# Playing *Hide-and-Seek*: An Abstract Game for Cyber Security

Martin Chapman
Department of Informatics
King's College London
Strand, London, UK
martin.chapman@kcl.ac.uk

Gareth Tyson
Queen Mary, University of London, UK
gareth.tyson@qmul.ac.uk

Peter McBurney
Department of Informatics
King's College London
Strand, London, UK
peter.mcburney@kcl.ac.uk

Michael Luck
Department of Informatics
King's College London
Strand, London, UK
michael.luck@kcl.ac.uk

Simon Parsons
Department of Computer Science
University of Liverpool
Liverpool, UK
s.d.parsons@liverpool.ac.uk

## ABSTRACT

In order to begin to solve many of the problems in the domain of cyber security, they must first be transformed into abstract representations, free of complexity and paralysing technical detail. We believe that for many classic security problems, a viable transformation is to consider them as an abstract game of *hide-and-seek*. The tools required in this game – such as strategic search and an appreciation of an opponent's likely strategies – are very similar to the tools required in a number of cyber security applications, and thus developments in strategies for this game can certainly benefit the domain. In this paper we consider hide-and-seek as a formal game, and consider in depth how it is allegorical to the cyber domain, particularly in the problems of attack attribution and attack pivoting. Using this as motivation, we consider the relative performance of several hide and seek strategies using an agent-based simulation model, and present our findings as an initial insight into how to proceed with the solution of real cyber issues.

## Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

## General Terms

Experimentation, Security

## Keywords

Cyber Security, Hide-And-Seek Games, Search Games, Agent-Based Modelling

## 1. INTRODUCTION

Many classic problems in cyber security are too complex to approach in their natural form. That is, these problems contain such a vast array of entities, parameters and technical details, that attempts to apply existing problem-solving methodologies to them often lead to unhelpfully specific and highly complex solutions. For example, many instances of existing agent-based cyber research, are based on models that attempt to capture too much of the domain [8, 10]. To address this, we abstract complex cyber security issues to more fundamental problems, and use the tools of agent-based modelling and simulation to investigate them at this basic level, rather than in their concrete and detailed form. By doing so, we aim to explore some of the more thematic problems of cyber security, and produce more intuitive and generalisable results.[1] These results are able to guide future research, and serve as heuristics to help solve the original problem.

As an example, we consider the currently open problem of attack *attribution* [9], in which the task of a network administrator is not only to locate the origin of an attack, but also to present a chain of attribution, through proxy servers and spoofed Internet Protocol (IP) addresses, back to the targeted network. This is a complex problem with a number of technical facets. However, in many respects, what the network administrator is faced with is a simple game of *hide-and-seek*: the attacker has situated themselves on a network and it is the administrator's goal to locate this attacker by traversing the network's connections for accountability. Moreover, the attacker normally does not wish to be found, and has thus chosen their position *strategically*, in an attempt to make the administrator's goal as hard as possible — an adversarial element that needs to be considered when a search is undertaken. Therefore, in this scenario, the game serves as a useful abstraction; investigating strategies for it

[1]Our process of abstraction is identical with that used elsewhere in computer science (e.g. Turing machines for computation), as well as in domains such as physics (e.g., planets assumed to be perfect spheres) or economics (where so-called *stylised facts* are used to calibrate models).

can yield some insight into how best to proceed with related problems. Indeed, not only are the dynamics of hide-and-seek games of interest in their own right, but we also believe that the principles of such a game are widely applicable to a number of other security scenarios.

In the remainder of this paper, we consider a number of different strategies for both hiding and seeking, within an agent-based model (ABM), in order to better understand the dynamics of adversarial interactions in the cyber domain. The games we will study have been considered by game theorists, such as [5, 3], and similar game-theoretic approaches have been applied to conventional security problems, e.g., [16]. However, the models used in this work are often predicated on an unrealistic assumption of opponent rationality. Even if opponents are rational, they may exhibit apparently-irrational behaviour when they are represented by software agents, which are susceptible to bugs or errors. Through simulation, we believe we are able to more freely explore the potential for irrational behaviour. Moreover, our long-term interest is in the dynamics of interacting strategies, under different environmental conditions, and the intractability of these problems means that computational simulation is normally the only means we have available.

## 2. BACKGROUND

Before considering different hide-and-seek strategies, we first introduce a model of hide-and-seek games, and explain where this sits in relation to existing work. Following this, we consider how the components of such a model relate to real problems of cyber security.

### 2.1 Hide-and-seek

Hide-and-seek games belong to a wider class of games known as *Search Games*, which generalise the notion of one agent, a seeker, attempting to locate another agent, the hider. This search is conducted over a specific space, such as a network. The topology of this network constrains the movements of the players. Several mathematical studies of these games have been carried out to date, most notably by Steven Alpern and Shmuel Gal [5, 7]. When referring to hide-and-seek games, we adopt a similar model: we consider two players, $H$ the hider and $S$ the seeker; 'he' and 'she', respectively. We represent the graph that these two players are situated upon as $G = (V, E)$, where $V$ is a set of nodes, and $E$ a set of edges (paired vertices), such that $E \subseteq V \times V$. Let $n = |V|$. The edges of this graph are *weighted* such that there is an explicit cost associated with its traversal: $Cost : E \to \{0, \ldots, c\}$. In our model, this cost is determined according to a given random distribution between 0 and an upper bound $c$. The cumulative traversal cost incurred by a seeker serves as a *payoff* to the hider, while its inverse is a payoff to the seeker.

In addition, we introduce a set of nodes, selected by a hider, within which to conceal a series of *objects*. We refer to this set as $\mathcal{H}$ (where $\mathcal{H} \subseteq V$). Let $k = |\mathcal{H}|$. This subset is selected according to the hider's *strategy*, and, initially, we assume that the topology of the graph does not constrain the hider in this choice. Introducing a set of hidden objects allows us to consider multiple hide points. We represent a seeker's traversal of a graph as one or a series of *walks* where $W_i = \langle v_0^i, v_1^i, \ldots, v_{\ell_i}^i \rangle$. These walks begin at a given node $v_0^i$ and traverse the intermediate edges to a node $v_{\ell_i}^i$, where $(v_j^i, v_{j+1}^i) \in E$ for $0 \leq j < \ell$. Let $v_0^{i+1} = v_{\ell_i}^i$, so that a search is a connected path through the network in an attempt to locate all of the nodes in $\mathcal{H}$. If a walk causes the seeker to traverse one of these nodes, we say that the seeker has successfully located a hidden object. We initially assume that the seeker knows the value of $k$, and thus knows when the search has ended.

**Table 1: Parallels between our hide-and-seek model and features of the cyber security domain.**

| Model Entity | Notation | Feature of Cyber Domain |
|---|---|---|
| Hider | $H$ | An adversary attacking a network; an administrator defending a network. |
| Seeker | $S$ | An administrator defending a network; an adversary attacking a network. |
| Hidden Items | $k$, $\mathcal{H}$ | Intermediate attack points; known vulnerabilities. |
| Graph | $G$, $E$ and $Cost$ | The application-layer network within which an attacker exists; the network within which a vulnerability exists, such as an enterprise network; an abstract representation of required visitations. |
| Graph Size | $n$ | The size of a network; the number of tasks or stages that must be traversed. |

### 2.2 Hide-and-seek for Cyber Security

In the opening section, we stated our belief that the game of hide-and-seek shares many common principles with several open problems in cyber security. Therefore, we aim to use it as an abstract representation of these problems. In this section, we continue to motivate this comparison by considering the relevance of each element of our model to the cyber domain. A summary of this information is shown in Table 1.

Given our two agents, $H$ and $S$, the most natural parallels between our model and this domain occur when we frame $S$ as the network administrator in a secure infrastructure, and frame $H$ as an assailant on this infrastructure, attempting to conceal his attack path. It is under this framing that the game becomes analogous to the task faced in attack attribution. Therefore, the aim of experimenting with search strategies is to understand how best to discern a path to the attacker, given their adversarial behaviour. Whilst this is the most intuitive comparison, it is also possible to frame this in the opposite way: the administrators are the hiders wishing to conceal vulnerabilities, and it thus becomes important to understand how a seeker – in this case a hacker – will attempt to locate these. Viewing the game in this way also allows us to draw an additional analogy to the problems faced by a network administrator wishing to defend against a potential *pivoting* attack.[2] That is, when

[2]Pivoting attacks leverage existing compromised systems to

configuring the network, it is important for an administrator to understand how a series of vulnerable systems may be used to circumvent firewall restrictions, with the aim of creating a path to a vulnerable host. Thus, the administrator becomes the hider, strategically positioning these targets within the network. Additionally, imagine, for example, a service placement strategy in which an administrator wishes to host services in different areas of the infrastructure. Many factors could be considered in this decision including cost, proximity to consumers, legal constraints and quality of service requirements. This multi-faceted problem means that security considerations might be compromised, e.g., hosting services in a mobile telecoms cloud might improve user perceived performance but reduce security. In such a circumstance, using hiding strategies that mitigate the risk of compromise becomes extremely desirable. Given these alternate framings, it is important for us to understand the performance of *both* hide and seek strategies, in the face of varying adversaries.

In the problem of attack attribution, $G$ represents an overlay network, a portion of the nodes of which ($\mathcal{H}$) have been compromised by $H$ to conceal his origin. This is a practice common in the use of *Botnets*[3] or in the manipulation of other application-layer proxies. Therefore, $E$ serves as the potential paths through this network to the location of the attacker. Similarly, in the problem of pivoting, $G$ represents the private network within which a path of target hosts resides. In these search spaces, we believe $n$ should be as large as possible, in relation to $k$ (the number of attack points, or potential target hosts), to reflect the complexity of each search task. In addition to modelling real search spaces, we believe that the process of graph traversal effectively models the sequential nature of most search procedures within the ecosytem of the Internet. For example, querying ISPs and other stakeholders for data (e.g., packet traces and intrusion logs) requires many manual tasks, resulting in sequential preferential ordering of search points (imagine having to separately contact 500 network operators).

Having drawn several parallels between the static elements of our model and the cyber domain, we next consider a number of different strategies for both hiding and seeking (Table 2). These strategies could be adopted by the entities in the specific security problems referenced or, in fact, any suitable application case. Our aim in enumerating potential hide and seek strategies (and their corresponding parameters) is to introduce several ways to strategically engage in a game, and thus understand how to act when faced with comparable situations in cyber security, and how to program agents to act automatically.

# 3. STRATEGIES FOR HIDING

First, we consider strategies for hiding: either an attacker trying to obfuscate its attack or an administrator trying to hide their resources from such attackers.

## 3.1 Random Strategy

attack other systems within the same network, to an arbitrary length, in order to create an attack path to a targeted host.

[3]A Botnet is a collection of compromised hosts, elicited by a third-party in order to carry out large scale attacks.

**Table 2: Summary of hide-and-seek strategies.**

| Name | Agent | Description |
|---|---|---|
| `Stochastic` | Hider | Objects are hidden at random. |
| `Bias` | Hider | Preference is expressed towards a portion of nodes, when selecting where to conceal objects. |
| `Random` | Seeker | A search is conducted by randomly selecting an outbound edge when moving between the nodes of a network. |
| `Exploit` | Seeker | The search path is adapted to sequentially visit a set of nodes that have been recorded as likely to contain hidden objects. |

In order to attain maximum payoff, a hider must conceal his objects in a manner that will afford the seeker the highest cost. This would reduce the number of potential finders and delay the remaining ones. Thus, it is rational for him to consider each node as a hide location with equal probability: $P(v_i) = \frac{1}{n}$. We refer to this as a stochastic hiding strategy (`Stochastic`); intuitively, this strategy is optimal for an *ignorant* hider, in that it is invariant under the choice of search strategy. That is, we note that this optimality only holds under the assumption that a hider has no knowledge about the likely entry point of a seeker's traversal. If they do have this knowledge, it is clearly of more benefit to conceal all objects as far from this entry point as possible.

## 3.2 Biased Strategy

In reality however, a hider who is able to demonstrate such clinical objectivity is rare: they are, ultimately, human and are thus either *unable* or *unwilling* to express true randomness. In the former case, we refer to the notion of the *trembling hand* [15], in which human fallibility or other external factors cause players in competitive games to unintentionally exhibit bias or repeat behaviour. These 'trembles' cause a player to deviate from their optimal strategy. As mentioned, this can also manifest itself as bugs in software. In the latter case, such deviations are purposeful on the part of the hider; hiders do not wish to hide entirely randomly as they develop subtle preferences for hide locations due to, for example, the belief that some hiding locations may be more 'secretive' than others. Rubinstein has shown this to be true by examining real hider psyches, and demonstrating that even the most subtle properties of hide location – their relative positions, for example – can be used by a hider to frame a location as being preferable [14].

This behaviour often emerges in the cyber domain due to individuals' preferences and skill sets, which can often favour certain infrastructure. In the problem of attribution, for example, bias manifests itself as an attacker reusing proxies, repeating spoof points or using the same source IP address. This may be because an attacker prefers certain infrastructure, as stated above, or perhaps because he is simply unable to randomise his access points. Conversely, in the problem of attack pivoting, bias information is not necessarily the product of certain behaviour, but can be any information that allows a hacker to more easily locate and target vul-

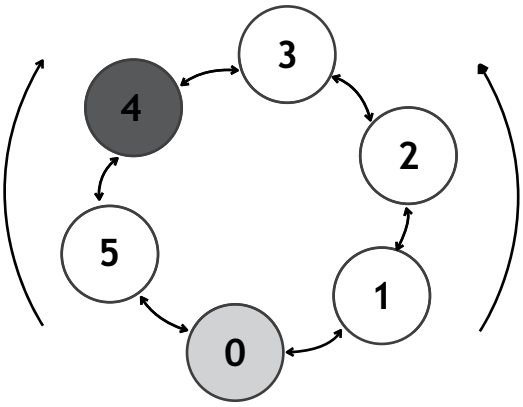


**Figure 1: Alpern and Gal's strategy proposes that the search for a hidden object (in this instance, concealed in node 4), should proceed either clockwise or anti-clockwise around an Eulerian circuit, from a starting point (0), with equal probability.**

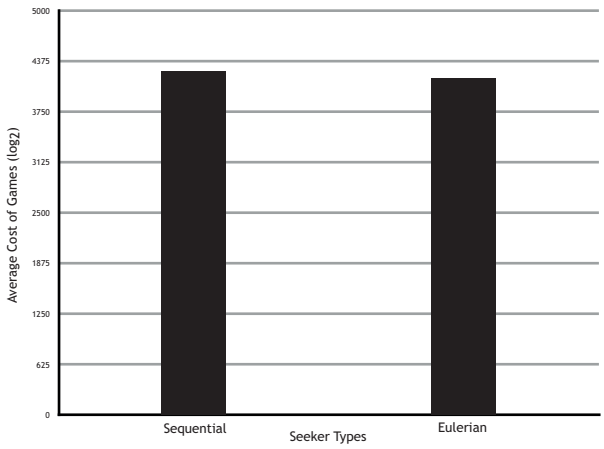


**Figure 2: The comparable performance of Alpern's strategy, and a sequential examination of the Eulerian Path.** $p = 0.23$ $(cl = .01)$**.**

nerable hosts, perhaps running particular operating systems with known vulnerabilities on intermediate machines.

As a result of these observations, we also introduce a *biased* hiding strategy into our model, `Bias`. That is, despite the optimality of a uniform random distribution under hider ignorance, hiders under this strategy make an irrational decision, and favour a fixed number of nodes ($q$) by a factor of $b$ (thus, $\max b = \frac{n}{q}$). Formally, we wish to attain that for certain bias nodes $v_j$, $P(v_j) = bP(v_i)$, therefore:

$$P(v_i) = \frac{1}{n + q \times (b-1)} \tag{1}$$

$$P(v_j) = \frac{b}{n + q \times (b-1)} \tag{2}$$

If $q < k$, then the remaining objects are hidden randomly.

## 4. STRATEGIES FOR SEEKING

Next, we consider strategies for seeking: either an attacker trying to find vulnerable resources or an administrator trying to find malicious entities. A random strategy is first presented, before detailing a biased strategy for exploiting known hider behaviour. Note that we do not present a full exploration of the design space; instead, we focus on two generic strategies that can be used to capture a variety of more complex equivalents.

### 4.1 Random Strategy

An Eulerian Graph contains an Eulerian Path, a path that visits each edge of a network exactly once. If we assume that $G$ is Eulerian, and the hider is uniform random in their strategy, then a well-cited search strategy is one proposed by Alpern and Gal. Under this strategy, a seeker traverses the Eulerian Path either forwards or backwards, with equal probability (shown in Figure 1 using a ring topology). Formally, given the closed walks $W = \langle v_0, v_1, \ldots, v_{n-1}, v_0 \rangle$ and $W' = \langle v_{n-1}, v_{n-2}, \ldots, v_1, v_0, v_{n-1} \rangle$, where the former constitutes the nodes in the Eulerian Path and has length $\mu$, and the latter represents the path in reverse order, the authors show that randomising between these walks with equal probability ensures that the payoff from a game does not exceed $\mu/2$. The choice between these two walks is made prior to the start of the search, and the trajectory it dictates is maintained until all hidden nodes are found. Despite the prevalence of this strategy, and the upper bound it places on the seeker's payoff, in reality it affords us little improvement on search cost beyond that of an examination of the Eulerian Path (i.e. always going in the same direction). This can be seen by running a simple simulation of two seekers on an Eulerian Graph, the results of which are shown in Figure 2. In addition to this, due to its dependence on the existence of such a graph, it is difficult to apply Alpern's strategy to many of the problems in cyber security; to assume that such a circuit exists – and can easily be calculated – is unrealistic in some of the complex networks where cyber security problems lie.

Thus, we consider an additional random search strategy that also considers a hider to be uniform random, but instead only assumes that $G$ is *connected*. This structure is far more flexible as it can also capture situations where a high degree of freedom is given to the seeker's traversal. For example, if seeking only requires sending a single probe to a host, this can be done in any sequence without the constraints of an Eulerian Path. This strategy can be modelled as a *random walk* on the graph (`Random`). That is, a walk of length $\ell$, $\langle v_0, v_1, \ldots, v_\ell \rangle$, where each $v_{i+1}$ is a vertex chosen at random from those nodes which are connected to $v_i$.

### 4.2 Exploit Strategy

As previously discussed, agents often have a tendency to show bias in their hiding behaviour. Consequently, it is logical that seekers should be sensitive to this bias. Thus, an alternate strategy emerges (`Exploit`) that aims to predict the behaviour of an opponent, and then use this information to guide the search. In order to predict behaviour, we introduce the function $Prob : V \rightarrow [0, 1]$, that maps each

node to a real number denoting the likelihood of $H$ concealing an object in that node. For example, a poorly secured node (e.g. an open relay SMTP server) would have a high likelihood of attack. This probability value is derived from an intermediate function, $Freq : V \rightarrow \mathbb{N}$, that maps each node to a natural number denoting the number of times a hider has chosen to conceal an object in that node. Initially, each value returned by $Prob$ is equally weighted as $\frac{1}{n}$, but upon observing a hider's behaviour, values are scaled proportionally according to the following:

$$Prob(v_j) = \frac{Freq(v_j)}{\sum_{v_i \in V} Freq(v_i)} \quad (3)$$

Thus, a crude form of learning takes place. We use the parameter $r$ to determine the portion of $k$ hidden objects that are located using this probabilistic information. If $r = 1$, only a single high probability node is used to guide the search. Formally, let $Pr_0 = \{Prob(v) \mid v \in V\}$ be our set of probability values and let $T_0 = \{v \in V \mid Prob(v) = \max(Pr_0)\}$ be those values which are maximal amongst this set. The node $t_0$ is selected from $T_0$ randomly. Let $L_0 = \{t_0\}$ hold this single node. Given this information, the walk $\langle v_0, v_1, \ldots, t_0 \rangle$, denotes the initial search path, under the constraint that the edges between $v_0$ and $t_0$ constitute the edges in the *shortest path*. At node $t_0$, if all elements of $\mathcal{H}$ have not been located, a random walk is undertaken until all concealed objects are found.

If $r > 1$, then multiple high probably nodes are used, and determined as a set of the maximal nodes indicated by $Prob$. To do this, the node that corresponds to the highest probability is considered first (selecting one randomly if multiple maximal nodes exist), it is removed, and then the next most likely considered until $r$ likely nodes are found. We formalise this inductively, such that $Pr_k = \{Prob(v) \mid v \in V \setminus L_{k-1}\}$, $T_k = \{v \in V \setminus L_{k-1} \mid Prob(v) = \max(Pr_k)\}$, and $t_k$ is chosen from $T_k$ randomly. Thus, $L_k = L_{k-1} \cup \{t_k\}$ and the set $L = \bigcup_{i=0}^{r-1} L_i = L_r$ contains the $r$ nodes.

To traverse the graph based on this information, a series of $r$ walks, $\langle W_0, \ldots, W_r \rangle$, are produced, where the edges between $v_0^i$ and $v_{\ell_i}^i \in W_i$ constitute the edges in the shortest path. Recall that $v_0^{i+1} = v_{\ell_i}^i$. To incorporate the information on high probability nodes, we ensure $v_{\ell_i}^i = t_i$. In other words, this strategy chooses the first node in $L$, traverses the shortest path to this node, and repeats this process until the paths to all permitted high probability nodes have been traversed. If, at this point, $k$ items have not been located, the graph is traversed according to a random walk until all hide locations have been discovered. Note that our future work involves expanding the sophistication of this biased strategy to better model other real-world considerations of importance (e.g. urgency and cost).

# 5. SIMULATIONS AND RESULTS

In order to assess the relative performance of these search strategies under different conditions, and against different hiders, we translate the conceptual model detailed in the previous sections into a simulation model. Our aim in doing so is to dynamically observe the interactions between different agents, in order to gain the empirical perspective not provided by existing analytic work [4, 1, 2]. We argue that only by using this approach are we able to fully understand the practical value of our strategies.

**Table 3: Parameters in a game of hide-and-seek**

| Parameter | Notation |
|---|---|
| Games per Simulation | $gps$ |
| Rounds per Game | $rpg$ |
| Topology | $top$ |
| Number of nodes in a graph | $n$ |
| Number of hide locations | $k$ |
| A hider's bias | $b$ |
| Number of high probability nodes used in a search | $r$ |
| Proportion of hide locations which are bias | $q$ |

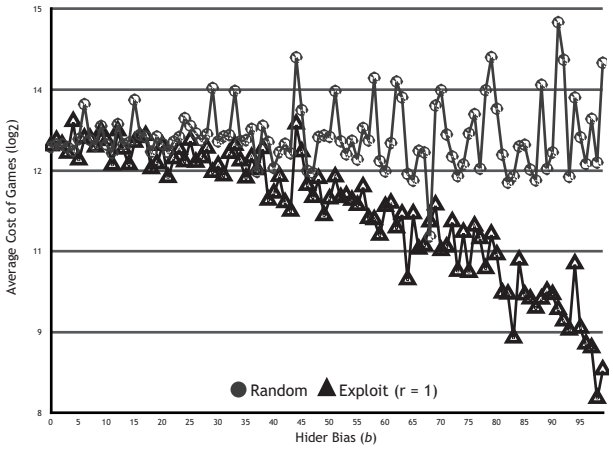


**Figure 3: The performance of a bias-sensitive strategy against a random walk. ($gps$ =100; $rpg$=120; $top$=random; $n$=100; $k$=1) $p = 1.36 \times 10^{-20}$ ($cl = .01$).**

## 5.1 Methodology

Each simulation contains a number of *games*, and each game consists of a number of *rounds*. Structuring our simulations as individual games allows us to vary certain parameters, the most pertinent of which are shown in Table 3. During a round, $H$ computes $\mathcal{H}$ and updates the network as such, and $S$ traverses the network until each element of $\mathcal{H}$ is discovered. The presence of rounds gives a seeker with the capacity for information collection (learning) the opportunity to do so. We therefore view rounds as *repeated interactions* between an attacker and a defender in the security domain. The outcome of each interaction yields some information about the other's strategy; for example, preferred attack source points or favoured exploit paths. At the end of each round we record the cumulative traversal cost incurred by the seeker. The average cost across all rounds in a game becomes that game's cost. Clearly, a seeker (a security administrator) wishes to limit the measured cost, whilst a hider (an attacker) wishes to inflate this figure.

## 5.2 Searching for a Single Object

We first consider the relative performance of $S_{\texttt{Random}}$ and $S_{\texttt{Exploit}}$, against $H_{\texttt{Bias}}$, when searching for a single hidden

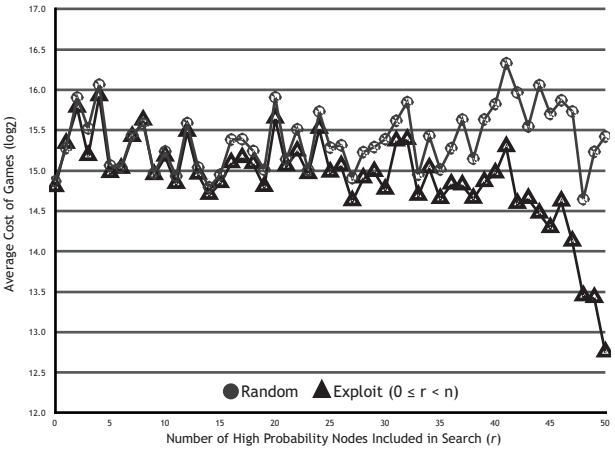


**Figure 4: The performance of a bias-sensitive strategy against a random walk, when it incorporates different proportions of trend information into its search. This is measured against a hider with known bias. (*gps*=50; *rpg*=120; *top*=random; $n$=100; $k$=50) $p = 4.30 \times 10^{-7}$ ($cl = .01$).**

object ($k = 1$). More specifically, we are interested in the degree to which $S_{\texttt{Exploit}}$ is able to exploit the biased behaviour of $H_{Bias}$, and if so, how *much* bias is required for this exploitation to occur. To test this, we ran a number of games in which we increased the bias of our hider, such that $0 \leq b < n$. This models a situation where an attacker is injecting attacks on nodes with varying degrees of preference for each node. Our evaluation topology in this instance is *randomly wired*, where $|E| = 3n$ to loosely ensure full connectivity on initial generation. We consider $c$ (the maximum cost of traversing an edge in a random distribution) to be 100. The results of this simulation are shown in Figure 3. The downwards curve in cost seen for $S_{\texttt{Exploit}}$ shows us that performance gains *are* attained when a seeker employs a strategy that is sensitive to the bias of a hider. This is intuitive as it allows a seeker (security administrator) to learn the behavioural traits of a hider (attacker). However, the gradient of this curve is by no means sharp, leading us to conclude that the bias shown must be significant ($b \gtrsim 45$) before notable improvements in cost can be made: a discovery that might exacerbate the complexity of preempting attacks. Whilst this is not a positive result for the seeker, it does tell us that a hider can afford to favour a particular node a significant amount, and still obtain a reasonable payoff. Therefore, this suggests attackers could potentially take a relatively lazy approach to hiding their attack points.

## 5.3 Searching for Multiple Objects

With $k > 1$, we again consider the relative performance of $S_{\texttt{Random}}$ and $S_{\texttt{Exploit}}$, against $H_{\texttt{Bias}}$. Recall that the parameter $r$ indicates the portion of high probability nodes that are used by $S_{\texttt{Exploit}}$ to guide a search. We consider the optimum value of $r$ given complete knowledge of a hider's bias, and no knowledge of this bias.

### 5.3.1 *r*-values with a known bias

Let $q = k$ and $b = \frac{n}{k}$, so that our hider conceals all objects with maximum bias. In this scenario, we are interested in the value of $r$ that allows the `Exploit` strategy to maximise its performance against $H_{\texttt{Bias}}$. To do this, $r$ becomes the primary variable in our simulations, such that $0 \leq r < k$, and we increase $k$ to a much larger value: 50.

The results of this simulation are shown in Figure 4. This graph shows us that when $r = k$, maximum performance is attained by $S_{\texttt{Exploit}}$, as it utilises complete information on the hider's behaviour to guide its search. Whilst this is intuitive, the increase in performance attained by incorporating an increasing number of high probability nodes into a search is not linear. Indeed, it is only when $r \simeq k$, that performance is notably different from a random walk. This contrasts with our natural assumption that additional information, although still only partially contributing to the understanding of a hider's behaviour, will always increase the efficiency of a search. Knowing that additional information does not necessarily equate to an increase in performance has several implications for both a seeker and a hider.

Assuming that a seeker is in the process of *gathering* information on the behaviour of a hider, she should have no incentive to include this information into her search unless she has information on how $\sim k$ objects are hidden (assuming she knows the value of $k$). If the seeker does not have sufficient information, she can expend less computational effort on conducting a random walk, and obtain comparable performance. An information gathering process such as this is common in the cyber domain, when a complete picture of the hider's behaviour (favoured proxies, for example) is not always immediately clear, due to their obscurity. This result also tells us the importance of having up-to-date behavioural information, as any partial obsolescence of this data significantly affects performance. Conversely, this graph tells the hider the proportion of nodes that he can *afford* to be biased towards. That is, if we assume that bias is unavoidable (as motivated in Section 3), the next best thing to consider is the proportion of $\mathcal{H}$ that *can* contain bias nodes, without the hider's strategy becoming too predictable. As this graph shows us that the seeker does not benefit from incorporating information on high probability nodes into her search strategy until she has information on $\sim k$ of them, this also tells us that the hider can afford to be biased towards $\sim k$ nodes, before he experiences a significant reduction in payoff.

### 5.3.2 *r*-values with unknown bias

In the previous section, we showed that the number of high probability nodes that need to be incorporated into a search in order to maximise performance against an entirely biased hider is equal to $k$. However, if $q < k$, then a portion of objects may not be hidden with bias, but simply randomly. More worryingly, not only may a portion of objects be hidden randomly, but they may also be hidden with deceptive intent, in order to purposefully allow a seeker to learn incorrect patterns. This is a practice observed in the cyber domain prior to large attacks. It is also motivated by the graph in Figure 4, which shows us that only a small amount of incomplete or inaccurate information can detrimentally affect the performance of a seeker, i.e. a hider can easily undermine a search by introducing false information into their knowledge base. In both these cases, searching with probability information for all nodes may be misleading. Because of this, we again consider the performance

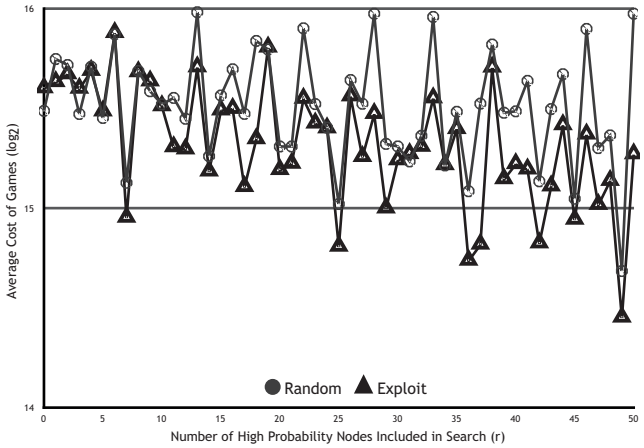


**Figure 5: The performance of a bias-sensitive strategy against a random walk, when it incorporates different proportions of trend information into its search. This is measured against a hider with unknown bias. ($gps$=50; $rpg$=120; $top$=random; $n$=100; $k$=50) $p = 0.00054$ ($cl = .01$).**

of the `Exploit` strategy under various values of $r$, but this time when $q$ is unknown. To do this, we ran a simulation, again in which $0 \leq r < k$, but in each round $q$ was selected randomly, to reflect its unknown value. Figure 5 presents a graph of this simulation, and shows comparable performance between both strategies, regardless of the amount of probabilistic information included in a search. Thus, faced with an unknown ratio of objects hidden with bias to those hidden randomly (or with misleading trend information), the choice of $r$ becomes arbitrary. Indeed, we could even infer that the choice of $r$ should be random to provide a strategically symmetric choice to the random selection of $q$. For the hider, this information tells us that if he does have a preference for certain locations, he should randomise which of these locations are used, with each interaction that he has with the seeker. By doing this, it is clear that the seeker's learning process is significantly impaired.

## 5.4 Summary of Results

From our investigation of hide-and-seek as a formal game, it is clear that it possesses many difficult problems. Faced with a hider who is entirely stochastic, a seeker must look to traverse an Eulerian Path, if one exists, and if not, engage in a random walk in order to ensure that all nodes are examined. This is a costly activity. In reality, however, intuition and existing studies have shown us that despite the rationality of a uniform hide strategy, hiders often exhibit notable bias. Given the reliable existence of this bias, we have shown that a seeker is able to improve her performance by recording a hider's behaviour and producing predictive information about his hiding strategy. Whilst it may be tempting to use this information when it is available for a significant portion of the $k$ objects that need to be located, we have shown that only once enough information is obtained to guide the search towards nearly all the nodes in $\mathcal{H}$, does performance exceed that of a random walk. This information also shows us that a hider can afford to be biased in some cases, without immediately opening himself up to exploitation by the seeker. If a seeker is unsure about the *extent* of a hider's bias, or suspects they have intent to exhibit false behaviour, we have shown that the best she can do is to incorporate information on an arbitrary number of hidden objects into her search.

The difficulty of this game certainly reflects the difficulty of those cyber problems with which it shares common principles. What we are able to now suggest, however, are certain *heuristics* for how to act, given the results we have obtained. For example, in the task of attribution, we are able to suggest that a network administrator should not attempt to create an attribution path to an attacker when she only has information on a portion of his intermediate source points, as this partial knowledge does not aid her in the discovery of the remainder of the path. Instead, she should wait for more holistic information, and in the interim, examine all network connections in turn. This may not be entirely intuitive, as the administrator may be tempted to use *any* information she has to guide her search, under the belief that as she gains more knowledge she will attain linear performance gains. If we consider our alternate framing, in which the hider is instead a benevolent entity – specifically, an administrator preparing his network for a potential pivoting attack – this result also tells us that whilst a large proportion of the exploitative paths to certain targets in the network may be known by the attacker, this does not necessarily mean that other targets will be also easily be compromised. Again, this may not be immediately intuitive.

# 6. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a methodology that seeks to abstract complex cyber security problems to more simple representations, in order to foster solutions that are not overly complicated by lower-level concerns. Whilst we believe this is the most effective way to begin to solve difficult cyber problems, it is important to acknowledge the sensitivity of the results obtained to the parameters chosen for simulation. That is, we acknowledge the impact that our chosen variables are likely to have on the results, and make it our primary point of future work to conduct further experiments to measure this impact. We suspect, for example, that the value of $k$, and the ratio of $k$ relative to $n$, will impact the performance of some of our strategies, and the optimal configuration of their parameters (e.g. $r = k$ may not always allow a seeker to attain maximum performance against a biased hider at varying values of $k$). We suggest that to guide the choice of these parameters, it may be necessary to source real security data.

In addition to this, we note the existence of a number of cyber scenarios that are not constrained by any form of conceptual topology, and thus for which our current model does not serve as a suitable analogy. In these scenarios, the seeker is able to move freely between hide locations, and cost is instead attributed to an exploration of the node itself. Therefore, another key point of future work will be to run a number of games *without* the presence of a topology, using additional strategies as necessary. In doing this we are likely to draw on the work of Lidbetter, amongst others, who has proposed a strategy for node search which does not consider the constraints of a network [11].

Whilst this paper has considered hide-and-seek strategies *separately*, we are also keen to investigate the potential for the dynamic *co-evolution* and *co-adaptation* of strategies. That is, by endowing our agents with the ability to both learn their opponent's strategy, and update their own, we are interested in how each player's performance adapts, given the adaptation of their opponent. Specifically, we are interested if any notable equilibria emerge. To help analyse this phenomenon, we will use techniques from evolutionary game theory, as in [12, 13], which have explored co-evolution and co-adaptation marketplaces. Introducing a capacity for co-evolution and co-adaptation into our model may also allow us to better explore the psychological element of the game, if, for example, players attempt to deceive one another about their chosen strategy, and this must be reasoned about across many levels, as in [6].

Finally, we wish to expand our model of the hide-and-seek paradigm. This expanded model is likely to feature a hider that is constrained by the topology, yielding more diverse hide strategies, or a hider who similarly tracks the behaviour of a seeker; players with variable initial knowledge, who must either learn the values of $k$ or discern the topology of the search space; and multiple hiders and seekers, organised into a hierarchy of control.

## 7. ACKNOWLEDGEMENTS

We acknowledge the impact of Christopher Hampson, Thomas Lidbetter and Elizabeth Sklar on this work.

## 8. REFERENCES

[1] S. Alpern. Hide-and-seek games on a tree to which eulerian networks are attached. *Networks*, 52(3):162–166, 2008.

[2] S. Alpern, V. Baston, and S. Gal. Network search games with immobile hider, without a designated searcher starting point. *International Journal of Game Theory*, 37(2):281–302, 2008.

[3] S. Alpern, R. Fokkink, L. Gasieniec, R. Lindelauf, and V. Subrahmanian, editors. *Search Theory: A Game Theoretic Perspective*. Springer, 2013.

[4] S. Alpern and S. Gal. Searching for an agent who may or may not want to be found. *Operations Research*, 50(2):311–323, 2002.

[5] S. Alpern and S. Gal. *The Theory of Search Games and Rendezvous*. Springer, 2002.

[6] H. de Weerd, R. Verbrugge, and B. Verheij. Higher-order social cognition in rock-paper-scissors: A simulation study. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1195–1196. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[7] S. Gal. On the optimality of a simple strategy for searching graphs. *International Journal of Game Theory*, 29(4):533–542, 2001.

[8] V. I. Gorodetsky, I. V. Kotenko, and J. B. Michael. Multi-agent modeling and simulation of distributed denial of service attacks on computer networks. In *Proceedings of the Third International Conference on Navy and Shipbuilding Nowadays, St. Petersburg, Russia*, 2003.

[9] C. Guitton. *Achieving Attribution*. Ph.D thesis, King's College London, London, UK, 2014. *Forthcoming*.

[10] I. Kotenko. Agent-based modelling and simulation of cyber-warfare between malefactors and security agents in [the] internet. In *Proceedings of the 19th European Simulation Multiconference*, 2005.

[11] T. Lidbetter. Search games with multiple hidden objects. *SIAM Journal on Control and Optimization*, 51(4):3056–3074, 2013.

[12] S. Phelps, P. McBurney, and S. Parsons. Evolutionary mechanism design: a review. *Autonomous Agents and Multi-Agent Systems*, 21(2):237–264, 2010.

[13] E. Robinson, P. McBurney, and X. Yao. Market niching in multi-attribute computational resource allocation systems. In W. Ketter, K. R. Lang, and K. J. Lee, editors, *Proceedings of the 13th International Conference on Electronic Commerce (ICEC 2011)*, Liverpool, UK, August 2011 2011.

[14] A. Rubinstein. Experience from a course in game theory: pre-and postclass problem sets as a didactic device. *Games and Economic Behavior*, 28(1):155–170, 1999.

[15] R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International journal of game theory*, 4(1):25–55, 1975.

[16] M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.