2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Comparison of Strategies for Honeypot Deployment

Joel Brynielsson*†, Mika Cohen*†, Patrik Hansen†, Samuel Lavebrink*, Madeleine Lindström*, Edward Tjörnhammar*†
*KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
†FOI Swedish Defence Research Agency, SE-164 90 Stockholm, Sweden
Email: joel@kth.se, mikac@kth.se, patrik.hansen@foi.se, samlav@kth.se, madeli@kth.se, edward@foi.se

***Abstract*—Recent experimental studies have explored how well adaptive honeypot allocation strategies defend against human adversaries. As the experimental subjects were drawn from an unknown, nondescript pool of subjects using Amazon Mechanical Turk, the relevance to defense against real-world adversaries is unclear. The present study reproduces the experiments with more relevant experimental subjects. The results suggest that the strategies considered are less effective against attackers from the current population. In particular, their ability to predict the next attack decreased steadily over time, that is, the human subjects from this population learned to attack less and less predictably.**

***Index Terms*—Cybersecurity; honeypot; game theory; defense strategy; behavioral learning.**

I. Introduction

Deception and the prospects for reasoning about well-thought-out adversarial actions, play an increasingly important role in cyber defense [1]. The subject is today an important part of nations' counterintelligence and security efforts in general, as well as observed in itself by government-supported initiatives such as the National Cyber Deception Laboratory (NCDL) in the United Kingdom.[1] In order for deception mechanisms to successfully deceive adversaries, their application (timing, location, configuration, etc.) must be unpredictable: a mock-up that is always deployed at the same location, and with the same appearance, will fool no one.

Honeypots, fake hosts introduced into a network in order to attract attackers, are an established form of deception mechanism in network defense [2]. Recent work explores how effective various honeypot allocation strategies are at outwitting human adversaries [3]. Each strategy is evaluated based on how well it defends against a human adversary in a game simulating the interaction between an attacker and a defender in a computer network. The experiments are performed using Amazon Mechanical Turk,[2] which provides a large pool of experimental subjects at low cost. Unfortunately, however, the subjects remain completely unknown. It is unclear, therefore, to what extent the population is relevant to the performance of honeypot allocation strategies, i.e., real world adversaries such as hackers, have the same cognitive profile as the pool of unknown subjects participating in the experiments. It is an open and contested question as to what extent, in general, results from controlled experiments with human subjects transfer from one kind of population to another [4].

[1]https://www.cyberdeception.org.uk/.
[2]https://www.mturk.com/.

The current study reproduces the honeypot experiments and econometric analysis presented by Aggarwal et al. [3], but with a population of high-achieving students from science and technology programs at the Royal Institute of Technology in Stockholm, Sweden. Arguably, this population is more relevant when evaluating network defense than the pool of unknown subjects from Amazon Mechanical Turk; presumably, their cognitive profile is more similar to that of actual adversaries in network defense. The results suggest that the proposed adaptive honeypot allocation strategies considered, are less effective against the studied population.

II. Defense Strategies

As cyber adversaries become more dynamic, adapting their behavior to the defense they meet, a static cyber defense policy may be outdated soon after it is deployed. This section briefly reviews some adaptive strategies for allocating limited security resources (e.g., honeypots) to defend assets (e.g., databases on a network) against an adaptive adversary (e.g., a network intruder) considered in the literature. The selection of strategies is inspired by the work by Aggarwal et al. [3].

A. Static Pure (SP)

An allocation strategy that might suggest itself immediately to a defender is to distribute the available security resources over the most valuable assets. This static, somewhat naive strategy, here referred to as static pure (SP), is introduced in the experiments below as a baseline to compare other, less predictable strategies with.

B. Static Equilibrium (SE)

The problem of allocating resources (to defend assets against an attacker) can be viewed as a double sided game in which one player (the defender) chooses between a number of possible allocations while the other player (the adversary) chooses between a number of assets to attack [5]. The optimal strategy for the defender in such a game, according to the classical game-theoretic solution (Nash), is to randomize allocations in such a way as to provide the adversary with the same expected pay-off no matter what asset the attacker chooses to attack. This leaves the adaptive adversary nothing to adapt to; there is no bias or tendency in the behavior of the defender for the adversary to exploit.

ASONAM'23, November 6–9, 2023, Kuşadası, Türkiye

ACM ISBN 979-8-4007-0409-3/23/11
https://doi.org/10.1145/3625007.3631602

C. Learning with Linear Rewards (LLR)

The resource allocation problem can alternatively be seen as a multi-armed bandit (MAB) [6]. Each possible allocation forms an arm of the bandit machine, with its effect determined by the unknown, possibly stochastic, behavior of the adversary. With the allocation problem viewed as a MAB, the goal of the defender (gambler) is to minimize the adversary's long-term reward over repeated interactions by balancing the trade-off between exploitation (choosing resource allocations that has the best observed record so far) and exploration (choosing less explored allocations). Learning with linear rewards is an adaptive strategy for that purpose [7]. It generally favors the action that seems optimal given the rewards observed so far, but occasionally chooses an at the time sub-optimal action, to explore new alternatives.

Pseudocode for the algorithm is presented in Algorithm 1. Here, F is the set of all possible actions and A_a is the index set $\{i : a_i \neq 0, 1 \leq i \leq N\}$ where the action a is a vector of size $1 \times N$ with elements $a_i = 1$ if arm indexed i is to be pulled and $a_i = 0$ otherwise. The total number of different arms is denoted N, θ is a vector that contains the mean observed reward for each arm, m is the vector containing the number of times each arm has been included in an action, and $0_{1\times N}$ is a zero matrix of size $1 \times N$.

Algorithm 1 Pseudocode for LLR

Initialize: $\theta = 0_{1\times N}$, $m = 0_{1\times N}$, if $\max_a |A_a|$ is known, let $L = \max_a |A_a|$; else $L = N$
for t=1 to N **do**
  Play any action a such that $t \in A_a$
  Update θ, m accordingly
end for
for t=N+1 to ∞ **do**
  Play an action a which solves the maximization:
  $a = \arg\max_{a \in F} \sum_{i \in A_a} \theta_i \sqrt{\frac{(L+1)\log(t)}{m_i}}$
  Update θ, m accordingly
end for

D. Best Response with Thompson Sampling (BR-TS)

Best response with Thompson sampling is an adaptive strategy that chooses the action with the highest estimated reward by predicting the adversaries' behavior [3], [8]. As the adversary behavior model, Thompson sampling for the MAB problem is used, where it has been shown to be a good predictor of human behavior in games [9], [10]. Since Thompson sampling is being used as a predictor, the actions of the attacker might not be known. In that case, the game is simulated thousands of times with the action sampled from a prior attack distribution, to predict the action taken by the adversary. The goal in this implementation of Thompson sampling is to minimize the possible reward for the attacker in the single round ahead; later rounds are not considered.

E. Probabilistic Best Response with Thompson Sampling (PBR-TS)

Probabilistic best response with Thompson sampling is an adaptive strategy, which uses randomization [3]. It is similar to the strategy BR-TS described above. The difference between the two methods is that PBR-TS samples the action from the distribution of possible rewards instead of choosing the one with the highest expected reward as BR-TS does. An action which yields a higher expected reward results in a greater probability of being played by the defender.

F. Follow the Regularized Leader (FTRL)

Follow the regularized leader is an adaptive strategy that takes into consideration that the attack actively changes its perception of the expected reward for each arm, resulting in a change in attack distribution as the game progresses [11]. To avoid overfitting the model based on previously chosen arms, a regularizer has been added. Viewing the allocation of security resources as a MAB as in the case for the LLR strategy results in the FTRL algorithm presented in Algorithm 2.

Algorithm 2 Pseudocode for FTRL

Input: $\gamma \in (0, 1]$, sampling scheme P
Initialize: $L = 0_{1\times N}$
for t=1 to ∞ **do**
  $\eta_t = \frac{1}{\sqrt{t}}$
  $x_t = \arg\min_{x \in Conv(x)} \langle x, L_{t-1}\rangle + \eta_t^{-1}\psi(x)$ for $\psi(x)$
  Sample action a from $P(x_t)$
  Observe result $o_t = a \circ l_t$
  for i=1 to N **do**
    $\hat{l}_{ti} \leftarrow \frac{(o_{ti}+1)}{x_{ti}} - 1$
  end for
  $L_t = L_{t-1} + \hat{l}_t$
end for

Here, the regularized leader x_t is computed with the cumulative estimated loss L_{t-1} and regularizer $\psi(x)$ [3]. The regularizer is defined as:

$$\psi(x) = \sum_{i=1}^{N} -\sqrt{x_i} + \gamma(1 - x_i)\log(1 - x_i),$$

where N is the total number of arms. The regularizer is applied with the learning rate η_t. The loss vector l_t is created by the environment based on the adversary's action, and $a \circ l_t$ is the observed result of the action chosen by the algorithm where $\circ$ is the element-wise multiplication operator.

III. Method

The effectiveness of each defense strategy above, as a policy for allocating honeypots in a computer network, was evaluated based on how well it defends against human adversaries in an artificial game simulating the interaction between an attacker and a defender. In the following, the game, implementation details, experimental settings, and evaluation metrics, are described.

A. HoneyGame

The HoneyGame [12] is a simple two-player game intended as an abstraction of the interaction between an attacker and a defender in a computer network. The game consists of a network of six nodes, each with a cost of defending the node, c_i^d, and a cost of attacking the node, c_i^a. The value of a node is the sum of the cost of defending and attacking the node: $v_i = c_i^d + c_i^a$. For the adversary, the possible reward of a node is equal to the cost of defending, c_i^d, and the possible loss is equal to the cost of attacking, c_i^a. The sixth option, pass, has $c_i^d = 0$ and $c_i^a = 0$. The node parameters c_i^d, c_i^a and v_i for each node i is presented in Table I.

TABLE I
NODE PARAMETERS IN THE HONEYGAME.

	Node 1	Node 2	Node 3	Node 4	Node 5	Pass
c_i^d	10	20	15	15	20	0
c_i^a	5	20	10	5	15	0
v_i	15	40	25	20	35	0

In each round, the defender spends a maximum budget of $D = 40$, to place honeypots on a subset of the nodes. The placement of honeypots on a subset of the nodes is called an action, a. The attacker then chooses a node to play. If the attacker chooses a node without a honeypot, he/she receives the reward c_i^d while the defender incurs a loss of 0. However, if the chosen node is a honeypot, he/she receives the loss $-c_i^a$ and the defender acquires a reward of v_i. If the attacker chooses pass, he/she does not receive a reward nor a loss.

The game is carried out over 50 rounds, where the attacker is given the node options presented above as well as the current round number, total points, and time remaining that round. The attacker is not presented with the previous placements of honeypots nor which strategy he/she plays against.

B. Defense Strategy Implementation

Static Pure: In a single round of the game where the opponent's strategy is unknown, using a deterministic action, maximizing the defender expected reward results in placing the honeypots on the nodes with the highest values. With the defense budget $D = 40$ and the node parameters presented in Table I, this results in the honeypots being placed on nodes 2 and 5. This pure strategy is implemented in all 50 rounds of the game.

Static Equilibrium: The resulting distribution of the mixed strategy Nash equilibrium for a single round of the game is presented in Table II. The static equilibrium defender samples an action a randomly from this distribution in each round.

TABLE II
STATIC EQUILIBRIUM DISTRIBUTION.

Action (defended nodes)	$\{1,3,4\}$	$\{2,3\}$	$\{2,5\}$	$\{3,5\}$
Probability	≈ 0.303	≈ 0.095	≈ 0.557	≈ 0.045

Learning with Linear Rewards: Connecting the HoneyGame with the notion of MAB, the pulling of arms on the MAB can be seen as the placement of honeypots on nodes. When implementing Algorithm 1 as a defense strategy in the HoneyGame, L is the maximum number of honeypots placed on nodes in a round; here $L = 3$ for the action $a = \{1, 3, 4\}$, θ is a vector that contains the mean observed reward for each node, m is the vector containing the number of times each node has been defended, and $N = 5$ due to there being 5 possible nodes honeypots can be placed at.

Best Response with Thompson Sampling: The implementation of the defense strategy BR-TS in HoneyGame results in Algorithm 3. For every round t, the resulting attack distribution $\mathcal{D}$ over the $N = 5$ different nodes, is computed from $n_{sim} = 1500$ simulations. Thompson sampling is used in every simulation to predict the action of the adversary. In the case where the attacked node is unknown, the action is sampled from a uniform distribution. The defender chooses the action that results in the highest estimated reward based on $\mathcal{D}$.

Algorithm 3 Pseudocode for BR-TS/PBR-TS

Initialize: $S = 0_{1\times N}$, $F = 0_{1\times N}$
for t=1 to ∞ **do**
  $\mathcal{D} = 0_{1\times N}$
  for n=1 to n_{sim} **do**
    for all previous rounds t **do**
      if chosen node i is a honeypot **then**
        $F_i(t+1) \leftarrow F_i(t) + 1$
      else if chosen node is not a honeypot **then**
        Sample node i from uniform distribution over nodes without honeypots
        $S_i(t+1) \leftarrow S_i(t) + 1$
      end if
    end for
    $\mu_i \leftarrow Beta(S_i + 1, F_i + 1), \forall i$
    $\mathcal{D}_i = \mathcal{D}_i + \frac{1}{n_{sim}}$ where $i = \arg\max_{i'} \mu_{i'}$
  end for
  BR-TS: Choose action a that would result in the highest expected reward for the attacker if played next round, based on the attack distribution and value v_i of the node
  or
  PBR-TS: Choose action a by sampling from the distribution of possible rewards
  Observe result
end for

Probabilistic Best Response with Thompson Sampling: The difference between the two methods PBR-TS and BR-TS is in how they choose which nodes to place honeypots on. When PBR-TS chooses which action to play, it samples from the distribution of possible rewards, instead of choosing the greatest one as BR-TS does. An action which yields a higher expected reward results in a greater probability of being played by the defender. The resulting algorithm is presented alongside BR-TS in Algorithm 3.

TABLE III
RESULTING P-VALUE OF PERMUTATION TESTS OF STRATEGY-PAIRS, TOTAL POINTS PER STRATEGY.

	SP	SE	LLR	BR-TS	PBR-TS	FTRL
SP	–	0.0002	0.0002	0.26	0.0002	0.0002
SE	0.0002	–	0.0008	0.0004	0.95	0.027
LLR	0.0002	0.0008	–	0.0002	0.0006	0.0002
BR-TS	0.26	0.0004	0.0002	–	0.0002	0.023
PBR-TS	0.0002	0.95	0.0006	0.0002	–	0.016
FTRL	0.0002	0.027	0.0002	0.023	0.016	–

Follow the Regularized Leader: To implement FTRL as a defense strategy, Algorithm 2 is adapted to HoneyGame where action a is sampled from the sampling scheme P. $P(x_t)$ is sampled by calculating the action closest to the argument x_t (i.e., the action a that minimizes the distance to x_t in the 5D space). The parameter γ is set to 0.999.

C. Experiment

A total of 124 games were played by students from science and technology programs at the Royal Institute of Technology in Stockholm, Sweden. In groups of 5–10 students, each player was given a base reward of two chocolate bars for participating, and the player with the highest score received three additional bars.

Every participant filled out a form where information about their age, highest achieved university degree, occupation, and average university grade, was collected. The participants were then informed about the rules of HoneyGame. The game was played as described in Section III-A, with a randomly selected defense strategy, and a time limit of 30 seconds per round, using a graphical user interface similar to Table I.

The participants' ages ranged from 20 to 49; 98 in ages 20-24, 20 in ages 25-29, and 6 in ages 30-49. Between 16 and 28 games were played against each strategy; 16 against SP, 19 against SE, 20 against LLR, 22 against BR-TS, 28 against PBR-TS, and 19 against FTRL. The total time per game was calculated to 56.8 ± 8.3s.

D. Evaluation

From the data collected during the experiments, the average points, the proportion of attacks on honeypots, and the switching behavior, were visualized and studied for every strategy according to Fig. 1–3.

To validate the statistical significance of the results for the total points per strategy, permutation tests were performed. Here, the hypothesis is that the average points from two different defense strategies differ significantly. The null hypothesis is that the samples of total points from the two different strategies are from the same distribution. The significance level $\alpha = 0.05$ was used.

IV. RESULTS

The average points per strategy is presented in Fig. 1. The mean and standard deviation for each strategy were calculated

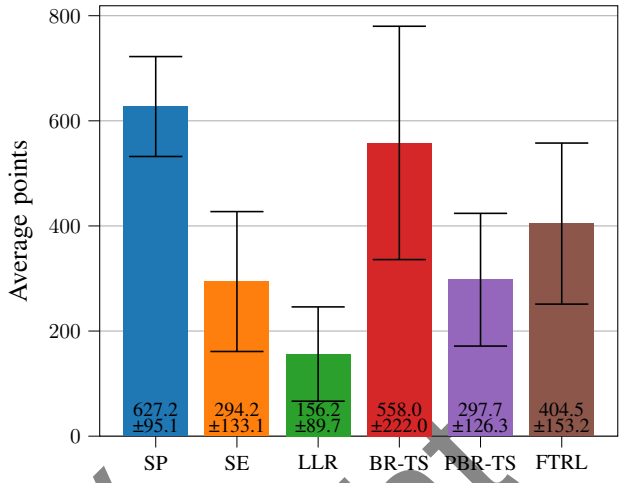


Fig. 1. Average points per strategy scored by attackers.

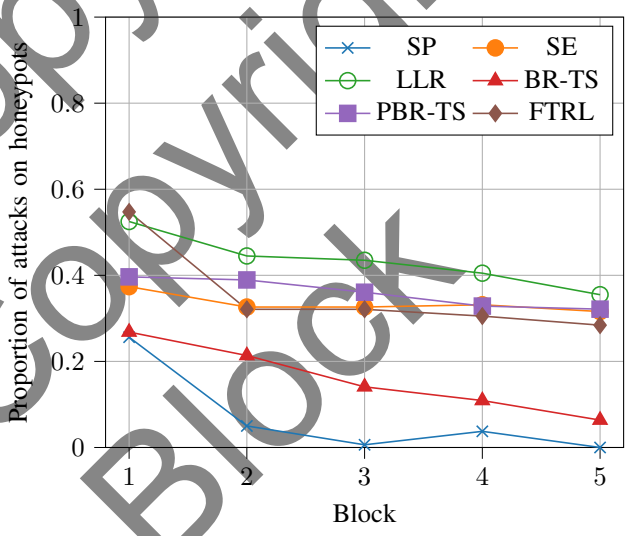


Fig. 2. Proportion of attacks on honeypots per block and strategy.

to: 627.2 ± 95.1 for SP, 294.2 ± 133.1 for SE, 156.2 ± 89.7 for LLR, 558.0 ± 220.0 for BR-TS, 297.7 ± 126.3 for PBR-TS, and 404.5 ± 153.2 for FTRL. This is also shown at the bottom of each bar. The results from the permutation tests are presented in Table III. All p-values are below the significance level $\alpha = 0.05$, apart from the strategy-pairs SP – BR-TS and SE – PBR-TS.

The proportion of attacks on honeypots, i.e., the number of times the attacker chose a honeypot divided by the total number of attacks, per block and strategy is shown in Fig. 2. Here, a block consists of ten rounds, i.e., block one refers to round 1–10, block two refers to round 11–20, etc.

Fig. 3 shows the switching behavior for the six strategies, i.e., the likelihood of someone staying on the same node or switching node based on whether their previous attack hit a honeypot or not. The top boxes show the probability of shifting which node to attack; the top left box is representing *Honeypot-Shift* behavior and the top right box is representing

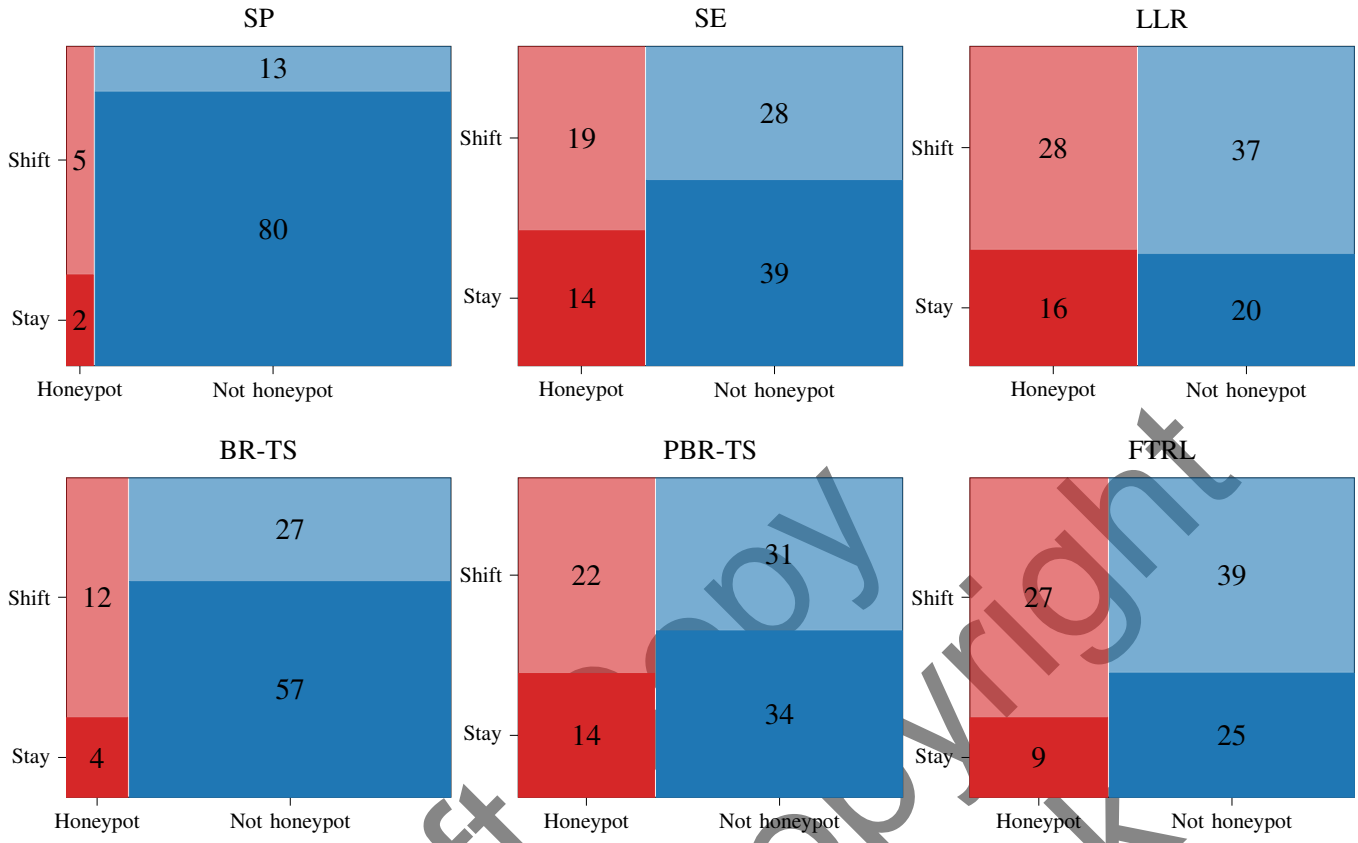


Fig. 3. Attacker switching behavior for SP, SE, LLR, BR-TS, PBR-TS, and FTRL. The numbers represent the proportion (%) per behavior.

Not honeypot-Shift behavior. The bottom boxes show the probability of staying at the same node; the bottom left box is representing *Honeypot-Stay* behavior and the bottom right box is representing *Not honeypot-Stay* behavior. A visualization of how the switching behavior evolved over time can be seen in Fig. 4. The plots corresponding to the same behavior are located in the same place as the boxes in Fig. 3, i.e., the top left plot indicates *Honeypot-Shift* behavior, and so on. Specifically, Fig. 4(d) shows how the switching behavior changes over time specifically for the *Not honeypot-Stay* case. Fig. 4(c) shows how the switching behavior changes over time specifically for the *Honeypot-Stay* attacker strategy. As before, the game is divided into five blocks of ten rounds each.

V. Discussion

In the following, the performance of the strategies and the observed attacker behavior are discussed and analyzed.

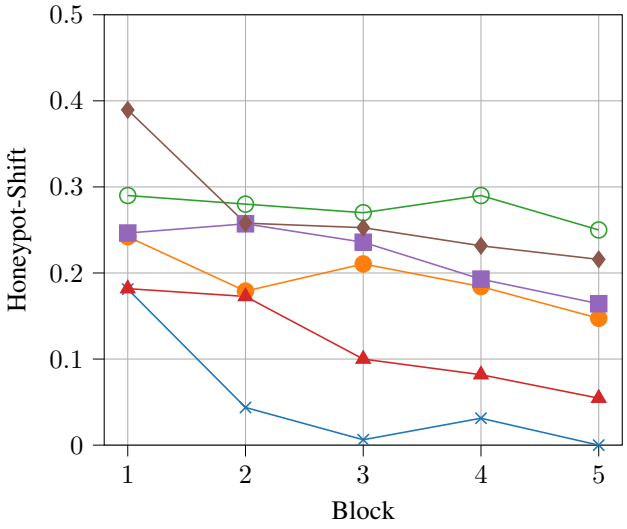
A. Performance of the Defense Strategies

The experiment resulted in various total points for the attacker, dependent on which strategy they played against (Fig. 1). SP resulted in the highest mean points for the attacker. It also resulted in the second-lowest standard deviation, indicating that most players against SP received many total points. Comparing SP to the strategy with the second highest average total points, BR-TS, it is noticeable that BR-TS has a higher standard deviation, the largest of all strategies. This suggests that some attackers could get as many points against BR-TS as other players got against SP. However, it also indicates that some players did not get as many points against BR-TS as those who played against SP. BR-TS therefore seems like an inconsistent strategy, considering the points for the attacker.

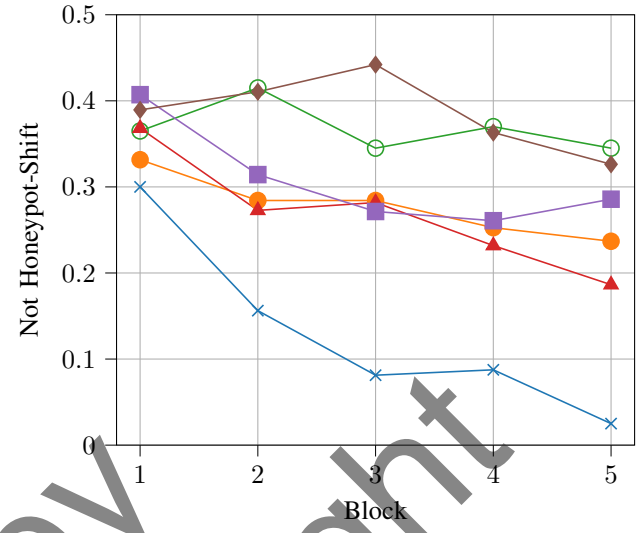
The permutation test for the strategies SP and BR-TS resulted in a p-value of 0.26, a value above the significance level (see Table III). Therefore, we cannot reject the null hypothesis for SP and BR-TS; hence, the samples of total points for SP and BR-TS could come from the same distribution.

In Fig. 2, the proportions of honeypot attacks are seemingly the same for SP and BR-TS in blocks one and five. However, the proportions of attacks on honeypots decreased more rapidly for SP, suggesting that the attacker learned how to play against SP faster than against BR-TS. From block two and onwards, the proportion of attacks on honeypots is below 10% for SP. This proportion is, however, not reached for BR-TS until block five.

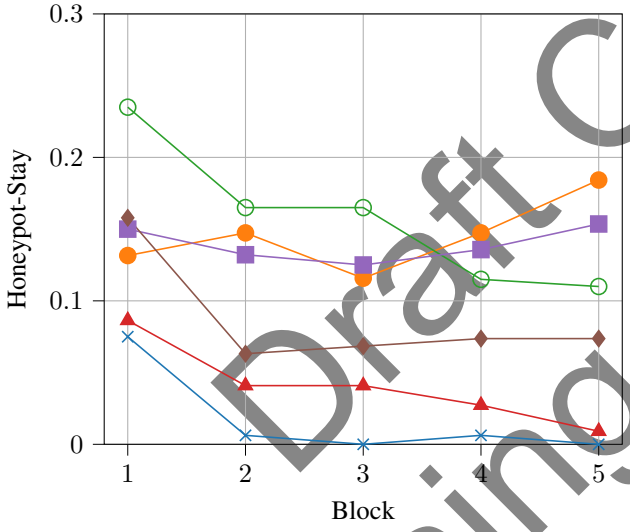
The strategy which the players achieved the third-highest average points against was FTRL, as seen in Fig. 1. FTRL had a mean of 404.5 ± 153.2 points, roughly 100 points less than BR-TS but also 100 points more than the strategy with the fourth highest total points. The standard deviation of 153.2 sits between the standard deviation of SP and BR-TS, suggesting that FTRL is more consistent than BR-TS but not as consistent



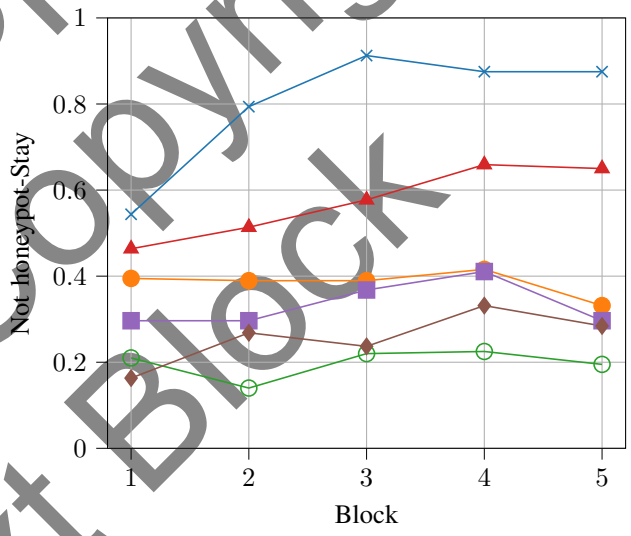
(a) Attacker switching behavior for the *Honeypot-Shift* attacker strategy.



(b) Attacker switching behavior for the *Not-Honeypot-Shift* attacker strategy.



(c) Attacker switching behavior for the *Honeypot-Stay* attacker strategy.



(d) Attacker switching behavior for the *Not honeypot-Stay* attacker strategy.

Fig. 4. Attacker switching behavior per block for SP (—×—), SE (—●—), LLR (—○—), BR-TS (—▲—), PBR-TS (—■—), and FTRL (—◆—).

as SP. The solution, x_t, to the minimization problem:

$$x_t = \underset{x \in Conv(x)}{\arg\min} \ \langle x, L_{t-1} \rangle + \eta_t^{-1} \psi(x),$$

often had all values close to one. In most cases, x_t was closest to the point $(1, 0, 1, 1, 0)$ and the algorithm therefore favored the action $\{1,\ 3,\ 4\}$. This is because it is the only valid action with more than two nodes defended. Only occasionally FTRL chose to defend another set of nodes.

The strategies resulting in the second and third least average points were SE and PBR-TS. PBR-TS had a mean total points of 297.7 ± 126.3, and SE had a mean of 294.2 ± 133.1. Hence, these strategies performed similarly. This is supported by the result from the permutation test, where this strategy-pair resulted in the p-value 0.95 (see Table III). The null hypothesis could, therefore, not be rejected for this strategy pair. Both SE and PBR-TS use randomization when placing honeypots, making it more difficult for attackers to find non-honeypot nodes. Based on how simple the implementation of SE is, this difference in average points is most likely because the players performed poorly against this defender. Due to the randomness used in SE, it could be assumed that the attackers' performances vary against this algorithm. Similarly to the average points, the proportions of attacks on honeypots in Fig. 2 are also evenly matched between SE and PBR-TS.

The defense algorithm that resulted in the least average points for the attacker was LLR (156.2 ± 89.7). From Fig. 1, one can see that the average points were almost half of the second-best defense algorithm. LLR also has the smallest

standard deviation of the strategies. Compared to SP, which resulted in the highest average points, the difference in standard deviation is negligible. Percentage-wise, SP is, therefore, the most consistent strategy. From Fig. 2, it is clear that LLR was the best performing strategy, as it has proportionally the most number of attacks on honeypots in all blocks apart from the first one, where it is marginally lower than FTRL.

To conclude, the ability of the adaptive strategies (LLR, PBR-TS, BR-TS, and FTRL) to predict attacks decreased over time (see Fig. 2); the human adversary seemed to learn how to attack less and less predictably. By contrast, no such learning could be observed in the subjects in Aggarwal et al. [3]. Indeed, FTRL gained in efficiency, and LLR sustained its efficiency over time in Aggarwal et al. [3].[3]

B. Behavior of the Attacker

The switching behavior of the attacker in Fig. 3 shows that the *Not honeypot–Stay* behavior is common in the strategies SP (80%) and BR-TS (57%). As described in Section II-A, SP always chooses the action $\{2,\ 5\}$; therefore, an attacker should find it easy to circumvent it. The results confirm this as it had the highest total points and the highest *Not honeypot–Stay* percentage. SP showed similar results in the study by Aggarwal et al. [3], where the strategy resulted in 75% for *Not honeypot-Stay* behavior. The idea that these participants learned over time is supported by the results in Fig. 4(d), where the behavior *Not honeypot–Stay* increased over time for SP and BR-TS. The proportion of attacks on honeypots (Fig. 2) also decreased for the two strategies during the game, supporting the idea that the attacker learned where not to attack.

When playing against FTRL, the attacker leaned more towards *Shift* behavior; both when previously choosing a honeypot (27% for *Shift* and 9% for *Stay*) and not hitting a honeypot (39% for *Shift* and 25% for *Stay*), as seen in Fig. 3. The preference to *Shift* rather than to *Stay* when hitting a honeypot was also presented by Aggarwal et al. [3]. However, the preference to do the same thing when not hitting a honeypot was not reported by Aggarwal et al. [3]. They instead received no difference between *Not honeypot-Stay* and *Not honeypot-Shift* behaviors. The algorithm often, but not always, chose to play the action $\{1, 3, 4\}$. However, the attackers did not notice that the strategy was almost static. This observation is supported by the non-changing behaviors reported over the game's duration. Neither did the proportion of attacks on honeypots change significantly during the game (see Fig. 2).

The strategies utilizing randomization to place honeypots, SE and PBR-TS, both resulted in a relatively even distribution of *Stay* and *Shift* behaviors; both when the attacker hit a honeypot or not (see Fig. 3). The attackers were slightly more prone to stay if they did not hit a honeypot and more prone to shift if they did. There are no discernible changes in the attacking behavior over time for either SE or PBR-TS. This is consistent with the findings presented by Aggarwal et al. [3], both for overall behavior and over time. Due to the randomness of these strategies, it can be assumed that the attacker did not find a pattern to increase attack performance, resulting in a lower amount of total points (see Fig. 1) and no change in the proportion of attacks on honeypots (see Fig. 2) for these two strategies.

The strategy that resulted in the lowest total points for the attacker (see Fig. 1), LLR, also resulted in the attacker's behavior being independent on whether he/she hit a honeypot. Fig. 3 shows the proportions for *Honeypot-Stay*/*Honeypot-Shift* compared to *Not honeypot-Stay*/*Not honeypot-Shift*, which are almost identical. The attacker exhibited a higher inclination to *Shift* than to *Stay* regardless of whether he/she hit a honeypot. Similar to SE and PBR-TS, no considerable change in attacking behavior can be seen for LLR in Fig. 4(d), implying that the attacker did not find a performance-increasing strategy.

VI. Conclusions

Previous research has shown that authorities are poorly prepared for antagonistic deceptive cyber behavior, with examples related to, e.g., critical infrastructure [13] and the financial system [14]. Such unpredictable antagonistic behavior requires that one's own organization also act unpredictably, and have a plan for strategic modeling and use of deception and information manipulation. In this regard, honeypots are valuable tools in a cybersecurity arsenal, which can be used to lure hackers, cybercriminals, and malicious actors, in a controlled and monitored environment [2]. Honeypots help organizations gain insights into cyber threats, enhance their security posture, and provide a proactive approach to identifying and mitigating risks. However, their deployment should be carefully planned and managed, and honeypots must be strategically placed in an unpredictable manner to effectively deceive and deter hackers [15]. Random and unexpected deployment locations enhance their efficacy by keeping malicious actors guessing, making it more challenging for them to distinguish between genuine assets and traps within the attacked network.

The study described in this paper has explored how effectively diferent honeypot allocation strategies defend against human adversaries in a game simulating the interaction between an attacker and a defender in a computer network. The study reproduced experiments from the literature, but with experimental subjects more relevant to cyber defense.

In the study, the ability of adaptive defense strategies to predict attacks decreased over time; the human adversary learned to attack less and less predictably. By contrast, no such learning could be observed in the subjects in previous studies. Indeed, FTRL gained in efficiency, and LLR sustained its efficiency over time in the earlier studies.

[3]Here the strategies PBR-TS and BR-TS in Aggarwal et al. [3] are disregarded, since the implementation differs from the one herein, cf. Section II. Moreover, as implemented by Aggarwal et al. [3], PBR-TS and BR-TS begin their interaction with each subject with a model that has been fine-tuned beforehand to match the expected initial behavior of subjects. Therefore, initially, PBR-TS and BR-TS perform well. On the other hand, their initial behavior is also very rigid—preferring a small number of possible allocations—making it almost impossible for a subject to fail and notice this preference and adapt, thereby rapidly reducing the effectiveness of the strategy.

The static and almost static algorithms, SP and BR-TS, performed the worst in this study. BR-TS intends not to be static, but in reality it became almost static due to the lack of a prior attack distribution. As with BR-TS, the strategy FTRL could not be implemented as desired, resulting in a worse result than in earlier studies. For FTRL, the absence of a distribution to sample honeypot placements from, is suspected to be the reason for this difference.

As expected, the strategies where randomization was used, SE and PBR-TS, performed considerably better. There was no distinct change in the behavior of the attacker during the course of the game for these strategies. The fact that no pattern of honeypot locations could be detected may explain this.

The best performing strategy, however, was LLR. While the effectiveness of LLR decreased over time, steadily approaching that of the static equilibrium (SE), its effectiveness stayed slightly above the equilibrium even towards the end of the fifty interactions long experiment, suggesting that LLR retained some ability to exploit the behavior of subjects even after repeated interactions. Perhaps the subjects would eventually have learned how to outsmart LLR, pushing its effectiveness below that of the equilibrium. To answer this, further experiments, with an increased number of interactions between defender and attacker, will be needed.

A. Future Work

Further research ought to focus on improving BR-TS, and potentially also PBR-TS, by using a prior attack distribution. When implemented, a comparison to earlier BR-TS studies should be made to evaluate the improved implementation. Future work should also investigate the possibility of presenting an attack distribution to be used in FTRL, when sampling honeypot placements. This distribution should be weighted in a way that does not favor the action $\{1, 3, 4\}$ as much as in the implementation reported on herein.

Another focus for future studies would be to include new strategies for comparison with the ones already examined. The new strategies could either be already existing strategies (that were not included in the present study), or newly developed strategies. Strategies that are constructed from scratch should include the properties that were found to be successful within this work in terms of being adaptive and including a certain degree of randomness.

Finally, future work should be directed to evaluating the different defense strategies within different populations. In this regard, it would be interesting to test the strategies relative to a group of test subjects being experienced in cybersecurity, to investigate how the strategies perform against a group that can be compared to more experienced hackers. This since it is more likely that cyberattacks are performed by people possessing more experience within cybersecurity compared to university students, and thereby potentially being able to exploit defense strategies in both better and different ways. Testing the strategies on, for example, personnel active with daily operational work in CERTs (computer emergency response teams), would therefore be a kind of ultimate test of the strategies' performance in reality.

ACKNOWLEDGMENTS

The authors would like to acknowledge and thank Farzad Kamrani, FOI Swedish Defence Research Agency, for insightful methodological discussions, and Ulrik Franke, RISE Research Institutes of Sweden, for comments that improved outline and readability.

REFERENCES

[1] U. Franke, A. Andreasson, H. Artman, J. Brynielsson, S. Varga, and N. Vilhelm, "Cyber situational awareness issues and challenges," in *Cybersecurity and Cognitive Science*, A. A. Moustafa, Ed. Elsevier, 2022, ch. 10, pp. 235–265.

[2] L. Spitzner, *Honeypots: Tracking Hackers*. Addison-Wesley, 2003.

[3] P. Aggarwal, M. Gutierrez, C. D. Kiekintveld, B. Bošanský, and C. Gonzalez, *Evaluating Adaptive Deception Strategies for Cyber Defense with Human Adversaries*. John Wiley & Sons, Ltd, 2021, ch. 5, pp. 77–96.

[4] A. Geib, *Predicting Human Decision-Making: From Prediction to Action*, 1st ed., ser. Synthesis lectures on artificial intelligence and machine learning. Netherlands: Springer Nature, 2018.

[5] A. Sinha, F. Fang, B. An, C. Kiekintveld, and M. Tambe, "Stackelberg security games: Looking beyond a decade of success," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 5494–5501. [Online]. Available: https://doi.org/10.24963/ijcai.2018/775

[6] L. Xu, E. Bondi, F. Fang, A. Perrault, K. Wang, and M. Tambe, "Dual-mandate patrols: Multi-armed bandits for green security," *arXiv:2009.06560 [cs, stat]*, 2020, arXiv: 2009.06560. [Online]. Available: http://arxiv.org/abs/2009.06560

[7] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1466–1478, 2012.

[8] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3–4, pp. 285–294, 1933, doi: 10.1093/biomet/25.3-4.285.

[9] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," in *Proceedings of the 25th Annual Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, S. Mannor, N. Srebro, and R. C. Williamson, Eds., vol. 23. Edinburgh, Scotland: PMLR, Jun. 2012, pp. 39.1–39.26. [Online]. Available: https://proceedings.mlr.press/v23/agrawal12.html

[10] M. Speekenbrink and E. Konstantinidis, "Uncertainty and exploration in a restless bandit problem," *Topics in Cognitive Science*, vol. 7, no. 2, pp. 351–367, 2015.

[11] J. Zimmert, H. Luo, and C.-Y. Wei, "Beating stochastic and adversarial semi-bandits optimally and simultaneously," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, Jun. 2019, pp. 7683–7692. [Online]. Available: https://proceedings.mlr.press/v97/zimmert19a.html

[12] M. Gutierrez, N. Ben-Asher, E. Aharonov, B. Bošanskỳ, C. Kiekintveld, and C. Gonzalez, "Evaluating models of human adversarial behavior against defense algorithms in a contextual multi-armed bandit task," in *41st Annual Meeting of the Cognitive Science Society (CogSci 2019), Montreal, QC (2019 (in press))*, 2019.

[13] S. Varga, J. Brynielsson, and U. Franke, "Information requirements for national level cyber situational awareness," in *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2018)*, Barcelona, Spain, Aug. 2018, pp. 774–781, doi: 10.1109/ASONAM.2018.8508410.

[14] S. Varga, J. Brynielsson, and U. Franke, "Cyber threat perception and risk management in the Swedish financial sector," *Computers & Security*, vol. 105, Jun. 2021, doi: 10.1016/j.cose.2021.102239.

[15] C. Kiekintveld, V. Lisý, and R. Píbil, *Game-Theoretic Foundations for the Strategic Use of Honeypots in Network Security*. Cham: Springer International Publishing, 2015, pp. 81–101. [Online]. Available: https://doi.org/10.1007/978-3-319-14039-1_5