Toward Realistic Modeling Criteria of Games in Internet Security

By Jonathan M. Spring

There have been various attempts to apply game theory to various aspects of security situations. This paper is particularly interested in security as relates to computers and the Internet. While there have been varying levels of success in describing different aspects of security in game-theoretic terms, there has been little success in describing the problem on a large scale that would be appropriate for making decisions about enterprise or Internet security policy decisions. This report attempts to provide such a description.

We propose that there are three types of players in the game: the computer user, the malicious actor, and the security architect. This paper is not about how to "win" the game of Internet security or a prescription of the clever strategy — as game theorists make clear, "the search for effective decisions is not a central problem of game theory" [29]. The aim of this paper is two-fold, one for theorists and one for practitioners. For game theorists, this paper provides a more accurate description of the actual dynamics of security-related interactions on the Internet. For practitioners, we will provide a framework to clarify existing motivations and intuitions about the current situation and why it is, or is not, working. Hopefully this perspective on the dynamics of the situation will enable more effective decisions and guide the search for clever solutions using other fields of study.

This paper does not focus on building mathematical tools for analysis. We focus on the description of the game. The three players — user, rogue, and architect — all have competing interests. The main interactions are thus: (1) The user and architect negotiate a suitable system configuration which includes trade-offs between productivity (of the user), security (architect's goal), and cost; this is a non-zero sum game. This occurs on a much slower time scale than the other two interactions. (2) The rogues attempt to steal resources from the user; this feature is also not a zero-sum game, and so presents some interesting challenges. (3) The third interaction is between the architects and the rogues. Although these two parties are defined as diametrically opposed, their interaction is also not zero-sum.

With these interactions laid out, we make the following important observation about the game itself: the user can ignore, or even be complicit with, the rogue without immediate loss. This fact makes it harder to convince the user to work with the architect to improve security. There are other interesting points to consider related to the game: (1) The game is modeled with three players, and we assert that at least this many players is necessary to maintain fidelity with the real Internet; (2) perfect security cannot be promised, even in principle, because the features of the game are such that there is no guaranteed method to compute a globally-optimal strategy (three player game, the fact that it is non-zero-sum, and the fact that there is imperfect information).

1 Introduction

Game theory was founded as a sub-discipline of mathematics in the mid-20th century. It is a description of how rational decision makers compete. However, this paper is not about how to "win" the game of Internet security or a prescription of the clever strategy — as game theorists make clear, "the search for effective decisions is not a central problem of game theory" [29]. What game theory can illuminate is how an interaction proceeds, certain rules about the outcome given the inputs, and to help an analyst clarify a situation by reducing a complex situation to a more compact description.

For the purposes of this paper, we will assume the payoffs to the players are already defined. How to do this is non-obvious. However, a process such as the model described in

[35] provides a plausible method for arriving at the payoffs, measured in monetary resources lost or gained.

Game theory assumes we have rational decision makers. Kahneman's psychological work, and the resulting behavioral economics literature, demonstrate that people are not purely rational. This has important ramifications for actually selecting policies that will be effective, however from our abstract point of view it just means we might have to adjust our payoff values to account for the fact that people may value something more or less than is rational. As such, we will leave this issue aside for now.

When describing the game, we will describe the payoff matrices to the extent possible — which values are positive or negative, their relative magnitudes, etc. However, our goal is not to formulate games to the level of detail that analytic or numeric solutions are possible. There is still much work to be done before that can be achieved. The goal of this paper is to provide the shape of a game as it relates to information security on the Internet.

2 Related work

Game theory was kicked off in 1944 as a robust field by [37] and saw application to such national security issues as nuclear deterrence and mutually assured destruction. The essential problems of bargaining and non-cooperative games were laid out by John Nash in the early 1950s [27, 26]. Founded as a branch of mathematics, after the theory acquired conceptual foundations (see [30, 29] for a summary), notions from game theory spread to a number of fields, notably economics (for example [31]). Some game theorists have also taken influence from other fields, such as evolution and dynamical systems [15]. Some game theory texts are broad, mathematical treatments such as [28]. Useful for the work described in this paper are treatments of non-cooperative games and games of incomplete information, which is included in some of the above but focused in some texts such as [14, 25].

There have been previous efforts to extend game theory into the field of information security; [34] summarize and categorize the efforts. Game-theoretic models have been proposed for both organization-scale [7] and single-wireless-node-scale [40] information security games; both as single-play [36] and repeated games [20]. As economics intersects game theory it also intersects information security; for a summary of the extensive work on the economics of information security, see [5].

We heuristically derive our model from case studies and empirical reporting of information-security relevant behavior on the Internet. There are several organizations that report on various aspects of cyber-crime and human behavior, in varying levels of detail, such as [4, 33, 2, 23, 13, 19, 24, 9, 8, 1]. These sources do not generally attempt to derive a general model from the information observed. There is some work in cyber-crime and risk dynamics such as [23, 35] that model criminal behavior, which inform our game theoretic modeling directly.

It seems that all existing applications of game theory to information security force the game to be a two-player game. Some study population dynamics of users and adversaries [39], which has richer descriptive power, but these retain still only two types of players. These efforts do not seem satisfactory in describing the Internet-scale phenomenon of information security, as reported by the economics, cyber-crime and dynamics literatures. We assert that a primary reason for this shortcoming is that the game cannot be described with fewer than three players.

3 Theory

The following subsections describe a more adequate treatment of the modern Internet security landscape. First, we describe the players; secondly, how they interact informally; finally, a more formal definition of the interaction.

3.1 The players

We shall define three classes of players. Granted, these classes may be subdivided for certain purposes, but we shall treat them as the essential units for our purpose of providing an accurate and useful model of the security interactions on the Internet. A single person or machine may change roles during its lifetime, and the ability to do so presents practical challenges, however we shall treat the three classes of players as describing mutually exclusive and exhaustive roles. The first step is to describe these players, their goals, and their capabilities.

User is an agent who utilizes a computer system. By definition, they have not designed the system they are using.[1] The user may have access to a limited number of configuration options provided by the architect of the system. The main goal of the user is to produce some product of value, using the computer

1 An agent may both use one system and be the architect of another; most software developers fit this description. However the roles of user and architect qua roles do not overlap.

system as a tool to that end. Possible products span the range of human ingenuity. An important consequence is that the Internet as we are analyzing it is not a closed system, it is a tool of the larger human economy. This is a factor in the assessment that games involving users are not zero-sum.

Architect is the agent that has designed a computer system or the policy under which the system operates. This may be operating system developers, enterprise security policy designers, or the IETF; there is a wide range of systems and they all have architects. Architects can also be identified with the owner and administrative operator of a system, especially in the case of enterprise organizations. The architect is who selects and enforces security policies.

Architects, as a group, are the hardest to unify as one label. Members of this group are highly specialized and fractal. Since no organization builds all of its own software, every architect is also the user of other systems. However, the essential element is not what role a particular person has. The key fact is that every system has an architect or architects that have designed it. The Internet is not a natural phenomenon, and so while it is bound by some physical laws the key feature is that every system that operates on the Internet has an architect who made decisions about that system, its capabilities, and so on. In the general case, the architect's goal is for their system to be used by users. A part of this goal is making it secure enough to be used, however it would be naive to say that an architect's primary goal was a secure system. If this were the only goal, the systems could all be turned off and encased in concrete to accomplish the goal. To specify what it means "to be usable" the architect specifies aims in reference to what users need to accomplish user's goals.

Rogue is the attacker. The definition of an attack can be disputed, but we shall mean attack as defined by Howard and Longstaff: "a series of steps taken by an attacker to achieve an unauthorized result; ... among these steps is an action directed at a target (an event), as well as the use of some tool to exploit a vulnerability. Second, an attack is intended to achieve an unauthorized result as viewed from the perspective of the owner or administrator of the system involved" [33]. Considering a system as large as the Internet, and given global political disagreements, it would be naive to think that we could agree on one rogue or set of rogues. An entity that is a user according to one point of view may be a rogue according to another point of view, and we may not be able to say that either point of view is correct. However, each user will experience a rogue that perpetrates attacks. The scope or goal of these attacks may vary; money, fame, chaos, or national interest may all be motivators for different rogues in varying degrees.

We assert not only that these are three players in a game describing Internet security, but that these are the only three types of players. Further, that all three must be modeled if a description of Internet security is to be accurate. For discussion of modeling more than three players at once, see Subsection 3.2.4.

3.1.1 Realizations

There will be multiple realizations of this game occurring in the world simultaneously. There are more than three agents on the Internet at any given time, so the above is a simplification. From different points of view different agents will be considered to be in different rolls, whether user, architect, or rogue. The fact that agents can change roles certainly can have real-world impacts. For example, the NSA's involvement in the design and architectural review of DES can be seen from many points of view [18]. The historical claim was that NSA may have negotiated down the key size because it wanted to be able to attack the protocol more easily in its role as rogue. However, it was permitted at the negotiating table in the first place because it was going to be a legitimate user of the protocol as well.

That one organization may have competing goals does not break the user-architect-rogue model. Various realizations of the model in the real world will alter the agents playing the game, their rolls, and their payoffs. However, the changes do not affect the general description of the game. This is one benefit of describing the game at the proposed level of abstraction. Although agents can serve multiple roles simultaneously in the real world, to simplify modeling at this early stage we put that aside and focus on the goals of the agents in each of the three rolls and the essential features of the interactions between a user, architect, and rogue.

3.2 Features of the Game

It is not possible to separate the three players from each other. The observation that there are three distinct, essential roles in the game is a vital observation. The other attempts (see Section 2) at bringing game theory into security have focused solely on two-player games. We believe this to be a primary factor in why these attempts have had unsatisfactory applicability to actual security decisions.

A three-player game results in a three-dimensional strategy space. Such a space can be difficult to conceptualize. In order to introduce the dynamics more gradually, we describe the essential features of each two-way interaction separately before combining this into the holistic problem.

One feature common to all three interactions is that they are not games of perfect information. The user does not know everything the architect has done, and vice versa, and likewise with the rogue. This is true in principle for the rogue, but it is also true in practice for the user-architect interaction. Security best practice such as least privilege and least access, legal standards of privacy, technical limitations on data processing, and the use of closed-source programs all make imperfect information a practical reality that is ingrained in the day-to-day use of the Internet. In fact, one plausible negotiating point is how much visibility the architect has into what the user is doing, and so on. This means the game is not guaranteed to have a globally-optimal strategy, as only games of perfect information are guaranteed to have one. The gist of the interactions of the players is summarized in Figure 1.

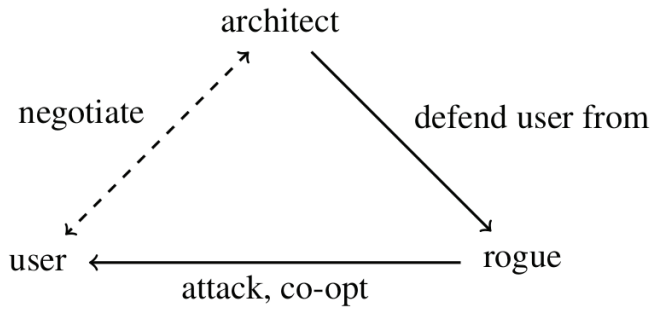

Illustration 1: Representation of the three distinct types of players in the proposed game. Dashed line indicates a game that is repeated at a slower pace. Labels on the edges describe the intent of the interaction. Although the arrows indicate that, for example, "architects defend user from rogues", this interaction is not independent of the others; all three players play simultaneously.

3.2.1 user-architect

The user and the architect are negotiating features of the system being used. Either side may be advocating for adding, removing, or modifying features of the system. Security requirements and rules can easily be viewed in this space. The simplicity of the statement hides a degree of difficulty in game-theoretic terms, however negotiation games have been reasonably well studied [6].

This part of the game is non-zero-sum; the user and architect can clearly come to agreements which are better for both of them. While the interaction is not antagonistic, it is not truly cooperative, either. The user and architect have different, ostensibly unrelated goals. So we should expect the user and architect to cooperate only insofar as it is mutually beneficial based on the payoffs provided to each.

One common aspect of both game theory and economics is the idea of discounting future payoffs in a repeated scenario. Colloquially, this is captured by "a bird in the hand is worth two in the bush." Rational decision-makers will value an equal payoff now rather than later if they have an expectation the game will end or change before that future payoff [31]. This is precisely the scenario we are building here, as the payoffs will be renegotiated at future points. Discounting is a reasonably well-studied feature in game theory. One important aspect that is practically important is that different entities can have different discounting rates; that is, entities are not equally patient [31]. To model the dynamics, a valuation of the initial capital of the parties is also necessary, which would have to take into account physical and information assets.

Not only does this user-architect interaction involve variable discounting rates by the parties, but the payoffs going forwards are also a function to some extent of investments made by players in the past. For example, if the user wants a capability in a system that does not exist, the architect will have to build that capability over a period of time. This requires resource investment before the benefits of the capabilities can be realized. Game theorist have studied games in which the players' past actions affect future payoffs, especially in the context of financial investment [16]. Although multiple investments could be modeled, in Section 3.3 the investment that is modeled is the infrastructure controlled by the rogue, which the user and architect have an interest in minimizing and the rogue wants to maximize.

One interesting characteristic of the user-architect interaction is that what the two parties are negotiating boils down to the payoffs for the parties in the user-rogue and architect-rogue interactions. Realistically, every several months system configurations could be renegotiated, however the other two interactions occur on second-to-second time scales. Conceptually, the user and architect negotiate payoffs in a repeated game every so many plays of the game. How often renegotiation happens would also probably be a feature of the negotiation. Whether this can be modeled as a situation in which the user and architect usually only have the option to "change nothing" at most stages of the game is not known. In principle, this could

be done without loss of generality or specificity. In practice that approach seems unrealistic — the user and architect do not check in every few seconds to confirm "change nothing" — but it may be a feasible model.

However it is modeled, the players are assumed to be bound by the terms negotiated for some number of repetitions of the faster games. Another possible option for modeling the problem could be borrowed from multiscale mathematics [38]. Unfortunately, we are unaware of any applications of multiscale mathematics to game theory at this time.

3.2.2 user-rogue

Conceptually, the interaction between the user and the rogue is one in which the rogue is attempting to steal the user's resources. Since this is theft, it would appear on the face of it to be a zero-sum game. However, we do not believe this to be the case. The rogues are not necessarily stealing purely rivalrous goods. If the resources stolen are non-rivalrous, then the user is not inconvenienced by the rogue's usage, and so the game is non-zero-sum. Money is rivalrous, but money is not the only resource the rogues steal. Rogues can also steal computer resources or information.

For an example of rivalrous and non-rivalrous goods, consider a sweater. If it is cold out, I like to wear a sweater. If you steal my sweater, I cannot wear it and I will be cold. Sweaters are a rivalrous good. Stealing my sweater would be a zero-sum game, because one's loss is precisely the other's gain. Now consider Pythagoras's theorem concerning the lengths of sides of a triangle. If a teacher knows it, and teaches the students that $a^2 + b^2 = c^2$ the teacher is not excluded from using that information. It is not as if the teacher gave out 20 sweaters. The usage of the theorem does not prevent others from also using it. Theorems, and information items generally, are non-rivalrous.

Internet access and computer processing cycles are not precisely the same as information in this regard, but they are more alike to non-rivalrous goods than they are like sweaters. If a user is only consuming 10% of available Internet bandwidth because, perhaps, they are asleep or out of the house most of the time, then a rogue with control of the computer can use the rest of the bandwidth without inconveniencing the user. Likewise with processor cycles and disk storage space. Precisely how the rogue must act in order to achieve this goal may require some technical cleverness, however here we are interested in specifying the nature of the game, not clever ways to attack or protect systems.

Information (and computer resources) can be given a monetary value. Information is often given monetary value in intellectual property rules and debates, for example. Yet the same information may easily have different value to different parties. Thus while we may reasonably expect to value the resources in the game we are describing, the game will be non-zero sum not just because the goods in question are non-rivalrous but also because the different players value the resources differently. For example, even if Eve gains something that Alice loses, if Eve considers it to be worth 1 unit, yet Alice valued having it at 2 units, the transfer is non-zero sum.

The payoffs for this repeated game accrue on a relatively short time scale compared to the user-architect interaction, as noted above. Also similar to as noted above, the rogues likely have different discounting rates than either the users or the architects for the repeated aspect of the game. Given the illicit nature of the rogues' activity, it is plausible that the rogues are the least patient.

Prior actions will have an effect on future payoffs in this interaction. As the rogue compromises more user resources, the rogues can use those resources to compromise further user machines. These invested resources also play in to the architect-rogue interaction because rogues can use these compromised resources to evade the architects. Therefore, actions taken in this plane of the game directly affect others, just as in the user-architect interaction.

3.2.3 architect-rogue

This plane of the game describes the interaction between those who design and own the systems and those who are attempting to subvert those systems. It is the more difficult plane to characterize intuitively. Since all of this occurs on computer technology, the rogue can directly attack the systems the architect is using to protect the users. However, in order to maintain clarity, the rogue is not attacking the architect directly; they are attempting to subvert the protections that the architect has in place to protect the user. This includes aspects such as email filtering, anti-virus signatures or other host-based protection systems, firewall rules, intrusion detection system (IDS) signatures, etc. The architect can generally reconfigure these rules at machine speed, and the rogue can likewise employ countermeasures quickly, and so this interaction's timescale is approximately the same order of magnitude as the user-rogue interaction.

This game is clearly non-cooperative and is not a game of perfect information. Both parties are intentionally obscuring

their methods from the other. Even though the two players are directly competing to block or use resources, we posit the game is non-zero-sum. The architect does not directly lose resources if the rogue can successfully steal resources from the user, nor does the architect directly gain resources if the rogue cannot use them. In fact, the architect will generally have to expend resources to block the rogue.

Just as the rogue can build up resources in the user-rogue interaction as a kind of investment in future payoffs, the architect-rogue interaction can effect that investment. In this case, the effect is that the architect can reduce or block the resources available to the rogue. This is precisely the same set of resources that the rogue is building up in the interaction with the user; it seems reasonable that these invested resources could be modeled together, as described in Subsection 3.3.

3.2.4 More than three players

To this point, we have taken for granted the simplifying assumption that there are only three agents — a user, an architect, and a rogue. In order to model more agents, the modeler could consider each of these groups a coalition of agents. Then there is a coalition of users, a coalition of architects, and a coalition of rogues. Individual agents may switch coalitions if it is in their interest to do so and they are permitted to do so by the other members of the coalitions.

This coalition model would be able to account for some of the complexities of the modern Internet. No architect would wittingly negotiate their system settings with an adversary. However, if the users are a coalition of users, an agent may be able to enter the coalition of users and corrupt the negotiation process and then enter the coalition describing rogues to attack the system.

Admitting coalitions and arbitrarily many agents complicates models, and these added complexities would cloud the initial formulation we are pursuing here. For the time being, we will resume the simplifying assumption of three agents in order to describe the game requirements satisfactorily at this stage of development. Generalization to coalitions of users, architects, and rogues should be possible in future work as necessary to improve the expressiveness of the model.

3.3 Formalizing the game

The description provided in Subsection 3.2 now permits us to describe the game more formally. Conventions for mathematical representations are drawn from [29], which also contains an accessible explanation of the symbols.

3.3.1 Initial definition

First, we can posit that there will be payoffs accrued to each of the three players. We may represent these as *a* for architect, *u* for user, and *r* for rogue.

Secondly, there are some general state variables that will be held across the game. Namely, the vector $\vec{v}$ and the matrix S:

$\vec{v}$: the infrastructure available to the rogue

S : the payoff matrix that will be used for the fast-scale games, based on system configurations the user and architect negotiate

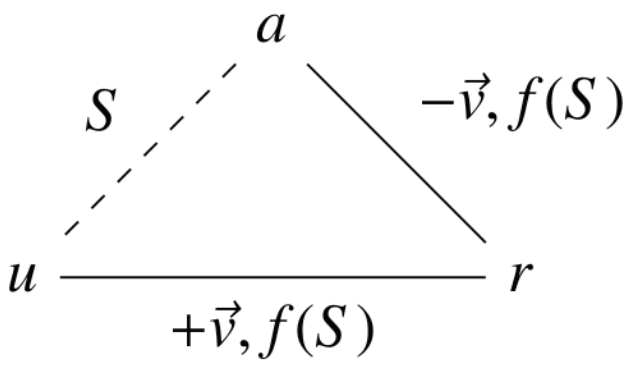

Illustration 2: Representation of the three distinct types of players in the proposed game. Dashed line indicates a game that is repeated at a slower pace. Labels on the edges denote values that are involved in each interaction; either the values are adjusted by the result of the game or the result of the game is a function of that value. The sign of the effect is noted if it is constant.

The rogue's infrastructure is really two things, hardware and software. Hardware are physical computing resources. This includes various incommensurate computing features, such as network bandwidth, processing power, stable disk storage, and volatile memory (e.g., RAM). In general, in both benign and malicious cases software is used to control this hardware. However, in considering rogue malicious infrastructure, software means software which has been developed in order to attack computers and wrest control from their rightful owners. In this sense, software does not mean how many machines each copy of software is deployed upon, but rather the total engineering effort the rogue has at their disposal.

One might expect a set of two vectors, one for hardware and one for software; this is not what is proposed. Both elements of malicious infrastructure are captured by $\vec{v}$ in different ways.

Modeling hardware is relatively straightforward. Hardware is the set of all machines in question, and a continuous value from 0 to 1 as to how much of the machine's operations the rogue controls. Being able to unplug the machine counts as control, so they do not likely have full control over many machines. We can thus define $\vec{v}$ more rigorously as:

(1) $\vec{v}$: = $[v_1, v_2, ..., v_q]$; $0 \leq v_\kappa \leq 1$

Where q is the number of computers in use. The operations of a specific computer κ are controlled to the extent v_κ by the rogue, where 0 is not at all and 1 is completely. Thus v_κ is a value for the percentage of the computer's resources that are controlled by the rogue.

How to model malicious software infrastructure is less clear. Software is a more complex set of capabilities the rogue has at their disposal. The architect is likewise constantly developing new software, patching vulnerabilities, and so on. Some architect activities, including detecting malicious software and patching vulnerabilities, do reduce the software infrastructure available to the rogue because these activities make certain malicious software ineffective against the rogue's targets.

One might imagine defining a matrix of the software available to the rogue and its effectiveness in targeting and maintaining control of each computer κ which has a representation in $\vec{v}$, perhaps similar to methods of modeling physical combat such as [22, 11]. We choose to simplify our model somewhat, and will consider these software interactions as part of what determines the payoff matrices and the changes to $\vec{v}$, but will not model them explicitly at this time. Modeling software infrastructure would be important future work in quantifying the game with realistic numbers from observation, but it is not necessary to understand the shape of the game as is described at the present level of abstraction.

An important feature is that the number of effective players in the game cannot be reduced by coalitions between the players. Although it is possible for the players to jointly improve their payoffs, because the game is non-zero-sum, the players cannot coordinate their actions in any coherent manner.

The rogue cannot cooperate with the user because they do not have an effective means to communicate or enforce agreements. The rogue is, by definition, achieving an unauthorized result. If the user were in a position to authorize the rogue's actions and come to an agreement, it would not be an attack scenario.

The architect and the rogue cannot be collapsed, although in the worst case for the user the architect and the rogue both are trying to undermine the user. Practically, if the architect designs a weak system the user may have few or no options that do not permit the rogue to perform a successful attack. However, the architect and the rogue have very different relationships with the user. The user and architect negotiate the payoff matrix, representing features of available software. The user will not wittingly negotiate with the adversary. And in most cases, the rogue also attacks the architect in order to bypass security controls, and so the same logic applies as to the user-rogue interaction. The architect cannot cooperate with the rogue because, by definition, the rogue is achieving an unauthorized result.

The user and the architect are practically prevented from forming a coalition because they cannot adequately communicate, share information, or enforce binding commitments. Thus, while the two players might wish to cooperate, and may be able to signal their intent to improve their situation [31], neither player is bound to cooperate. This situation seems similar to the classic "Battle of the Sexes" game [30]. The two players' interests do not align, but failure to agree — even when unilaterally choosing their preferred option — is worse than even the less preferred choice when agreeing.

3.3.2 Payoffs

We will not consider payoffs to be transferable and conservative, although arguments for doing so are plausible. A transferable payoff is one such that one player could transfer it, partly or wholly, to another player readily and without loss of value. Some of the elements of the payoff, such as time or computer resources, are not transferable. Some, such as money, are. If payoffs are transferable, bargaining becomes easier to analyze [29]. We do not believe this is accurate in our case. Since the players are not forming coalitions, non-transferable payments are less important, since most transfers are modeled as bargaining within coalitions or to entice a player to join a coalition. Even though money is a payoff, when involved in cyber-crime there is usually no practical way for the rogue to make payments to the user or the architect. For our purposes, we will thus simplify the payoffs and consider the payoff space S to be non-transferable, and thus exhaustive of all payoff options.

Each entry in S is a four-tuple, or an element with four distinct data elements. This includes the payoffs to each of the three players and the rogue's control over infrastructure vector, $\vec{v}$. Thus, S may be defined as:

(2) $s_{ijk} \in S; 0 \leq i \leq l; 0 \leq j \leq m; 0 \leq k \leq n$

(3) s_{ijk}: = $(p_a, p_r, p_u, \vec{v})$, *where* $p_a, p_r, p_u \in [0, 1]$

The values in each element s_{ijk} are the normalized real-number payoffs to the architect, rogue, and user, respectively, as well as the changes to vector $\vec{v}$ that is the outcome of that choice. The indexes i, j, and k are finite — at the present time there is no reason to believe the game has an infinite solution space. These values are the total number of strategies available to each of the architect, rogue, and user, respectively, at each step of the game.

The elements i, j, and k each indicate a strategy chosen by one of the players. Each time the game is played, each of the players receive the payoff at s_{ijk} and $\vec{v}$ takes on the value at s_{ijk}. The element s_{ijk} is selected out of S each time the game is played according to the strategies the players choose. Thus, the architect chooses the value of i, the rogue chooses the value of j, and the user chooses k. Strategies are chosen simultaneously. Each player chooses the value that will maximize their payoffs, however since s_{ijk} also affects $\vec{v}$ this consideration is more complex than usual. Players will consider investment and discounting when choosing their maximum payoff.

The payoffs themselves are represented as a function of $\vec{v}$. In this way, the payoffs can change in between renegotiations. More properly, the payoffs in each play of the game are a function of $\vec{v}$ at the previous play of the game. Therefore, we introduce the variable t to keep track of time in the game; it shall be incremented by 1 every time the game is repeated.

(4) $t \in [0, 1, 2, ..., T]$

(5) $p\alpha^t = f\alpha(\vec{v}^{\,t-1})$; $\alpha \in \{a, r, u\}$

While the payoff to each player is a function of $\vec{v}$, the payoff functions are also negotiated every so often by the user and architect. Thus, the extent to which $\vec{v}$ actually effects the payoffs to each is negotiable. The function is $f_\alpha()$ because it will be a different function for each of the architect, rogue, and user. The function $f_a()$ and $f_u()$ will produce smaller payoffs for the architect and user, respectively, with larger $\vec{v}$ since more malicious infrastructure will reduce their payoffs. $f_r()$ will produce larger payoffs for the rogue with larger $\vec{v}$. However, besides that the function is monotonically decreasing or increasing, respectively, the shape of the function (logarithmic, linear, etc.) is, in principle, negotiable.

More information about negotiated games can be found in [30]. How the payoffs are actually decided crosses into the psychology of the players and their relative power, and thus out of what pure game theory can determine. From a utility point of view the players will try to maximize their payoffs. The physical and psychological constraints of the world must be brought to bear on this negotiation modeling; otherwise it would be trivial for the players to simply set very large payoffs for everyone.

3.3.3 *Information sets*

The information available to each player will also need to be defined. In some cases, it is convenient to supply each player's subjective probability distribution over certain events for which information is incomplete [28]. However, this approach is perhaps more detailed than the present model is able to incorporate. More pertinent is each player's *information set*. The information set $\omega_\alpha^{\,t}$ for a player α is different at each point in the game t. The set $\omega_\alpha^{\,t}$ is the set of states of the game that the player knows may be the actual state of the game at time t, but between which the player cannot directly distinguish [31]. Thus, each player "knows which information set he is in, but not which vertex of the information set" [28].

Information sets help describe situations with uncertainty. In a game of 5-card poker, each player knows what cards they have, but not the cards any other player holds. However, the player knows each player has 5 cards. Certain probabilities can be calculated knowing the composition of a regular deck of cards, the player's hand, and how many people are playing. For example, if I hold 4 aces, I know that all situations in which another player holds an ace are impossible, and my information set of possible opposing hands does not include them.

Information sets in information security games are more complicated. One concrete example of this is when a user does not know whether a rogue has or has not compromised the user machine. If the user machine is infected, either the architect or the rogue could make a choice to change the user's information set. The architect can deploy accurate detection technologies and notify the user. The rogue can consume all the machine's resources, or erase the disks, which the user would notice. The user may select different strategies based on a change to their information set. Further, it does not seem that any player's information set is independent of the actions of any other player.

One element of ω_α includes a possible current state of $\vec{v}$, possible past states of $\vec{v}$, player α's past actions (in the case of imperfect recall, this will not be all past actions), as well as player α's beliefs about the possible past payoffs to the other players.

In modeling Internet security, the game's information structure is imperfect (6) and asymmetric (7), following the definitions in [31]. So there are information sets which contain more than one possible state of the game, and the information sets of different players are different. In symbols:

(6) $\forall\alpha(\exists\omega:\|\omega_\alpha\| > 1)$

(7) $\omega_\alpha{}^t = \omega_\beta{}^t \Leftrightarrow \alpha = \beta$

Exactly which elements are in $\omega_\alpha{}^t$ for each player α may be a matter of negotiation, as noted in Section 3.2. The extent the architect is permitted to monitor the user, for example, is in practice a function of the user's privacy concerns. Limiting information sets provides a formal way to discuss such concerns, as privacy partly means not being able to distinguish one user's data from another's.

4 Discussion

The game as proposed indicates some useful ways to think about the true nature of the real-world situation. The fact that the interaction between the rogue and the user is non-zero-sum is critical. This fact is due to the nature of digital resources — they are not truly rivalrous. Thus, there may be a strategy in which the rogue benefits and the user has negligible losses. In this case, the architect could not expect to impose constraints on the user to prevent the rogue's gains. The user's payoff may well be higher by not accepting such constraints. This situation helps explain the general difficulty the security community experiences with getting users to heed their warnings [3], for example.

The assertion that the game of network and Internet security as (at least) a three-person game is noteworthy. The game as described cannot be reduced to two players by putting two of the three players in a coalition. The facts of the Internet ecosystem prevent genuine coalitions in practice, and many interests of the parties do not align even in principle. Since the game has three players, a straightforward calculation of a globally-optimal strategy is not possible.

The game description also provides some practical guidance for policy and decision making. For example, if the payoff matrix is affected by the size of the rogue's infrastructure, and negotiations with the user community is stalled, then the architect's efforts would be best targeting at removing key elements of the criminal infrastructure. It also may be able to highlight certain areas that can only be solved politically as Internet governance issues, and so on.

The fact that each player has imperfect information, and that each player has different information about the game, is also a key point. Internet security is not chess, in which each player knows all the moves the other player makes — chess is a game of perfect information [30]. In chess, if one could enumerate the strategy space then one can select the globally-optimal strategy. Internet security should not be modeled as such a game, as the Internet does not function as a system with perfect information. Operational security cannot, in principle, hope to find a globally-optimal strategy.

5 Future Work

High level simulations of the posited formalisms would help to guide the plausibility of the formalisms. Establishing some hypothetical payoff matrices and attempting to calculate a solution or preferred strategy would also be an important next step. In general, all the formalizations can be made more detailed. More detail would then allow for a more rigorous analytic treatment, which would probably reveal more subtle strategic elements of the game.

The existence of any equilibria needs to be determined in order to guide other inquiries into intelligent strategies. Nash equilibria usually exist [31], for example, and a more detailed analysis could prove their existence for this game.

There is also a gap between this abstract analysis and practical measurement of the current state of affairs on the Internet that would need to be bridged before the model could be applied directly to the Internet. The present model is not sufficiently detailed to begin such measurement. Further, there is not a good framework for measuring crime on the Internet, as discussed in [5], although the authors therein propose some improvements. Eventually, such measurement efforts would need to be compatible with abstract modeling efforts so that the two can inform each other.

About the Author

Jonathan Spring is a member of the technical staff with the CERT Threat Analysis Group of the Software Engineering Institute, Carnegie Mellon University. He began working for the CERT program in 2009. He is the co-author of an information security textbook, "Introduction to Information Security: A Strategic-Based Approach," and also serves as an adjunct professor at the University of Pittsburgh's School of Information Sciences.

His research topics include monitoring cloud computing, DNS traffic analysis, and game theory. He holds a Master's degree in information security and a Bachelor's degree in philosophy from the University of Pittsburgh. Jonathan can be reached at netsa-contact@cert.org.

Acknowledgement

Thanks to Soumyo Moitra for his help in forming these ideas.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0000653

References

[1] : *2013 Data Breach Investigations Report (DBIR)*, 2014. URL http://www.verizonenterprise.com/DBIR/2013/.

[2] : *Black Tulip: Report of the investigation into the DigiNotar Certificate Authority breach*, 2012.

[3] Devdatta Akhawe, Adrienne Porter Felt: "Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness", *22nd USENIX Security Symposium*, 2013. URL http://www.cs.berkeley.edu/~devdatta/papers/alice-in-warningland.pdf.

[4] R. J. Anderson: *Security Engineering: A guide to building dependable distributed systems*. Wiley, 2008.

[5] R. Anderson, C. Barton, R. Böhme, R. Clayton, M.J.G. van Eeten, M. Levi, T. Moore, S. Savage: "Measuring the cost of cybercrime", *11th Workshop on the Economics of Information Security*, 2012. URL http://weis2012.econinfosec.org/papers/Anderson_WEIS2012.pdf.

[6] Steven J Brams: *Negotiation Games: Applying game theory to bargaining and arbitration*. Routledge, 2003.

[7] Huseyin Cavusoglu, Birendra Mishra, Srinivasan Raghunathan: "A model for evaluating IT security investments", *Communications of the ACM*, pp. 87—92, 2004.

[8] Adam Cummings, Todd Lewellen, David McIntire, Andrew Moore, Randall Trzeciak: *Insider Threat Study: Illicit Cyber Activity Involving Fraud in the U.S. Financial Services Sector*, 2012. URL http://www.sei.cmu.edu/library/abstracts/reports/12sr004.cfm.

[9] David Drummond: *A new approach to China*. Google Official Blog, 2010.

[10] L. Dolanskỳ: "Present state of the Lanchester theory of combat", *Operations Research*, pp. 344—358, 1964.

[11] Ellen Messmer: "RSA's SecurID security breach: What should you do?", *Network World*, 2011. URL http://www.networkworld.com/news/2011/031811-rsa-securid-breach.html.

[12] Ellen Messmer: "RSA's SecurID security breach: What should you do?", *Network World*, 2011. URL http://www.networkworld.com/news/2011/031811-rsa-securid-breach.html.

[13] Drew Fudenberg, Jean Tirole: *Game theory*. 1991. MIT Press, 1991.

[14] Herbert Gintis: *Game theory evolving: A problem-centered introduction to modeling strategic behavior*. Princeton University Press, 2000.

[15] Kuno JM Huisman: *Technology Investment: a game theoretic real options approach*. Kluwer Academic Pub, 2001.

[16] John Gilmore: DES (*Data Encryption Standard*) Review at Stanford University, 2005. URL http://www.toad.com/des-stanford-meeting.html.

[17] C. Kanich, N. Weaver, D. McCoy, T. Halvorson, C. Kreibich, K. Levchenko, V. Paxson, G.M. Voelker, S. Savage: "Show Me the Money: Characterizing Spam-advertised Revenue", *20th USENIX Security Symposium*, 2011. URL https://www.usenix.org/legacy/event/sec11/tech/full_papers/Kanich.pdf.

[18] Ioanna Kantzavelou, Sokratis Katsikas: "A game-based intrusion detection mechanism to confront internal attackers", *Computers & Security*, pp. 859—874, 2010.

[19] MK Lauren: *Describing Rates of Interaction between Multiple Autonomous Entities: An Example Using Combat Modelling*, 2001.

[20] S.D. Moitra: *Managing Risk from Cybercrime: Internet Policy and Security Management for Organizations*. Max-Planck-Institut f. ausländisches und internationales Strafrecht, 2008.

[21] Tyler Moore, Richard Clayton: "Evil searching: Compromise and recompromise of internet hosts for phishing", *Financial Cryptography and Data Security*, pp. 256—272, 2009.

[22] Roger B Myerson: *Game theory: analysis of conflict*. Harvard University Press, 1997.

[23] John F Nash Jr: "Non-cooperative games", *The Annals of Mathematics*, pp. 286—295, 1951.

[24] John F Nash Jr: "The bargaining problem", *Econometrica: Journal of the Econometric Society*, pp. 155—162, 1950.

[25] G. Owen: *Game theory*. Emerald Group Publishing, 1995.

[26] Anatol Rapoport: *N-person game theory: Concepts and applications*. Courier Dover Publications, 1970.

[27] Anatol Rapoport: *Two-person game theory: The essential ideas*. Courier Dover Publications, 1966.

[28] E. Rasmusen: *Games and Information: An Introduction to Game Theory*. Blackwell, 2007.

[29] R. Rasmussen, G. Aaron: *Global phishing survey: trends and domain name use in 2Q2012*, 2012.

[30] Sankardas Roy, Charles Ellis, Sajjan Shiva, Dipankar Dasgupta, Vivek Shandilya, Qishi Wu: "A survey of game theory as applied to network security", *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pp. 1—10, 2010.

[31] J.M. Spring: "Modeling Malicious Domain Name Take-down Dynamics: Why eCrime Pays", *IEEE eCrime Researchers Summit*, 2013. URL http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=88265

[32] T Spyridopoulos, G Karanikas, T Tryfonas, G Oikonomoug: "A Game Theoretic Defence Framework Against DoS/DDoS Cyber Attacks", *Computers & Security*, pp. 39—50, 2013.

[33] John Von Neumann, Oskar Morgenstern: *The theory of games and economic behavior*. Princeton university press, 1944.

[34] E Weinan, Bjorn Engquist, Xiantao Li, Weiqing Ren, Eric Vanden-Eijnden: "Heterogeneous multiscale methods: a review", *Communications in computational physics*, pp. 367—450, 2007.

[35] William Casey, Jose A. Morales, Thomson Nguyen, Jonathan Spring, Rhiannon Weaver, Evan Wright, Leigh Metcalf, Bud Mishra: "Cyber Security via Signaling Games: Toward a Science of Cyber Security", *ICDCIT*, pp. 34-42, 2014. URL http://dx.doi.org/10.1007/978-3-319-04483-5_4.

[36] Quanyan Zhu, Linda Bushnell, Tamer Basar: "Game-theoretic analysis of node capture and cloning attack with multiple attackers in wireless sensor networks", *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 3404—3411, 2012.