j} \mathbf{f}'(\tilde{\boldsymbol{x}}_{A,i}, \tilde{\boldsymbol{x}}_{B,j})(\boldsymbol{X}'\boldsymbol{X})^{-1}\mathbf{f}(\tilde{\boldsymbol{x}}_{A,i}, \tilde{\boldsymbol{x}}_{B,j}).$$

For the $r_{i,j} = 1$ case, the APV can be simplified as:

$$\underset{(\boldsymbol{x'}_{A,1},\boldsymbol{x'}_{B,1}),\ldots\in \boldsymbol{S}_e}{\text{Minimize:}} \frac{\sigma_A^2}{mn}\text{Tr}[\widetilde{\boldsymbol{X}}'\widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}] \quad (12)$$

where mn is the number of decision points, the number of runs, N, is fixed by the dimensions of $\boldsymbol{X}$, and "Tr" is the trace or sum of the diagonal elements. Even with only linear regression modeling, the APV formulation in equation (12) is NP-hard (Ko, Lee, and Queyranne 1995). However, using the Meyer and Nachtsheim (1995) coordinate exchange algorithm with 1,000 random starting points, as suggested by Goos and Jones (2011, p. 36), all the problems considered here were approximately solved in 10 seconds to within 0.1% of optimality with JMP® software.

3. Experimental Procedures

In this section, we combine the previous results to create a one shot and sequential empirical equilibrium enumeration procedures. Also, we describe decision making about the initial number of experimental runs (N_0).

3.1. Initial Equilibria Estimation Procedure

Procedure 1 begins by optimally planning and executing offline test runs, e.g., game simulations. Then, the payoff matrix inputs to the bimatrix game formulation are predicted. Finally, the estimates can be used to enumerate the Nash or other equilibria with standard equilibrium enumeration methods (e.g., Savani and von Stengel 2015) based on the approximate assumption that the bimatrix inputs in Equation (1) are known.

Procedure 1 (Initial Experimentation and Equilibria Estimation)

1. Identify the factors levels for experimentation and mn game combinations of interest.
2. Solve the APV formulation in Equation (12) with N_0 runs (see Section 3.2 for choosing).
3. Collect experimental data following the optimal plan to derive the vectors $\boldsymbol{Y}_A$ and $\boldsymbol{Y}_B$.

4. Estimate the empirical model parameters, e.g., using least squares estimation.
5. Estimate the payoff matrices $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ using coefficients $\widehat{\boldsymbol{\beta}}_A$ and $\widehat{\boldsymbol{\beta}}_B$ with Equation (8).
6. Derive the candidate equilibria, (w_A^i, w_B^i) for $i = 1, \dots, q$, e.g., by solving Equation (3) assuming $\boldsymbol{A} = \widehat{\boldsymbol{A}}$ and $\boldsymbol{B} = \widehat{\boldsymbol{B}}$.

In general, candidate equilibria from empirical procedures (such as Procedure 1) may not be true equilibria of interest. This follows because experimental random errors (and model bias) can make it so that $\boldsymbol{A} \neq \widehat{\boldsymbol{A}}$ and $\boldsymbol{B} \neq \widehat{\boldsymbol{B}}$. Yet, with sufficiently large experiments, i.e., $N \gg 0$, the empirically derived equilibria can be expected to converge to the desired equilibria as we describe in Section 4.

3.2. Average Prediction Variance Designs

As mentioned previously, the Average Prediction Variance (APV) objective offers a natural formulation for initial experimentation. Intuitively, if the variances of parameters $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ are minimized, the chances of identifying key insights and the correct equilibria improve. Also, excluding interactions from the model that are not critical, e.g., third order or higher interactions, is a technique commonly used in many types of experiments to limit costs (Goos and Jones 2011). In bimatrix games, the interactions between player one and player two variables are intuitively critical for identifying equilibria. Related results can likely be established rigorously because, without interactions, players could optimize separately, making the games uninteresting to play and study. Higher order interactions are likely critical for games with multiple players. Therefore, regression model forms will need to be changed to consider multiplayer games.

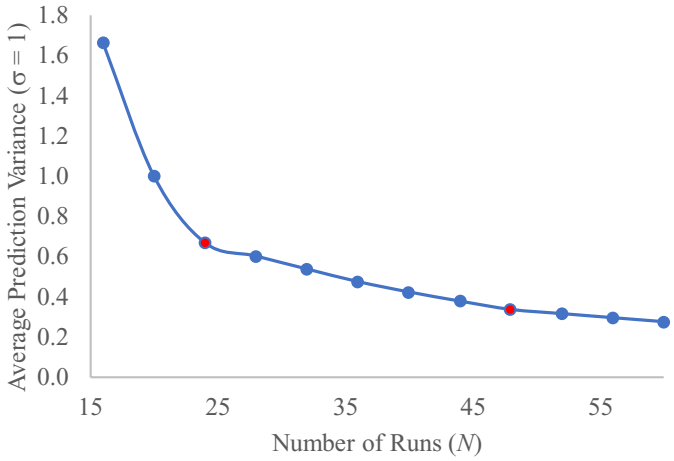
Selecting the initial number of runs, N_0, unavoidably involves some amount of subjectivity. In general, N_0, values greater than or equal to the number of model terms, K,

are desirable so that the estimation problem is fully determined. Yet, setting N_0 at much less than the full number of option combinations, mn, is also desirable to reduce costs.

Table 1 shows the $N_0 = 24$ experimental plan used in the cyber security Capture the Flag (CTF) case study. Additional information about the five factors involved is provided in Section 5. Here, we simply note that the model form in Equation (9) includes the three-level categorical factor "Firewall-Pivot" ($x_{A,i,1}$) and $K = 16$ terms.

Using JMP® software, it is possible to solve the APV formulation with equal weights ($r_{i,j} = 1$ for $i = 1, \dots, m$ and $j = 1, \dots, n$) for varying numbers of runs. Figure 1 shows the plot of the Average Prediction Variance for these optimal designs. The plot shows the typical "elbow" curve such that incurring additional expense beyond a certain point carries diminishing returns for prediction accuracy. This explains the choice of $N_0 = 24$ for the case study shown in Table 1. Note that the full factorial experimental design is APV optimal or near optimal with $N = 48$ for this problem. Yet, $N = 48$ offers only an incremental benefit for prediction accuracy as evidenced in Figure 1 on the right-hand-side.

Figure 1. Minimum average variances for designs with variable run numbers (N).



It is true, however, that using design with $N > 16$ (the minimum number for estimating the model we employ) could permit a more complete model form than the one in

Equation (9) to be fitted. Yet, in the context of the case study, the resulting model would (with high probability see Section 5) result in the same Nash equilibrium being identified with increased experimental cost.

Table 1. Average variance experimental design for cyber security case study with five factors and the indexed option selections.

Run	Firewall-Pivot ($x_{A,i.1}$)	Tenacity ($x_{A,i.2}$)	Start ($x_{B,i.1}$)	Forensic Firewall ($x_{B,i.2}$)	Tenacity ($x_{B,i.3}$)	Player 1	Player 2	$Y_{A,i}$	$Y_{B,i}$
1	First	Persist	Firewall	Never	Persist	1	3	40.7	60.8
2	Never	Persist	PC	Inspect Firewall	Move On	3	5	63.2	50.5
3	Never	Persist	PC	Never	Persist	3	4	52.6	30.3
4	Wait	Persist	Firewall	Never	Move On	2	7	50.8	64.6
5	Never	Move On	Firewall	Inspect Firewall	Persist	6	1	64.1	58.0
6	Never	Move On	PC	Inspect Firewall	Move On	6	6	47.7	64.0
7	Never	Persist	Firewall	Inspect Firewall	Persist	3	1	63.9	56.8
8	Never	Move On	Firewall	Never	Move On	6	7	41.5	70.7
9	Wait	Persist	Firewall	Inspect Firewall	Persist	2	1	54.1	61.4
10	Wait	Persist	PC	Inspect Firewall	Move On	2	6	51.6	47.8
11	Never	Persist	Firewall	Never	Move On	3	7	52.4	67.6
12	Wait	Persist	PC	Never	Persist	2	4	49.8	28.9
13	Wait	Move On	Firewall	Never	Move On	5	7	40.7	63.6
14	First	Persist	Firewall	Inspect Firewall	Move On	1	5	37.7	60.7
15	First	Persist	PC	Inspect Firewall	Persist	1	2	43.3	50.9
16	First	Move On	PC	Inspect Firewall	Persist	4	2	35.9	56.2
17	First	Move On	PC	Never	Move On	4	8	38.8	31.6
18	First	Move On	Firewall	Inspect Firewall	Move On	4	5	35.2	49.9
19	First	Move On	Firewall	Never	Persist	4	3	37.6	63.0
20	Never	Move On	PC	Never	Persist	3	4	50.4	26.1
21	First	Persist	PC	Never	Move On	1	8	53.5	28.0
22	Wait	Move On	Firewall	Inspect Firewall	Persist	5	1	52.1	65.9
23	Wait	Move On	PC	Never	Persist	5	4	47.8	27.2
24	Wait	Move On	PC	Inspect Firewall	Move On	5	6	50.5	55.1

3.3. An Augmentation Procedure

After initial experimentation and equilibria estimation, significant uncertainties can remain, i.e., the variances in the $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ estimates in Equation (12) may not be negligible leading to uncertainty about the relevant equilibria. This depends on the variances of the experimental random errors σ_A^2 and σ_B^2 and, possibly, on the bias from the model form approximation in Equation (9). The initial procedure unavoidably generates at least a single candidate

equilibrium because of the fundamental equilibrium existence theorem and the enumeration algorithms (Nash 1951). Intuitively, some experimental options are irrelevant to establishing whether candidate equilibria are the true equilibria. Here, the equilibria considered could be any type. Yet, the structure of the irrelevant set may vary. Our proposed procedure applies to Nash and correlated equilibria because related conditions involve only specific row and column values (Phade and Anantharam 2019).

Definition. *True* equilibria are those that would be derived after all parametric uncertainty is removed (e.g., from suitable infinite experimentation).

Avoiding irrelevant experiments for finding true equilibria is the driving objective of the proposed random search method (Procedure 2). A key parameter of this method is the probability of selecting from among the irrelevant options, $p_{irrelevant}$. We suggest 0.05 as a default value to minimally evaluate likely unhelpful options.

Procedure 2 (Augmentation Algorithm)

1. Update model (e.g., using sample mean estimates for $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ or, alternatively, regression prediction) and the associated candidate equilibria (w_A^i, w_B^i) for $i = 1, \dots, q$.
2. (Optional) If stopping conditions are met, stop.
3. Update the irrelevant set:

$$S_{irrelevant} = \left\{i, j | w_{A,i}^k = 0, w_{B,i}^k = 0 \; \forall \; k = 1, \dots, q\right\}. \tag{13}$$

4. Sample $U \sim [0,1]$

   If $U \leq p_{irrelevant}$, Then, random sample $i, j \in S_{irrelevant}$.

   Else, random sample $i, j \notin S_{irrelevant}$.

5. Perform experiment at $(\widetilde{\boldsymbol{x}}_{A,i}, \widetilde{\boldsymbol{x}}_{B,j})$.
6. Go to Step 1.

In the next section, the rigorous properties of the irrelevant set, $S_{irrelevant}$, and the convergence properties of Procedure 2 are investigated. A stopping criterion relating to the probability identifying true equilibria is proposed.

4. Properties of Experimental Procedures

In this section, the finite sample and asymptotic convergence properties of the procedures in Section 3 are characterized. We start with the probability that candidates are correctly identified. Then, we apply the results in the context of the initial experimentation and analysis (Procedure 1) and provide convergence results of the augmentation methods (Procedure 2).

4.1. The Probability That a Candidate Is a Nash Equilibrium

Consider that $\boldsymbol{A}$ and $\boldsymbol{B}$ are uncertain in the sense that the decision maker does not know fully what they are, i.e., there is "parametric uncertainty" which can be reduced through experimentation. Given a current parametric uncertainty level in the payoff values, there is uncertainty about whether a candidate equilibrium $(\boldsymbol{w}_A^0, \boldsymbol{w}_B^0)$ would be discovered to be an equilibrium if all the parametric uncertainty were removed through experimentation.

Definition. The probability that a candidate equilibrium $(\boldsymbol{w}_A^0, \boldsymbol{w}_B^0)$ is a true equilibrium, p_N, is the chance that the equilibrium is a Nash equilibrium for a random realization of the payoff matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$.

Theorem 1 provides a method to calculate this probability without the need for time-consuming Monte Carlo simulation of entire enumeration procedures. It also provides insights relating to the data sufficiency of many types of empirical methods for Nash equilibria estimation. The theorem starts with a given candidate equilibrium $(\boldsymbol{w}_A^0, \boldsymbol{w}_B^0)$, which is a pair of probability vectors. The theorem is based on assumed known values for

the means $(\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)$ and variances $(\boldsymbol{V}_A, \boldsymbol{V}_B)$ of the vectorized payoff matrices, i.e., $vec(\boldsymbol{A})$ and $vec(\boldsymbol{B})$. A key permutation matrix is $\boldsymbol{T}(m,n)$ which relates $vec(\boldsymbol{B}')$ to $vec(\boldsymbol{B})$. These payoff matrices are assumed to be multivariate normally distributed, MN, which is an often-relevant assumption for estimates derived from regression models. This theorem is relevant for general Gaussian stochastic regression (GSR), which is also based on a multivariate normal distribution, i.e., not merely least squares regression.

Theorem 1. Assume that the payoff matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are multivariate normally distributed: $\begin{bmatrix} vec(\boldsymbol{A}) \\ vec(\boldsymbol{B}) \end{bmatrix} \sim MN\left[\begin{pmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{pmatrix}, \widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{AB}}\right]$. Then the probability that the candidate feasible solution $(\boldsymbol{w}_A^0, \boldsymbol{w}_B^0)$ is a Nash equilibrium for a random instance of $\boldsymbol{A}$ and $\boldsymbol{B}$, p_N, is:

$$p_N(\boldsymbol{w}_A{}^0, \boldsymbol{w}_B{}^0) = \Pr\big(\{Z_1 \le 0\} \cap \{Z_2 \le 0\} \cap \ \ldots \{Z_{m+n} \le 0\}\big) \tag{14}$$

where $\boldsymbol{Z}$ is a random $m+n$ dimensional vector. The distribution of $\boldsymbol{Z}$ is defined in terms of $\boldsymbol{T}(m,n)$ which is a matrix that converts the vectorization of a $m \times n$ matrix into its transpose vectorization as:

$$\boldsymbol{Z} \sim MN\left[\begin{pmatrix} \boldsymbol{W}_1\boldsymbol{\mu}_A \\ \boldsymbol{W}_2\boldsymbol{\mu}_B \end{pmatrix}, \begin{bmatrix} \boldsymbol{W}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}_2 \end{bmatrix} \widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{AB}} \begin{bmatrix} \boldsymbol{W}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}_2 \end{bmatrix}'\right] \tag{15}$$

and where $\boldsymbol{W}_1 = \left(\boldsymbol{w}_B{}^{0\prime} \otimes \left(\boldsymbol{I} - \boldsymbol{e}\boldsymbol{w}_A{}^{0\prime}\right)\right)$ and $\boldsymbol{W}_2 = \left(\boldsymbol{w}_A{}^{0\prime} \otimes \left(\boldsymbol{I} - \boldsymbol{l}\boldsymbol{w}_B{}^{0\prime}\right)\right)\boldsymbol{T}(m,n)$. (16)

Proof. We seek to show that the event in Equation (14) is equivalent to the necessary and sufficient conditions in Equation (3) (Mangasarian and Stone 1964). If this is demonstrated, the probability in Equation (14) is the probability that $(\boldsymbol{w}_A^0, \boldsymbol{w}_B^0)$ is a Nash equilibrium. The last sets of constraints in Equation (3) are satisfied automatically since $(\boldsymbol{w}_A^0, \boldsymbol{w}_B^0)$ is a feasible solution. The first two constraints in Equation (3) are $\alpha^0 = \boldsymbol{w}_A{}^{0\prime}\boldsymbol{A}\,\boldsymbol{w}_B{}^0$ and $\beta^0 = \boldsymbol{w}_A{}^{0\prime}\boldsymbol{B}\boldsymbol{w}_B{}^0$. Plugging these into the following inequalities in Equation (3) and rearranging using scalar properties gives:

$$\boldsymbol{A}\boldsymbol{w}_B{}^0 - \boldsymbol{w}_A{}^{0\prime}\boldsymbol{A}\,\boldsymbol{w}_B{}^0\boldsymbol{e} = \boldsymbol{I}\boldsymbol{A}\boldsymbol{w}_B{}^0 - \boldsymbol{e}[\boldsymbol{w}_A{}^{0\prime}\boldsymbol{A}\,\boldsymbol{w}_B{}^0] \le \boldsymbol{0}.$$

$$(\boldsymbol{I} - \boldsymbol{e}\boldsymbol{w}_A{}^{0\prime})\boldsymbol{A}\boldsymbol{w}_B{}^{0} \leq \boldsymbol{0}. \tag{17}$$

Using the Knonecker product ($\otimes$) and a standard vectorization (vec) identity (Searle 1982, p. 333) gives:

$$\begin{aligned}(\boldsymbol{I} - \boldsymbol{e}\boldsymbol{w}_A{}^{0\prime})\boldsymbol{A}\boldsymbol{w}_B{}^{0} &= vec\left((\boldsymbol{I} - \boldsymbol{e}\boldsymbol{w}_A{}^{0\prime})\boldsymbol{A}\boldsymbol{w}_B{}^{0}\right) \\ &= \left(\boldsymbol{w}_B{}^{0\prime} \otimes (\boldsymbol{I} - \boldsymbol{e}\boldsymbol{w}_A{}^{0\prime})\right) vec(\boldsymbol{A}) = \boldsymbol{W}_1 vec(\boldsymbol{A}) \leq \boldsymbol{0}. \end{aligned} \tag{18}$$

Similarly, we have:

$$\begin{aligned}&\boldsymbol{B}'\boldsymbol{w}_A{}^{0} - (\boldsymbol{w}_A{}^{0\prime}\boldsymbol{B}\,\boldsymbol{y}^{0})\boldsymbol{l} = \boldsymbol{I}\boldsymbol{B}'\boldsymbol{w}_A{}^{0} - \boldsymbol{l}(\boldsymbol{w}_B{}^{0\prime}\boldsymbol{B}\,\boldsymbol{w}_A{}^{0}) \leq \boldsymbol{0} \\ &(\boldsymbol{I} - \boldsymbol{l}\boldsymbol{w}_B{}^{0\prime})\boldsymbol{B}'\boldsymbol{w}_A{}^{0} = \left(\boldsymbol{w}_A{}^{0\prime} \otimes \left(\boldsymbol{I} - \boldsymbol{l}\boldsymbol{w}_B{}^{0\prime}\right)\right) vec(\boldsymbol{B}') \\ &= \left(\boldsymbol{w}_A{}^{0\prime} \otimes \left(\boldsymbol{I} - \boldsymbol{l}\boldsymbol{w}_B{}^{0\prime}\right)\right)\boldsymbol{T}(m,n)\, vec(\boldsymbol{B}) = \boldsymbol{W}_2 vec(\boldsymbol{B}) \leq \boldsymbol{0}. \end{aligned} \tag{19}$$

Together, Equations (18) and (19) give $m + n$ inequalities. Introducing the random vector, $\boldsymbol{Z}$, Equations (18) and (19) become:

$$\boldsymbol{Z} = \begin{bmatrix}\boldsymbol{W}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}_2\end{bmatrix}\begin{pmatrix}vec(\boldsymbol{A}) \\ vec(\boldsymbol{B})\end{pmatrix} \leq \begin{pmatrix}\boldsymbol{0} \\ \boldsymbol{0}\end{pmatrix}. \tag{20}$$

Substituting the assumption for the multivariate payoff matrices, $vec(\boldsymbol{A})$ and $vec(\boldsymbol{B})$, and using standard matrix identities (Searle 1982 p. 400), $\boldsymbol{Z}$ is multivariate normal:

$$\boldsymbol{Z} \sim MN\left[\begin{pmatrix}\boldsymbol{A}_1\boldsymbol{\mu}_A \\ \boldsymbol{A}_2\boldsymbol{\mu}_B\end{pmatrix}, \begin{bmatrix}\boldsymbol{W}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}_2\end{bmatrix}\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{AB}}\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_2\end{bmatrix}'\right]. \tag{21}$$

Therefore, the necessary and sufficient conditions for an equilibrium are satisfied if and only if each element of the vector, $\boldsymbol{Z}$, is negative. Then, $p_N(\boldsymbol{w}_A{}^{0}, \boldsymbol{w}_B{}^{0})$ is equal to the probability that this condition occurs assuming Equation (21). ∴

The central result in Theorem 1 is intuitive. Once there is a candidate equilibrium, there is no need to enumerate all the equilibria for thousands of model scenarios for probability estimation. Instead, simulations need only study whether the small number of normally distributed parameters exceed the other normally distributed parameters

involved in the equilibrium. Equation (14) supports efficient estimation while addressing correlated prediction estimates and mixed equilibria, i.e., fractional probabilities. From the estimated equilibria, the player may be able to see which equilibrium an opponent would likely prefer (if there are more than one). If there is only one equilibrium, as appears to be the case in our cyber game example, it may be highly desirable to play related settings.

Note that Theorem 1 could apply to uncertainty of types other than parametric, i.e., uncertainty not caused from limited experimental data. For example, it could relate to games with intrinsically random payoffs. Also, Theorem 1 permits the computationally efficient estimation of Nash equilibria probabilities. The probabilities for more general equilibria such as correlated or Cumulative prospect theory equilibria can, of course, be estimated using Monte Carlo simulation. Further, regardless of the specific stopping criterion applied, the information about the rewards can support many types of decision support activities. Next, we relate Theorem 1 to empirical uncertainties.

4.2. Application to Procedure 1

In real empirical investigations, the analyst does not have the mean values of the payoff estimates, $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$, and the true covariances, $\boldsymbol{\Sigma}_{\boldsymbol{AB}}$. Instead, the analyst has response data, $\boldsymbol{Y}_A$ and $\boldsymbol{Y}_B$ which can be derived from Procedure 1. From this data, estimates can be generated using regression, e.g., linear regression in Equation (8). Many types of regression models are multivariate normal distributions. The standard linear regression multivariate normal assumption inspires the following corollary to Theorem 1 which characterizes the finite sample properties of Procedure 1.

Corollary 1. Assume the experimental outputs derive from a standard linear model with variances σ_A^2 and σ_B^2, i.e., $\begin{pmatrix}\boldsymbol{Y}_A\\ \boldsymbol{Y}_B\end{pmatrix} \sim MN\left[\begin{pmatrix}\boldsymbol{X\beta}_A\\ \boldsymbol{X\beta}_B\end{pmatrix}, \begin{bmatrix}\sigma_A^2\boldsymbol{I} & \mathbf{0}\\ \mathbf{0} & \sigma_B^2\boldsymbol{I}\end{bmatrix}\right]$. Also, the functional form, $\mathbf{f}(\boldsymbol{x}_A, \boldsymbol{x}_B)$, for the design matrix, $\boldsymbol{X}$, is the same as for the prediction design matrix $\widetilde{\boldsymbol{X}}$. Further,

Procedure 1 is applied. Then, the probabilities that any given candidate equilibrium (indexed by i) is a true equilibrium, $p_N\left(w_A^i, w_B^i\right)$, is given by Equation (14) substituting $\boldsymbol{\mu}_A = \widetilde{\boldsymbol{X}}\boldsymbol{\beta}_A$, $\boldsymbol{\mu}_B = \widetilde{\boldsymbol{X}}\boldsymbol{\beta}_B$, $\left(w_A^i, w_B^i\right) = (\boldsymbol{w}_A^0, \boldsymbol{w}_B^0)$, and

$$\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{AB}} = \begin{bmatrix} \sigma_A^2\widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\widetilde{\boldsymbol{X}}' & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_B^2\widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\widetilde{\boldsymbol{X}}' \end{bmatrix}. \tag{22}$$

Proof. Establishing the conditions for Theorem 1 proves that the probability applies. Applying Procedure 1 involves using the least squares estimators gives:

$$vec\left(\widehat{\boldsymbol{A}}\right) = \widetilde{\boldsymbol{X}}\widehat{\boldsymbol{\beta}}_A = \widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}_A \tag{23}$$

such that each entry is a linear combination of normally distributed random variables and thus is also multivariate normal. Similarly, $vec\left(\widehat{\boldsymbol{B}}\right)$ is multivariate normal. The mean vector, $\boldsymbol{\mu}_A$, is (for mean zero random $\boldsymbol{\varepsilon}_A$):

$$\begin{aligned} \boldsymbol{\mu}_A &= E\left[vec\left(\widehat{\boldsymbol{A}}\right)\right] = E\left[\widetilde{\boldsymbol{X}}\widehat{\boldsymbol{\beta}}_A\right] = E\left[\widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}_A\right] \\ &= E[\widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\boldsymbol{X}\boldsymbol{\beta}_A + \boldsymbol{\varepsilon}_A)] = \widetilde{\boldsymbol{X}}\boldsymbol{\beta}_A. \end{aligned} \tag{24}$$

Similarly, $\boldsymbol{\mu}_B = \widetilde{\boldsymbol{X}}\boldsymbol{\beta}_B$. The variance-covariance matrix is derived using $var[\boldsymbol{T}\boldsymbol{x}] = \boldsymbol{T}var[\boldsymbol{x}]\boldsymbol{T}'$ where $\boldsymbol{T}$ is an arbitrary matrix:

$$\begin{aligned} var\left[vec\left(\widehat{\boldsymbol{A}}\right)\right] &= var\left[\widetilde{\boldsymbol{X}}\widehat{\boldsymbol{\beta}}_A\right] = Var\left[\widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}_A\right] \\ &= \widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'var[\boldsymbol{Y}_A]\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\widetilde{\boldsymbol{X}}' = \widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\begin{bmatrix} \sigma_A^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_B^2\boldsymbol{I} \end{bmatrix}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\widetilde{\boldsymbol{X}}' \\ &= \begin{bmatrix} \sigma_A^2\widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\widetilde{\boldsymbol{X}}' & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_B^2\widetilde{\boldsymbol{X}}(\boldsymbol{X}'\boldsymbol{X})^{-1}\widetilde{\boldsymbol{X}}' \end{bmatrix}. \end{aligned} \tag{25}$$

Therefore, the payoff matrix estimates are multivariate normal with the prescribed mean and covariance and Theorem 1 applies. ∴

Remark: Corollary 1 inspires an approximate method to estimate the probability that a given candidate equilibrium, $(\boldsymbol{w}_A^0, \boldsymbol{w}_B^0)$, is a true equilibrium. Procedure 1 generates the

regression estimates: $\widehat{\boldsymbol{\beta}}_A, \widehat{\boldsymbol{\beta}}_B, \hat{\sigma}_A^2$, and $\hat{\sigma}_B^2$. Then, one can assume that $\widehat{\boldsymbol{\beta}}_A = \boldsymbol{\beta}_A, \widehat{\boldsymbol{\beta}}_B = \boldsymbol{\beta}_B, \hat{\sigma}_A^2 = \sigma_A^2$, and $\hat{\sigma}_B^2 = \sigma_B^2$ and use Equation (22) for inputs to estimate the probability the equilibrium is a Nash equilibrium from Theorem 1 in Equation (14). This approach can function as a stopping criterion in Procedure 2, i.e., stop when the estimated probabilities for all equilibria exceed a threshold such as 95%.

4.3. Irrelevant Experimental Options

Consider a list of candidate Nash equilibria from Procedure 1. Some of these candidates may not be true equilibria. In determining the accuracy of the given candidate equilibria, some experimental options are irrelevant. Lemma 1 clarifies conditions under which option combinations are known to be irrelevant to equilibrium probability calculations.

Lemma 1. Assume that the payoff matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are random. The probability that $(\boldsymbol{w}_A^0, \boldsymbol{w}_B^0)$ is a Nash equilibrium neither depends on the values $A_{i,j}$ with $w_{B,j}^0 = 0$, nor on the values of $B_{i,j}$ with $w_{A,i}^0 = 0$.

Proof. If we can show that the values in question are irrelevant to establishing the necessary and sufficient conditions to be an equilibrium in Equation (3), the result is proven. From Equation (17) we have $(\boldsymbol{I} - \boldsymbol{e}\boldsymbol{w}_A{}^{0'})\boldsymbol{A}\boldsymbol{w}_B{}^0 \leq \boldsymbol{0}$ and from equation (18) we have $\left(\boldsymbol{I} - \boldsymbol{l}\boldsymbol{w}_B{}^{0'}\right)\boldsymbol{B}'\boldsymbol{w}_A{}^0 \leq \boldsymbol{0}$. From the proof of Theorem 1 (and intuition), these are the only dependence that the necessary and sufficient conditions have on the values of the payoff matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Therefore, columns of $A_{i,j}$ with $w_{B,j}^0 = 0$ are irrelevant and so are rows of $B_{i,j}$ with $w_{A,i}^0 = 0$. ∴

Extending this result to all correlated equilibria using the generalized equilibria conditions (Phade and Anatharam 2019) is proposed for future work. In a problem with numbers of equilibria smaller than the number of option combinations ($q \ll mn$), few payoff

values are relevant. Intuitively, knowing that some option combinations are irrelevant should down-weight their importance in empirical data augmentation procedures. This suggests the application of the random search Procedure 2.

4.4. Convergence Results

For finite amounts of offline data, there is some generally uncertainty about the Nash, correlated, Cumulative prospect theory or other equilibria. Lemma 1 sheds light solely on which experimental option combinations are relevant with respect to a given list of equilibria. Focusing only on relevant experimental options, therefore, could conceivably miss some of the equilibria, even in the limit of infinite experimentation. Theorem 2 clarifies the implications for long run applications of Procedure 2 augmentation.

Theorem 2. Assume the experimental outputs derive from a standard linear model with variances σ_A^2 and σ_B^2, i.e., $\begin{pmatrix} \boldsymbol{Y}_A \\ \boldsymbol{Y}_B \end{pmatrix} \sim MN\left[\begin{pmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{pmatrix}, \widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{AB}}\right]$ for finite $\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{AB}}$. Further, the Augmentation Procedure is applied with sample mean-based estimation and no stopping rule. Both:

i. If $0 < p_{irrelevant} < 1$, in the limit $N \to \infty$ then all true equilibria are identified with probability 1, and

ii. If $p_{irrelevant} = 0$, in the limit $N \to \infty$ then all identified candidate equilibria are true (correlated or Nash) equilibria with probability 1.

Proof. If $0 < p_{irrelevant} < 1$ then in the limit, $N \to \infty$, we have $sample\ mean(\boldsymbol{Y}_A) \to \boldsymbol{\mu}_A$ and $sample\ mean(\boldsymbol{Y}_B) \to \boldsymbol{\mu}_B$ because of the central limit theorem with finite variances. This implies that (w_A^i, w_B^i) for $i = 1, \dots, q$ are the equilibria. Assume that $p_{irrelevant} = 0$ and the limiting candidate set (w_A^i, w_B^i) for $i = 1, \dots, q$. Then, the relevant sample means converge and the others are irrelevant to the candidate list. Without loss of generality, we assume that

$sample\ mean(\boldsymbol{Y}_A) \to \boldsymbol{\mu}_A$ and $sample\ mean(\boldsymbol{Y}_B) \to \boldsymbol{\mu}_B$. This gives the result for the candidate correlated or Nash equilibria in both cases. ∴

Theorem 2 implies that focusing on the relevant set of experimental options can increase the probability that points on a given list of candidates are true equilibria. Also, some amount of focus on (apparently) irrelevant options could conceivably aid in the identification of all the true Nash equilibria. This follows because lists of equilibria from finite samples may be incomplete. The implications of data from irrelevant combinations for related regression model-based, sequential procedures are a topic for future research.

4.5. A Stopping Criterion Based on the Expected Value of Information

Even expanded notions of equilibria may not predict behavior accurately. Therefore, the decision maker may desire to entertain specific assumptions about the policies of an opponent (or opponents) and then decision making can be based on the expected value of perfect information, e.g., see Delquié (2012). Assume that a specific w_B^0 is known or assumed, perhaps from studying the offline simulations or because there is a single policy of interest. The estimated expected value of perfect offline information (EEVPOI) is:

$$EEVPOI = \underset{A,\varepsilon}{\mathrm{E}}[\max_{\boldsymbol{w}_A}((\boldsymbol{w}_A{}'(\boldsymbol{A}\boldsymbol{w}_B^0) + \varepsilon_A))] - \max_{\boldsymbol{w}_A} \underset{A,\varepsilon}{\mathrm{E}}[(\boldsymbol{w}_A{}'(\boldsymbol{A}\boldsymbol{w}_B^0) + \varepsilon_A)] \tag{26}$$

$$= \underset{A,\varepsilon}{\mathrm{E}}[\max_{\boldsymbol{w}_A}(\boldsymbol{w}_A{}'(\boldsymbol{A}\boldsymbol{w}_B^0)] - \max_{\boldsymbol{w}_A}(\boldsymbol{w}_A{}'(\widehat{\boldsymbol{A}}\boldsymbol{w}_B^0),$$

where $vec(\boldsymbol{A}) \sim MN[\widehat{\boldsymbol{\mu}}_A, \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{A}}]$ with $\widehat{\boldsymbol{\mu}}_A$ are regression mean prediction covariance $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{A}}$ matrix estimates. Instead of stopping offline experimentation based on the Nash or other equilibrium probability estimates, stopping can be based on threshold values of the EEVPOI. In other words, stop when the expected gain in the utility (bound) is sufficiently small.

5. An Application to Red Team/Blue Team Capture the Flag

For our application, we consider a simulated cyber security Capture the Flag (CTF) red team/blue team game. This game seeks to help train many types of students to learn both cyber security basics and related policy decision making. Additional details about the game and the discrete event simulation model are described in the appendix. Briefly, Figure 2 and Figure 3 show key tasks for red and blue team players respectively with a port scanner (e.g., NMAP or Unicornscan) being a program to find IP addresses and scanning and exploiting activities supported by many commercial software.

Next, we describe the application of experimental methods described in Section 3. Zero sum games have ($\boldsymbol{A} = -\boldsymbol{B}$). By default, we assume that the game is not zero sum to mirror real life cyber security but converting to a zero-sum game is not difficult.

In applying Procedure 1, we identify the factor levels for experimentation and mn game combinations of interest. Figure 2 shows two player A variables: firewall-pivot ($x_{A,i.1}$) and tenacity ($x_{A,i.2}$). A major policy decision that the red team needs to consider is what to do when they successfully exploit the firewall. Should they pivot to attempting to use the firewall as a general-purpose host, exfiltrating any data and launching external attacks or, alternatively, pivot to attacking the internal host? Possible levels include: using the firewall first, waiting until after the internal host is exploited, or never attempting to use the firewall other than to attack the internal host (first, wait, or never). Intuitively, firewalls rarely contain helpful data for exfiltration and the access of firewall hosts is generally inferior to that of internal hosts. Another major policy choice for attackers is when to give up attempting to use the hosts that they compromise, i.e., their tenacity. Levels include giving up immediately upon failure (move on) and never (persist).

Similarly, defenders have policy options as indicated in Figure 3. Unlike attackers, defenders can start their activities on either the firewall or the internal host because of their

insider access (firewall or PC). Also, defenders can choose to do forensics on the firewall (inspect firewall) or to ignore its state of compromise (never inspect). Intuitively, forensic activities are time consuming and the compromise state of the firewall is not as important as the state of the internal host. Finally, the tenacity of defenders' in patching attempts can be set. They can give up immediately upon failing to find a patch (move on) or they can persist in patching attempts (persist). Note that, in our analysis, the game options are the same as the experimental level combinations.

In Step 2 of Procedure 1, we formulate and solve the APV experimental design problem in Equation (12) with N_0=24 runs. The solution is in Table 1 based on the factor levels in Table 2. The choice of the number of runs, N_0 =24, reflects a balance between experimental economy and prediction model accuracy. It also represents the "elbow" point in Figure 1 as described in Section 3. In Step 3, we collect experimental data using the SIMIO model indicated in part in Figure 4 with the 10 replicates following the optimal plan to derive the vectors $\boldsymbol{Y}_A$ and $\boldsymbol{Y}_B$. In Step 4, we estimate the empirical model parameters using least squares estimation with the results shown in Table 3.

Figure 5 plots the prediction model showing the interactions of player level selections on the scores for both teams (e.g., XA1*XB1 for the first Player A-Player B interaction). The most interesting interaction relates to the choice of the red team never to pivot to use the firewall. This choice benefits both teams unless the blue team is persistent in its patching attempts. Intuitively, this occurs because exploitation of the firewall permits the blue team to successfully patch the internal host, making its later exploitation by the attackers significantly more difficult. Also, starting at the firewall is generally more beneficial for the blue team regardless of the red team selections.

Then, in Step 5 we estimate the bimatrix game parameters $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$, e.g., $\widehat{\boldsymbol{\beta}}_A$ and $\widehat{\boldsymbol{\beta}}_B$ using Equation (8) as shown in Table 4. Enumerating the equilibria which are solutions to

Equation (3) by inspection or using standard methods (Savani and von Stengel 2015) results in a single equilibrium as indicated in Table 4 (bolded). It is also apparently the only correlated equilibrium. The equilibrium is option 3 (never use the firewall and persist in using the internal host) for the red team and option 5 (start at the firewall, inspect the firewall, and move on if searching for a patch fails) for the blue team. Notice the Nash condition applies. The highest value in column 5 in Table 3 (a) corresponds to option 3 and the highest value in row 3 corresponds to option 5 in Table 3 (b).

This candidate equilibrium is then evaluated using 10,000 simulations based on Equation (14). The result predicts an estimated (approximate) 96.5% chance of being a true equilibrium based on the right-hand-side brute force python code solution of each scenario. For this brute force approach, the computation times is approximately 8.9 hours on an i5-3475 3.2 GHz CPU and python code. This estimate is approximate because correlations between predictions are ignored.

Using the right-hand-side of Equation (14) Monte Carlo estimated the exact probability of 95.0% using 0.49 seconds on the MATLAB cloud. This demonstrates the potentially critical computational advantage afforded by Theorem 1 in computational efficiency and accuracy. Therefore also, Procedure 2 immediately stops using only 24 runs, which is half of a full factorial ($3 \times 2 \times 2 \times 2 \times 2 = 48$ runs).

Similarly, if one assumes that Player 2 will play action 5 with probability one ($\boldsymbol{w}_B^0$), the estimated expected value of perfect offline information (EEVPOI) can be estimated to a good approximation using column 5 in Table 4(a). The table values give estimated means. A constant variance of 7.51 utility units squared is based on the regression results. Then, the first term in Equation (26) is 60.93 and the second term is 60.80 so that the EVPOI is 0.13 or 0.21%. This may be regarded as negligible such that offline experimentation can terminate.

Figure 2. CTF system diagram, red team tasks, and factors: firewall pivot and tenacity.

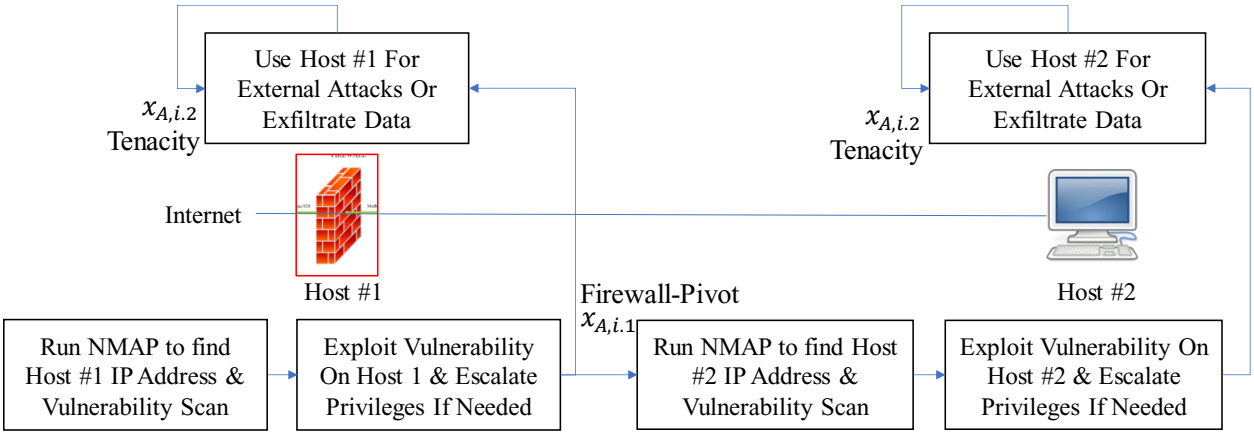


Figure 3. CTF system diagram, blue team tasks, and factors: start, tenacity, and forensics.

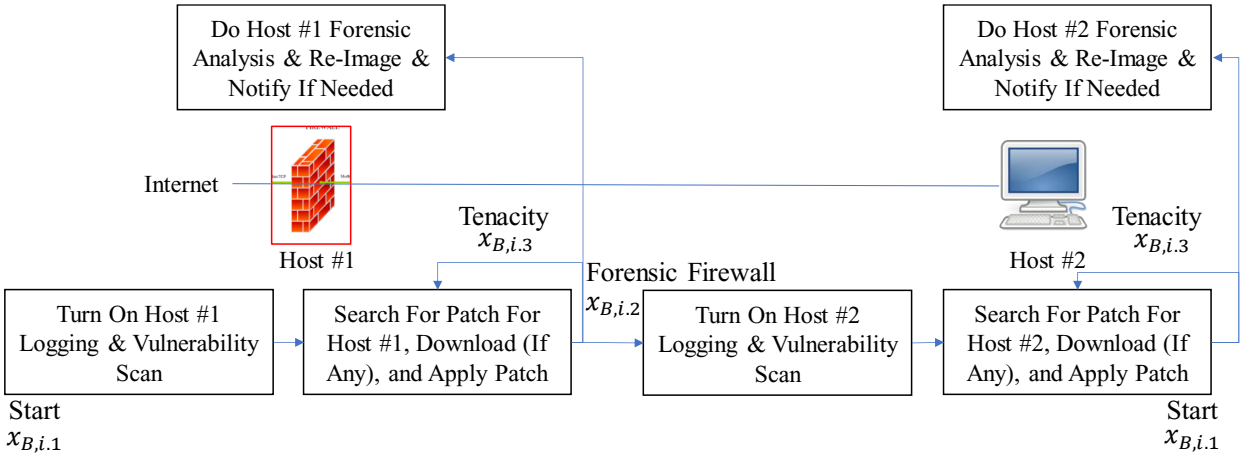


Table 2. Options as factor level combinations for the (a) red team and (b) blue team.

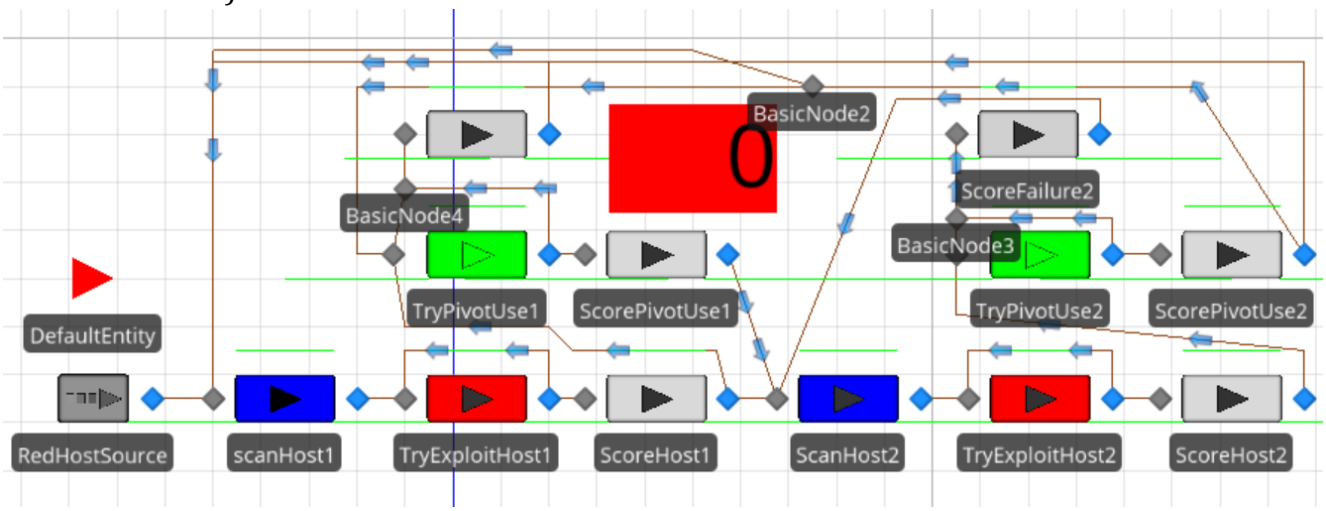
(a)

Option	Firewall-Pivot	Tenacity
1	First	Persist
2	Wait	Persist
3	Never	Persist
4	First	Move On
5	Wait	Move On
6	Never	Move On

(b)

Option	Start	Forensic Firewall	Tenacity
1	Firewall	Inspect Firewall	Persist
2	PC	Inspect Firewall	Persist
3	Firewall	Never	Persist
4	PC	Never	Persist
5	Firewall	Inspect Firewall	Move On
6	PC	Inspect Firewall	Move On
7	Firewall	Never	Move On
8	PC	Never	Move On

Figure 4. SIMIO model of the red team tasks which change the system states (along with blue tasks not shown).



The benefits of Procedure 1 and Procedure 2 for hypothetical players are clear. There are higher point values associated with activities for the inner hosts but determining sequence and when to give up is difficult. Yet, by selecting the levels indicated by the candidate Nash equilibria, the player likely maximizes their payoffs in point scores accounting for other players' selections.

The benefits for game designers are also clear. Point selections may be adjusted if the goal is to make deception and decision making important aspects of the game. After each iteration of point value changes by the game designer, Procedure 1 and Procedure 2 can be applied to generate the equilibria and balance the game (all equilibria have equal payoffs for both players). With the proposed experimental methods, the simulation times are reduced by a factor of two from the costs of a full factorial and the stopping condition times are greatly reduced (8.9 hours to 0.49 seconds). Therefore, approximate assurance is efficiently achieved such that the derived equilibria are actual Nash equilibria of the game irrespective of simulation replication errors.

Figure 5. Interaction plots showing predictions for scores: (a) red team and (b) blue team.

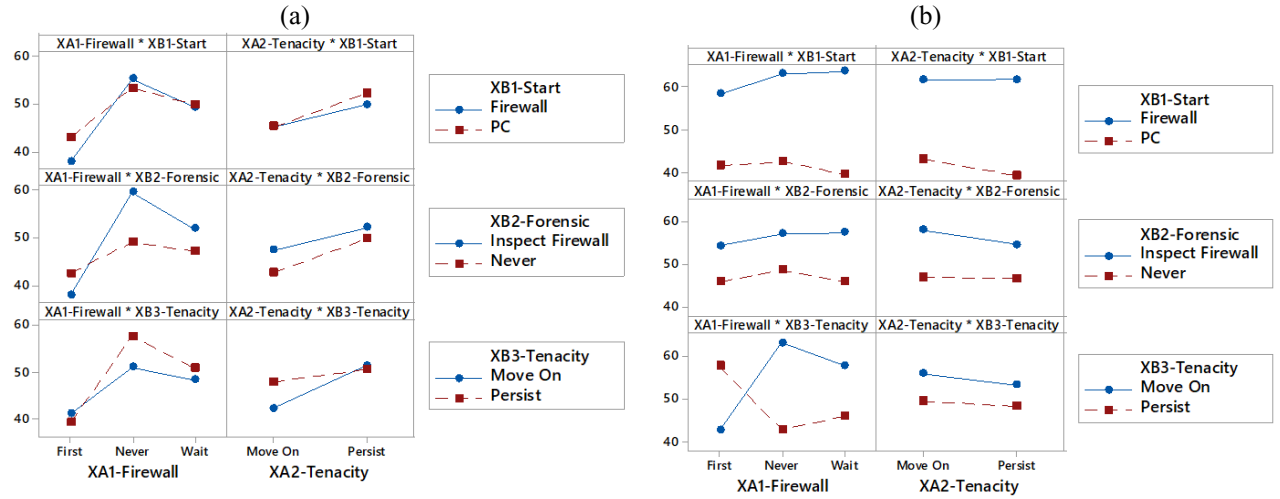


Table 3. Regression model summaries for (a) red team score and (b) blue team score.

(a)

Term	Coef	SE Coef	T-Value	P-Value
Constant	33.1	2.74	12.07	0
XA1-Firewall-Pivot				
Never	21	3.36	6.25	0
Wait	14.1	3.36	4.2	0.003
XA2-Tenacity				
Persist	6.7	2.74	2.44	0.04
XB1-Start				
PC	3.87	2.74	1.41	0.196
XB2-Forensic Firewall				
Never	3.4	2.74	1.24	0.25
XB3-Tenacity				
Persist	1.27	2.74	0.46	0.656
XA1-Firewall-Pivot*XB1-Start				
Never PC	-7.07	3.36	-2.11	0.068
Wait PC	-4.58	3.36	-1.36	0.21
XA1-Firewall-Pivot*XB2-Forensic Firewall				
Never-Never	-15.13	3.36	-4.5	0.002
Wait Never	-9.43	3.36	-2.81	0.023
XA1-Firewall-Pivot*XB3-Tenacity				
Never Persist	8.47	3.36	2.52	0.036
Wait Persist	4.48	3.36	1.33	0.219
XA2-Tenacity*XB1-Start				
Persist PC	2.42	2.74	0.88	0.404
XA2-Tenacity*XB2-Forensic Firewall				
Persist Never	2.45	2.74	0.89	0.398
XA2-Tenacity*XB3-Tenacity				
Persist-Persis	-6.38	2.74	-2.33	0.048

(b)

Term	Coef	SE Coef	T-Value	P-Value
Constant	56.38	3.65	15.43	0
XA1-Firewall-Pivot				
Never	22.5	4.48	5.03	0.001
Wait	20.28	4.48	4.53	0.002
XA2-Tenacity				
Persist	-2.15	3.65	-0.59	0.572
XB1-Start				
PC	-14.88	3.65	-4.07	0.004
XB2-Forensic Firewall				
Never	-10.16	3.65	-2.78	0.024
XB3-Tenacity				
Persist	14.48	3.65	3.96	0.004
XA1-Firewall-Pivot*XB1-Start				
Never PC	-3.62	4.48	-0.81	0.441
Wait PC	-7.2	4.48	-1.61	0.146
XA1-Firewall-Pivot*XB2-Forensic Firewall				
Never-Never	-0.08	4.48	-0.02	0.987
Wait Never	-2.9	4.48	-0.65	0.535
XA1-Firewall-Pivot*XB3-Tenacity				
Never Persist	-35.58	4.48	-7.95	0
Wait Persist	-27.1	4.48	-6.06	0
XA2-Tenacity*XB1-Start				
Persist PC	-4.1	3.65	-1.12	0.294
XA2-Tenacity*XB2-Forensic Firewall				
Persist Never	3.17	3.65	0.87	0.411
XA2-Tenacity*XB3-Tenacity				
Persist-Persis	1.4	3.65	0.38	0.712

Table 4. The predicted (mean) payoff matrices for scores: (a) red team and (b) blue team.

(a)

	1	2	3	4	5	6	7	8
1	34.68	40.97	40.53	46.82	39.80	46.08	45.65	51.93
2	53.26	54.97	49.68	51.39	53.90	55.61	50.33	52.03
3	64.16	63.37	54.88	54.09	**60.80**	60.01	51.53	50.73
4	34.37	38.23	37.77	41.63	33.10	36.97	36.50	40.37
5	52.94	52.23	46.92	46.21	47.20	46.49	41.18	40.47
6	63.84	60.63	52.12	48.91	54.10	50.89	42.38	39.17

(b)

	1	2	3	4	5	6	7	8
1	70.10	51.13	63.11	44.13	54.23	35.25	47.23	28.26
2	63.28	37.10	53.38	27.21	74.50	48.33	64.61	38.43
3	57.03	34.43	49.96	27.36	**76.73**	54.13	69.66	47.06
4	70.85	55.98	60.69	45.82	56.38	41.50	46.22	31.34
5	64.03	41.95	50.97	28.89	76.65	54.58	63.59	41.52
6	57.78	39.28	47.54	29.04	78.88	60.38	68.64	50.14

6. Conclusions and Future Work

In many real management situations, payoff matrices are not readily available, but the ability to experiment offline is. For example, the manager might have a simulation model with inputs from multiple decision makers or players. Also, there might be an ability to conduct relatively inexpensive test marketing experiments or sets of gaming exercises with a variety of stakeholders. These considerations have motivated a new class of experimental planning and analysis problems. We analyzed these problems and provided experimental plans for initial data collection, sequential methods for efficient follow-up experiments, and stopping rules, e.g., stop when all candidate Nash equilibria are likely true equilibria. Additionally, we characterized the finite sample and convergence properties of the proposed experimental procedures.

In our case study game application, we demonstrated the practical benefits of the proposed experimentation and analysis procedures. These procedures permitted the

estimation of the Nash equilibrium enumeration procedures at a fraction (half) of the offline experimental costs of full factorials. Reduction like this could be a critical enabler as simulations can take days to run and war gaming exercises with key stakeholders can be difficult to arrange, making test runs extremely expensive. Also, the provided formulas greatly reduced the computational burden of the associated probability estimation process (from 8.9 hours to less than one second). At the same time, the results here focus on single-period games with deterministic payoffs. Limited results apply to cases in which the payoffs may be intrinsically random or estimated using Gaussian stochastic regression (Theorem 1).

Therefore, many opportunities for further research exist. First, efficient stopping rules like the one in Theorem 1 can be developed for equilibria that more accurately predict human behavior than Nash equilibria, e.g., cumulative prospector theory. Second, the subject of offline experimental planning and analysis can be extended to address many other types of games, e.g., repeated, learning, and distributed games. Third, generalizing to more than two players can be explored together with the associated three-factor or higher order interactions.

Fourth, the use of equilibrium probability models for improving the efficiency of sequential experimentation procedures can be investigated. Fifth, issues about approximate and mixed equilibria (e.g., see Feder, Nazerzadeh, and Saberi 2007) and related support points (pure strategy points with positive probability) and empirical estimation-related offline supporting runs can be studied. Sixth, many applications additional to cyber CTF game design can be explored including test market design and efficient methods for testing military systems building on previous experimental design results (Johnson et al. 2012). Seventh, results can be generalized to address Cumulative prospect theory. Experiments can measure irrationality (in addition to reducing parametric uncertainty) as explored in Phade and Anantharam (2019). Finally, more advanced empirical modeling methods than linear models can be considered including multi-fidelity modeling (possibly addressing real and

offline experiments), multi-response, and Gaussian stochastic regression (Kleijnen and Mehdad 2014) methods can reduce total costs and further extend the practical relevance of game theoretic analyses and be related to relevant and irrelevant option combinations.

References

Antonova LV, Klyuchnikov MM, Loktionov AA, and Samsonovich AV (2018) Model of communication and coordination in a capture-the-flag paradigm. *Proc. Comp. Sci.*, 145, 72-76.

Aumann RJ (1987) Correlated equilibriumas an expression of Bayesian rationality. Econometrica 55(1):1–18.

Bolte, J, Gaubert S, Vigeral G (2014) Definable zero-sum stochastic games. *Math. Oper. Res.* 40(1): 171-191.

Breiman L, Friedman JH (1997) Predicting multivariate responses in multiple linear regression. *J. of the Roy. Stat. Soc.: Ser. B* (Stat. Method.), 59(1), 3-54.

Conitzer V, Sandholm T (2002) Complexity of mechanism design. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence* (pp. 103-110) Morgan Kaufmann Publishers Inc.

Conitzer V, Sandholm T (2008) New complexity results about Nash equilibria. *Games and Econ. Beh.* 63, 621–641.

Chapman AC, Leslie DS, Rogers A, Jennings, NR (2013) Convergent learning algorithms for unknown reward games. *SIAM J. on Control and Optim.* 51(4): 3154-3180.

Chen X, Deng X (2006) Settling the complexity of two-player Nash equilibrium. In *Proceeding of the 47th An. IEEE Symp. on Found. of Comp. Sci.* (FOCS).

Daskalakis C, Goldberg PW, Papadimitriou CH (2009) The complexity of computing a Nash equilibrium. *SIAM J. on Comp.*, 39(1), 195-259.

De Clippel G, Saran R, Serrano R (2018) Level-Mechanism Design. *The Review of Econ. Studies* (3):1207-27.

Delquié P (2012) Risk measures from risk-reducing experiments. *Decision Anal*. Jun;9(2): 96-102.

Foster DJ, Li Z, Lykouris T, Sridharan K, Tardos E (2016) Learning in games: Robustness of fast convergence. In *Adv. in Neural Info. Proc. Sys.* 2016: 4734-4742.

Feder T, Nazerzadeh H, Saberi A (2007) Approximating Nash equilibria using small-support strategies. *EC*, 7, 352-354.

Fruchter G. E (1999) The many-player advertising game. *Management Sci.* 45(11): 1609-1611.

Goos P, Jones B (2011) *Optimal design of experiments: a case study approach*. John Wiley & Sons.

Gorodetsky A, Marzouk Y (2016) Mercer kernels and integrated variance experimental design: Connections between Gaussian process regression and polynomial approximation. *SIAM/ASA Jour. on Uncer. Quant.* 4(1), 796-828.

Harsanyi JC (1967) Games with incomplete information played by "Bayesian" players, I–III Part I. The basic model. *Management Sci.* 14(3), pp.159-182.

Johnson RT, Hutto GT, Simpson, JR, Montgomery DC (2012) Designed experiments for the defense community. *Qual. Eng.*, 24(1), 60-79.

Jørgensen S, Zaccour G (1999) Equilibrium pricing and advertising strategies in a marketing channel. *J. Optim. Theory Appl*. 102(1): 111-125.

Keskin K (2016) Equilibrium notions for agents with Cumulative prospect theory preferences. *Decision Anal.* 13(3):192–208.

Kleijnen JP, Mehdad E (2014) Multivariate versus univariate Kriging metamodels for multi-response simulation models. *Eur. J. Oper. Res*. 236(2), 573-582.

Ko CW, Lee J, Queyranne M (1995) An exact algorithm for maximum entropy sampling. *Oper. Res.*, 43(4), 684-691.

Lee, K. H., Baldick, R. (2003). Solving three-player games by the matrix approach with application to an electric power market. *IEEE Trans. on Power Systems 18*(4), 1573-1580.

Lye K. W, Wing J. M (2005) Game strategies in network security. *Int. Sec.* 4(1-2): 71-86.

Mangasarian OL, Stone H (1964) Two-Person Nonzero-Sum Games and Quadratic Programming. *Journal of Math. Anal. and Appl.*, 348-355.

Meyer RK, Nachtsheim CJ (1995) The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics*, 37(1), 60-69.

Nash J (1951) Non-cooperative games. *Annals of Math.*, 286-295.

Phade SR, Anantharam V (2019) On the Geometry of Nash and Correlated Equilibria with Cumulative Prospect Theoretic Preferences. Decision Analysis. Apr 15;16(2):142-56.

Savani R, Bernhard von Stengel (2015) Game Theory Explorer – Software for the Applied Game Theorist. *Comp. Management Sci.* 12, 5-33.

Selten R, Chmura T (2008) Stationary concepts for experimental 2×2 games. *Amer. Econom.* Rev. 98(3):938–966.

Shubik M (1955) The uses of game theory in management science. *Management Sci.* 2(1), 40-54.

Shubik M (2002) Game theory and operations research: some musings 50 years later. *Operations Research*, 50(1), 192-196.

Solan E (1998) Discounted stochastic games. *Math. Oper. Res.* 23(4): 1010-1021.

Strom BE, Battaglia JA, Kemmerer MS, Kupersanin W, Miller DP, Wampler C, Whitley SM, Wolf RD (2017) Finding Cyber Threats with ATT&CK™-Based Analytics. MITRE Technical Report MTR170202. The MITRE Corporation.

Viscolani B (2012) Pure-strategy Nash equilibria in an advertising game with interference. *Eur. J. Oper. Res*. 216(3): 605-612.

von Neumann J, Morgenstern O (2007) *Theory of Games and Economic Behavior* (Princeton University Press, Princeton, NJ).

Appendix for online publication

A. The Proposed Cyber Capture the Flag (CTF) Game

CTF games divide into two types: Jeopardy style in which all participants attack a static network and red team/blue team exercises in which some participants attack, and others defend (Antonova et al. 2018). In this article, we propose a red team/blue team CTF game and a simulation model of that game. From our literature search, we believe that red team/blue team game designs are relatively rare. We differentiate in our terminology between actions (or equivalently tasks) and policy options (or factor level combinations).

Policy options are determined in an initial player meeting and govern action selection sequences within the game. Our expectation is that the game period is too short for policy changes as only a small number of actions are time feasible.

We believe that our proposed game offers multiple benefits including that it:

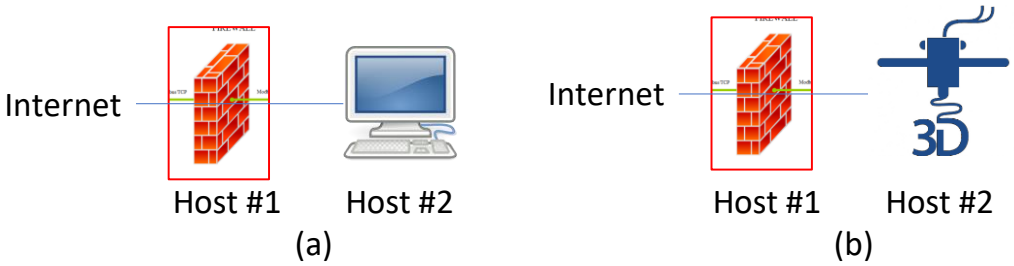
1. Supports relatively rapid training – Both the red team and the blue team learn three actions each of which requires only approximately one hour to study. Students can train for six hours (we estimate) and then play for three hours. At the end, they will have an understanding of scanning for IP address and **vulnerabilities** (e.g., bugs, weak passwords, or out-of-date encryption), **exploiting** vulnerabilities (e.g., applying an software script to gain access or cause mischief), **patching** vulnerabilities (i.e., applying code from vendors of the software to remove the vulnerability), **pivoting** to launch additional attacks (i.e., using the status gained from an exploit to score points such as launching more attacks), **exfiltration** (i.e., stealing data), escalating privileges (i.e., moving up to system administrator), and performing simple **forensic analysis** (i.e., trying to find evidence of intrusions). Of course, the student experiences will be limited and many attack options in the MITRE framework (Strom et al. 2017) are omitted.
2. Actions are relevant to real world cyber security professionals – The activities in the game are like those conducted currently by cyber security professionals and relate to multiple certifications.
3. Decision problems are relevant – Problems faced include pivoting options for the red team and starting options for the blue team. These choices can greatly affect the expected outcomes.

B. Game Description

Some Capture the Flag (CTF) games include the exploration of an extensive network over multiple days. For example, the MERIT game covers a virtual small town. Our game focuses

on a tiny network model in part so that all activities can reasonably occur within three hours. Figure 6 shows the network involving two hosts. We can imagine that one host is a firewall (Host #1) which is visible to the internet and the other (Host #2) is either: (a) a PC or (b) an advanced manufacturing equipment device such as a networked 3D printer. The specifics do matter in relation to which vulnerabilities and patches are relevant. Yet, for the purposes of our simulation model, the game activities are simply modeled by the associated mean service times.

Figure 6. Network model for game: (a) PC version, (b) equivalent manufacturing version.



There are many ways to define the cyber security state of a computer host which could be a personal computer (PC), server, printer, exercise machine, 3D printer, car, or cell phone. In the game system, there are four levels relating to the severity of the worst vulnerability on the host: low, medium, high, and critical. Recently, we have considered adding another state-based scheme including the presence of so-called "celebrity" vulnerabilities such as "Heartbleed," a bug famous enough to have its own logo (♡). Here, we consider only two types of hosts, i.e., hosts whose worst vulnerability achieves a medium on the CVSS scale and those whose worst vulnerability achieves a critical score for simplicity. Critical vulnerabilities are often so problematic that they can be seen externally to the organization and exploits are widely published. Then, hackers may gain full or near full access to the host almost as easily as by logging in with a known password.

Hosts can also be compromised in the sense that unauthorized personnel can have partial or full access. Therefore, we consider hosts in four states as indicated in Table 5. If a host has a critical vulnerability on it, it is easier to quickly and completely compromise it. If it is already compromised, it can be of use to hackers who can "pivot" to attack other hosts or exfiltrate data. Blue team personnel naturally seek to identify whether hosts are compromised and transform them into not-compromised hosts which have as many of their vulnerabilities patched as possible. Yet, of course, patching and forensic analyses take time as does compromising hosts through manually applied exploits.

Table 5. Four host states relating to compromise and vulnerability status.

Host State	Compromise State	Vulnerability Status
1	Not Compromised	Critical and Medium Vulnerabilities
2	Not Compromised	Medium Vulnerabilities Only
3	Compromised	Critical and Medium Vulnerabilities
4	Compromised	Medium Vulnerabilities Only

We imagine that the multiple members of both red teams and blue teams will follow the same workflow by agreement rather than branching out individually, a choice that might seem sensible given the limited amount of training. More complicated networks and independent team members can be considered in future work.

Red Team Actions

In our proposed game under development, we consider three red team actions:

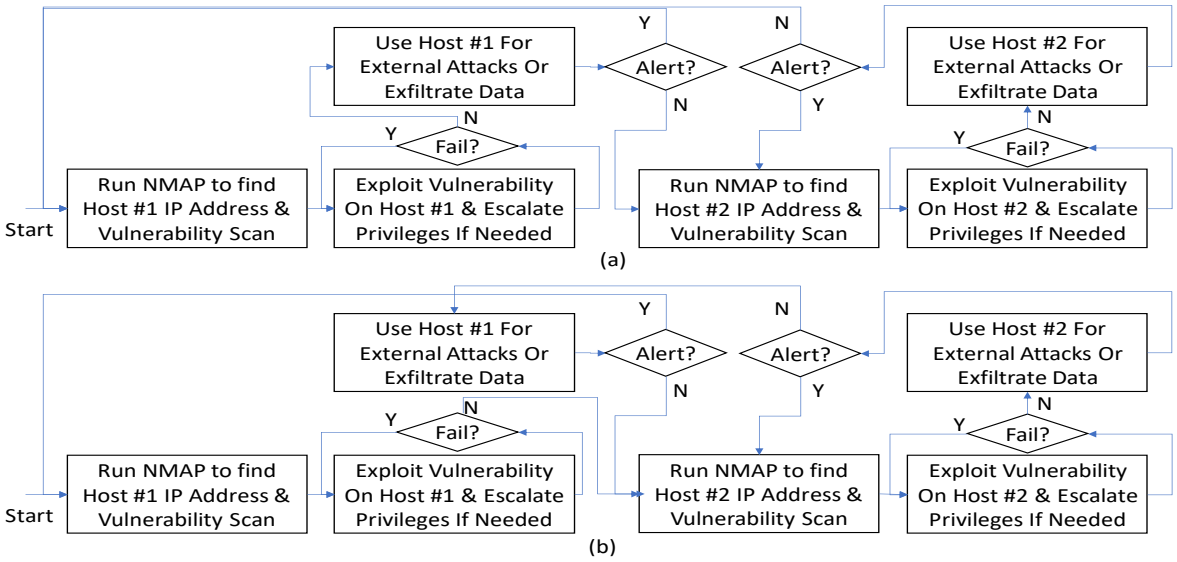
1. **Run script to find visible host IP address(es) & vulnerability scan** – On virtually any computer you can scan to find the visible IP addresses, e.g., using a port scan program (e.g., NMAP). With an address you can scan the host for vulnerabilities, e.g., using the Nessus scanner from OpenVAS, Tenable Security, or the Rapid7 scanner.
2. **Exploit vulnerability on host & escalate privileges if needed** – After identifying the vulnerabilities present on the target host -- generally these are so-called "network" vulnerabilities since they are in part visible without full access -- you can look for

associated available exploits. A product such as Metasploit facilitates finding the exploits and launching them. Depending on the level of the vulnerability and the quality of the exploit, one might not gain sufficient access to use the host. A privilege escalation activity may then be possible to gain additional access.

3. **Use host for external attacks or exfiltrate data** – Once the host is compromised and privileges have been achieved, the host is ready for use, i.e., to pivot. Pivoting to external attacks (actually they are internal hosts in our game) is possible but risky in that the intrusion detection system or firewall rules might identify the compromise and block access. In fact, once access is blocked, personnel can easily isolate that host. A less risky step might be to exfiltrate or steal the information already on the host. Of course, most hosts do not contain monetizable data (e.g., medical records or possibly credit card data).

Figure 7 shows a workflow that connects the red team actions. In the greedy version (Figure 7a), the red team immediately attempts to use the firewall for gain. In Figure 7b, the red team is patient. Note that the workflow implies that the red team can use a host with either external attack or exfiltration but not both. Also, game rules dictate that the red team must attempt a major activity before returning to reuse a compromised host. By "alerts" we mean declarations that hosts are compromised that limit direct access to attackers.

Figure 7. Red team work flow: (a) Greedy version, (b) Patient version.



Blue Team Actions

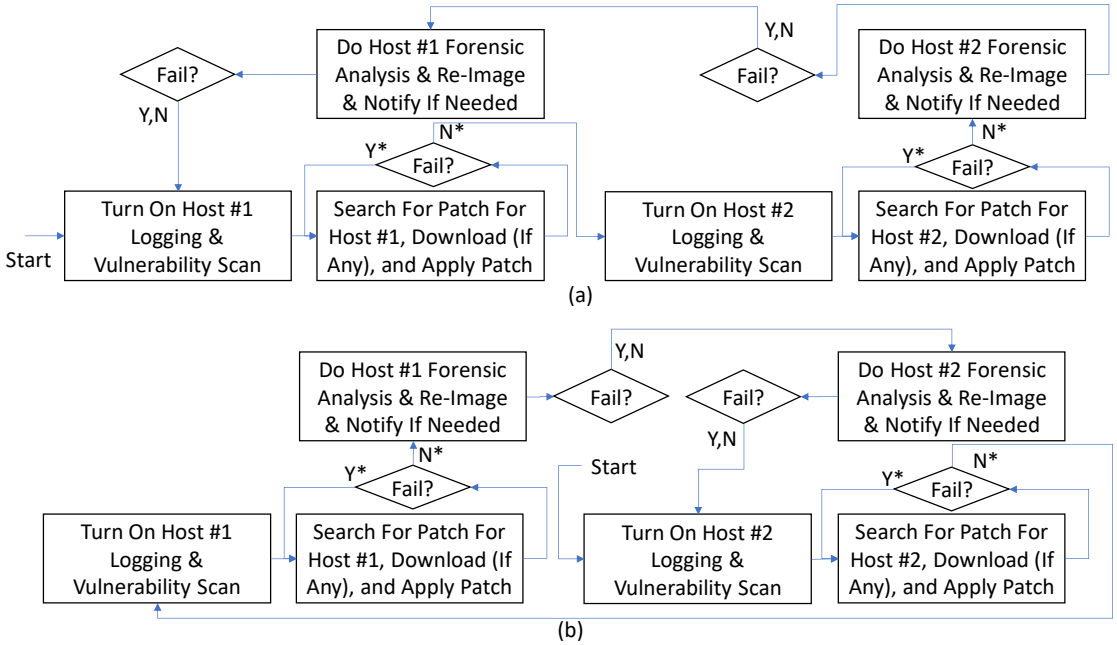
Similarly, we consider three blue team (compound) actions:

1. **Turn on host logging & vulnerability scan** – There are many logging options to record which hosts authorized or unauthorized are doing during their sessions. Enabling basic logging, e.g., through the Windows menus, can reasonably preserve privacy (sometimes) while facilitating effective forensic analyses of compromise. Also, the blue team needs to scan for vulnerabilities in a manner similar to that used by the red team.
2. **Search, download (if any), and apply patch (if any)** – Once vulnerabilities have been identified, there are often recommended patching or remediation actions provided by the scanner. Still, sometimes the security personnel must search the internet for patches and related information. Sometimes patches must be downloaded manually and applied, often only after successfully demonstrating that they do not interfere with needed software and services.

3. **Do forensic analysis & re-image & notify if needed** – Even during a CTF game with only two hosts, it is not clear at any given time whether a given host is compromised. Also, log analyses are supported by many software programs, but the process can be time consuming. It might also fail to find compromised hosts. If a host is found to be compromised, there will often be legal implications. Therefore, notification of affected individuals is likely to be legally required.

Figure 8 shows two workflows for blue team members. One option is to start with the firewall like the red team, hoping to patch it before the red team exploits it. Alternatively, they can start on the PC or 3D printer which is associated with many more points or payoffs in the game. They can try to patch the worst vulnerability on that host to make it more difficult and time consuming for the red team. Note that the red team must start on the left-hand-side because the firewall is the only host visible to the internet.

Figure 8. Blue team workflow: (a) Start on the left and counterclockwise, (b) Opposite.



Our scoring system assigns points to red and blue teams based on the achievements of host compromises (minor) and host uses (major) with an emphasis on success with the internal host (#2). The blue team scores mirror the red team scores with emphasis on successful forensic work on the internal host.

C. Input Analysis

For time estimates for the tasks, we use YouTube videos illustrating the actions in applications. By selecting vulnerabilities carefully, we believe that we can control the service time distributions to some extent. Also, we believe that cyber security activity times are associated with a high coefficient of variation such that the exponential distribution might reasonably apply.

For one vulnerability for which codes are already loaded and available, exploitation might be quick. For another, searching, research, downloads, and testing might require considerable time in at least some instances. Also, players might not be aware that the exploits that they need are preloaded in their Metasploit software, for example. Table 6

shows the parameters in our simulation model and their descriptions. It also shows the estimated mean values in minutes and the videos used for these estimates. Note that these videos describe the actions by team members with realistic example illustrations.

SIMIO Model

SIMIO is a commercially available software product with general simulation capabilities. It supports three-dimensional animation including objects from the Google library. It also offers many experimental design options and associated visualizations. In our simulation model (see Figure 4), we use multiple features including some for convenience (e.g., "dummy" hosts in which no task is performed):

Paths –Paths and user-defined properties are used to regulate the chances that activities are successful. With the user-defined variables, the model parameters or properties are accessible in the spreadsheet associated with the simulation experiments as well as the:

States – States are used to store the values associated with the hosts. Then, using the Math.If() formula construction, the properties and the states can set the service times and the success probabilities.

Dummy servers – By using "dummy" servers such as ScoreHost1, the server features permit assignment conditions for the properties and states. This allows the scores of both teams to be updated as well as the states of the hosts. If actions fail, paths route the attention away from the dummy servers such that the scores and system states remain unchanged.

Duplication for red and blue teams – Even though the two teams work on the same two hosts, they are likely in different rooms and not aware of each other. Also, their service times, success probabilities, and attention paths differ greatly. Therefore, we developed three copies of the network which mirror the red team and two distinct blue team workflows. The blue team has an additional factor and flow because its operations can start either on the firewall (Host #1) or on the internal host (Host #2). This relates to having full internal access.

Table 6. Mean time estimates (in minutes) and probability estimates with supporting YouTube videos used to ballpark initial values for the simulation.

Parameter (Property)	Description	Mean	Supporting Video or Notes (If Any)
HRserv1	Exploiting host with critical vuln.	15	https://www.youtube.com/watch?v=ZT7VYsJvh2Q
HRserv2	Exploiting with medium & escalation.	30	https://www.youtube.com/watch?v=RdnVC0kNxN4
HRserv3	Entering compromised host.	2	Similar to a usual login.
HRserv4	Entering compromised host.	2	Similar to a usual login.
HRharvC	Pivoting to third party attack	20	https://www.youtube.com/watch?v=qIEHUUt2Wfc
HscanTime	Mapping and vulnerability scanning	10	https://www.youtube.com/watch?v=hMKIIRhfk74, https://www.youtube.com/watch?v=9LA3iQfGGLY
HRprob1&2	Chance exploit works.	0.5	Exploits can fail.
HRharvNoC	Discovering access is lost.	5	Attempted logins and failure.
lowCompScore	Game score parameter.	0	Chosen by the game designer.
lowHarvScore	Game score parameter.	10	Chosen by the game designer.
highCompScore	Game score parameter.	5	Chosen by the game designer.
highHarvScore	Game score parameter.	25	Chosen by the game designer.
HBserv1	Enabling logs & vulnerability scanning.	15	https://www.youtube.com/watch?v=hTK0pywfmDE
HBserv2	Enabling logs & vulnerability scanning.	5	Fewer vulnerabilities and pre-scanned.
HBserv3	Enabling logs & vulnerability scanning.	15	https://www.youtube.com/watch?v=hTK0pywfmDE
HBserv4	Enabling logs & vulnerability scanning.	5	Fewer vulnerabilities and pre-scanned.
HBprob1	Patching critical vulnerabilities.	0.9	Likely patches are available because of rating.
HBprob2	Patching non-critical vulnerabilities.	0.5	Likely patches are not available because of rating.
HBprob3	Patching critical vulnerabilities.	0.9	Likely patches are available because of rating.
HBprob4	Patching non-critical vulnerabilities.	0.5	Likely patches are not available because of rating.
HBlogC	Forensic inspection, reimage, & notify.	45	https://www.youtube.com/results?search_query=inspect+host+logs+for+cyber+security+compromise
HBlogNoC	Forensic inspection	30	See similar
HBlogPC	Chance inspection finds compromise.	0.9	Chosen by the game designer.
HBlogPnoC	Chance inspection finds compromise.	0.9	Chosen by the game designer.
LimitBRight		0	Chosen by the game designer.
lowIndicentRepScore	Game score parameter.	20	Chosen by the game designer.
highIncidentRepScore	Game score parameter.	30	Chosen by the game designer.