# An Exploratory Study of a Masking Strategy of Cyberdeception Using CyberVAN

Palvi Aggarwal[1], Omkar Thakoor[2], Aditya Mate[3], Milind Tambe[3], Edward A. Cranford[4], Christian Lebiere [4] and Cleotilde Gonzalez[1]
[1] Dynamic Decision Making Laboratory, Carnegie Mellon University, Pittsburgh, USA
[2] University of Southern California, Los Angeles, California, USA
[3] School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
[4] Psychology Department, Carnegie Mellon University, Pittsburgh, USA

During the network reconnaissance process, attackers scan the network to gather information before launching an attack. This is a good chance for defenders to use deception and disrupt the attacker's learning process. In this paper, we present an exploratory experiment to test the effectiveness of a masking strategy (compared to a random masking strategy) to reduce the utility of attackers. A total of 30 human participants (in the role of attackers) are randomly assigned to one of the two experimental conditions: Optimal or Random (15 in each condition). Attackers appeared to be more successful in launching attacks in the optimal condition compared to the random condition but the total score of attackers was not different from the random masking strategy. Most importantly, we found a generalized tendency to act according to the certainty bias (or risk aversion). These observations will help to improve the current state-of-the-art masking algorithms of cyberdefense.

## INTRODUCTION

Just like in physical war, in cyber war deception is an important weapon. The art of war is the art of deception used by attackers as well as defenders (Griffith, 1963). Cyber attackers master deception quite well; they use various deception techniques such as frequently changing malware signatures, concealing code, encrypting exploits, and deceiving the end-user through social engineering attacks such as Phishing. Similarly, cyber defenders rely on many cyber deception techniques (Gonzalez et al., 2020). Decoys (e.g., honeypots) are commonly used to lure attackers (Aggarwal et al., 2016; Ferguson-Walter et al., 2017), and they have been effective in gathering information about attacker's tactics and to catch illicit activity or assist in slowing attackers down (Spitzner, 2003).

Network reconnaissance is the first step in the *cyber kill chain* cycle (i.e., involving reconnaissance, lateral movement, and exploitation), where attackers gather information about a target before an attack is executed. During reconnaissance, attackers use different scanning tools (e.g., Nmap, Nessus, Nikto, etc.) to learn about the network infrastructure, services and vulnerabilities. These scanning tools provide information such as number of systems and their connections, Operating System, and ports and services in the network.

Cyber defenders can use "masking" as a cyberdeception technique during reconnaissance; changing the configuration provided, instead of providing the truthful network configuration (Al-Shaer et al., 2019)). Masking is a form of camouflaging the systems' attributes to disguise valuable information. The goal of these techniques is to increase the attacker's time spent in planning and compromising the network. The major research challenge is to determine how to accomplish masking in order to minimize the expected losses from an attack.

Past research used game-theoretic solutions to design efficient masking algorithms (Schlenker et al., 2018; Wang & Zeng, 2018). Specifically, Schlenker et al. (2018) developed a zero-sum Stackelberg game intended to design an optimal association of systems' true configurations into observed configurations that minimize the utility of the adversary. This "Masking Strategy", is designed to optimize how the network will deceptively respond to the adversary's actions during reconnaissance. The optimal strategy was tested against synthetic "powerful" (i.e., who is fully aware of how the defender masks the information during reconnaissance) and naive adversaries (i.e., an adversary with a fixed set of preferences over the observed information), none of which exist in reality.

In this paper, we evaluate such optimal masking strategy against *human* adversaries. We also compare human performance in the optimal strategy against a random masking strategy. We evaluate these masking strategies in an experiment implemented on CyberVAN (Chadha et al., 2016), a realistic testbed that helps simulate the masking techniques in a virtual network. Our main contribution is to provide a first evaluation of such a masking algorithm, and initial base-level human performance in order to improve these algorithms in future work.

## MASKING STRATEGY

Schlenker et al. (2018) proposed an optimal defense masking strategy to be used during reconnaissance in order to respond to requests from the adversary. Their algorithm is based on a zero-sum Stackelberg game model, in which the defender configures the network with a deception strategy (i.e., how the systems should respond to scan queries from an attacker) and the attacker scans the network and chooses a system to attack based on the system's responses. The defense algorithm assumes the worst-case scenario for the attacker's response, i.e., considers the minimum utility that a particular deception strategy would yield, and consequently, aims to compute the strategy that maximizes such utility. The authors show that this problem is NP-hard and provide a mixed-integer linear program to compute the optimal solution (see (Schlenker et al., 2018) for details on their algorithm).

The essence of this masking strategy is in how systems' true configurations (TCs) are masked into observable configurations (OCs) under the feasibility constraints. Each system in the network has attributes: an operating system, port numbers, services hosted, service versions, etc., and an associated utility that defines how attractive this system is to the adversary. The defender masks the true features of the system with different observable features using Honeyd services. When the adversary attempts to gain information about every system on the network, via probes and scans, the adversary observes certain attributes, which constitute the observable configuration (OC) of the system.

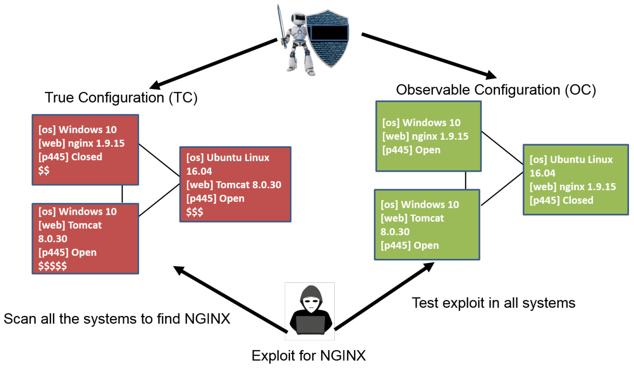

Figure 1: Masking Scenario

Figure 1 shows a masking scenario where the attacker is trying to exploit a NGINX webserver. If there was no masking in the network, the attacker would be able to scan all the machines and easily find the NGINX service. However, the defender may modify the observable features of the network nodes, so that when the adversary scans the machines, the features observed will be different than the real ones. Thus, in order to exploit the actual NGINX webserver, the adversary will need to use the exploit on all the machines. This will increase the time and effort the adversary spends during reconnaissance, and it will increase the chances for the defender to detect the irregularities of the activity.

The defender strategy determines how many of the systems having TCs should be assigned the OCs; that is, the strategy is represented by a matrix ($\Phi$) with the number of systems having TCs and OCs. We modified the optimal masking algorithm defined by (Schlenker et al., 2018) to include the decision to select the exploit in order to generate the $\Phi$ matrices used in our experiment. Given the defender strategy $\Phi$, if the attacker attacks using an exploit for TC $i$ on a machine masked with OC $j$, the attack is successful if the attacked machine is among the $\Phi_{ij}$ machines of TC $i$ masked by OC $j$. Since OC $j$ masks $\sum_i \Phi_{ij}$ machines in total, the success probability is $\frac{\Phi_{ij}}{\sum_i \Phi_{ij}}$ and consequently, the expected attacker utility is

$$u(\Phi, i, j) = \frac{\Phi_{ij}}{\sum_i \Phi_{ij}} v_i. \tag{1}$$

A rational attacker attacks a pair $(i, j)$ that maximizes the expected utility.

## EXPERIMENT

We conducted an exploratory experiment using the Cyber Virtual Ad hoc Network (CyberVAN) testbed (Chadha et al., 2016). CyberVAN is capable of creating network scenarios using virtual machines which could be controlled by either GUI or commands on a console. A major advantage of CyberVAN is that, it is possible to conduct human-in-the-loop studies for validating various defense algorithms. Specifically CyberVAN can be used to implement various deception tactics, including masking (e.g., by hiding/faking the configuration of nodes) and decoying (e.g., by using honeypots, honeynets, honeytokens, etc.). In this experiment, we used CyberVAN to implement masking strategies and test their effectiveness against human participants (in the role of attackers).

### Experimental Design

This experiment had 2 between-subjects conditions defined by the defense masking strategy: (1) the optimal masking and (2) random masking. The *Random Strategy* involved the random allocation of TCs to OCs and the *Optimal Strategy* strategy involved the generation of the $\Phi$ matrices according to the algorithm defined above, which minimizes the utility of the perfectly rational attacker and reduces the expected losses for defenders against rational attacker.

### Task

Using CyberVAN we manipulated the features of TCs using the $\Phi$ matrices produced by the optimal masking and random masking algorithms. TCs of virtual machines are masked to fake OCs using Honeyd service in CyberVAN. The Honeyd configuration file masks the operating systems and ports of TCs with OCs to fool the network scanning tools (Provos, 2003).

The task consists of 5 rounds (plus 1 practice round) and each round consists of 12 virtual machines. The configuration of virtual machines was different in each round. Each round consists of two phases: exploration phase and exploitation phase. During the exploration phase, participants probe the machines using *Nmap* command to obtain information of the open ports, operating systems, and running services. Participants are free to probe any machines in any order. The response from *Nmap* command may or may not represent the true configurations on the machine. Once participants explore the network and learn the features of network (where some of machines were masked accordingly), they decide which machine to attack and what type of exploit to use to conduct the attack. Each true configuration has a *utility* associated with it, which is in points currency (Table 1). The utility of TCs is the same across all 5 rounds. Participants earned the sum of the points accumulated across the 5 rounds, which were translated into dollars.

Table 1. Utility per True Configuration

| Configuration | Exploit | Utility |
|---|---|---|
| Slackware | slackware | 2 |
| Xbox | xbox | 2 |
| Ubuntu8 | ubuntu8 | 5 |
| Windows XP embedded | winxpemb | 6 |
| Avayagw | avayagw | 6 |
| Freebsd | freebsd | 8 |
| Windows XP | winxp | 8 |
| Windows 2008 | win2008 | 8 |
| Windows 2000 | win2k | 15 |
| Windows 7 Pro | win7pro | 15 |
| Windows 7 enterprise | win7ent | 15 |
| Openwrt | openwrt | 15 |

Participants were provided with a matrix Φ that describes the type and number of machines present in the network (TC) and their corresponding masked configuration (OC). Figure 2 presents an example of the matrix Φ in the Optimal and Random conditions. To help interpret the matrix, participants were given information regarding the way the TCs were mapped into OCs. For example, in the sample matrix for the Optimal condition, there are 6 TCs (avayagw, Ubuntu8, Win7pro, Win7ent, WinXP, Slackware) which are mapped to 3 OCs (freeBSD, Win7pro, and Ubuntu8). In the given matrix, 5 machines are shown as freebsd, out of which 3 are actually avayagw and 2 are Ubuntu8. Participants were allowed to use this information to calculate their probability of success and expected utility of attacking a particular machine.

**$\phi_1$ : (Optimal)**

$$\begin{bmatrix} TC\backslash OC & freeBSD & win7pro & Ubuntu8 \\ \text{avayagw} & 3 & 0 & 0 \\ \text{Ubuntu8} & 2 & 0 & 0 \\ win7pro & 0 & 2 & 0 \\ win7ent & 0 & 2 & 0 \\ winXP & 0 & 2 & 0 \\ Slackware & 0 & 0 & 1 \end{bmatrix}$$

**$\phi_1$ : (Random)**

$$\begin{bmatrix} TC\backslash OC & freeBSD & Win2008 & WinXP & Ubuntu8 \\ \text{avayagw} & 3 & 0 & 0 & 0 \\ \text{Ubuntu8} & 1 & 0 & 0 & 1 \\ Win7pro & 0 & 2 & 0 & 0 \\ Win7ent & 0 & 1 & 1 & 0 \\ WinXP & 0 & 1 & 1 & 0 \\ Slackware & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 2: Φ Matrices for Optimal and Random Conditions

## Participants

Participants were randomly assigned to one of the two experimental conditions: Random and Optimal. We collected 30 participants (15 in each condition) (57% male, Age Mean: 27, SD: 8). About 57% reported having or pursuing a degree in cybersecurity, 23% bachelor's degree in computer science, 13% other STEM degree, 3% a degree in electrical engineering and 3% reported other form of education. Only 20% participants had no practical cybersecurity experience.

After the successful completion of the experiment, all participants were paid the base payment of \$18. In addition, for each successful exploit participants received 1 point, which accumulated and were converted to a monetary bonus (conversion rate of 10 points = \$1). Participants could earn up to \$7.5 based on their accurate detection. The average time taken to complete this experiment was 65 minutes.

## Procedure

Participants were recruited through advertisement in email groups of cybersecurity researchers from Carnegie Mellon University. To be qualified to participate, participants were required to pass an online test of basic cybersecurity knowledge, which included general knowledge questions such as the definition of honeypots, SQL Injection attack, firewall, and protocols such as TELNET or SMTP.

Qualified participants were scheduled to come to a laboratory for individualized sessions of 90 minutes. First, participants provided informed consent, then they were asked to watch a video with instructions regarding the goal of the test and the general procedure. Participants were also provided with text instructions. Instructions were followed with a brief quiz to evaluate their comprehension of the instructions. They received feedback if they incorrectly answered a question in the quiz. Once all their questions were clarified, they were allowed to proceed with the experiment.

Participants were informed that the experiment would take up to 90 minutes and consisted of six rounds. They were also provided with a cheat sheet that listed all the commands required in the task, round wise Φ matrices, and the utility table. In each round participants were asked to probe the machines using the command "nmap -O 192.168.0.100" to retrieve information about open ports and operating systems on this IP address. They were also allowed to scan a batch of IP addresses together using an nmap command like, "nmap -O 192.168.0.100-111". After probing the machines, participants were allowed to calculate the likelihoods of the true configurations of the machines by looking at the utilities of each of the configurations and the Φ matrices given. Next, using the attack script, they decided what IP addresses to attack by selecting an appropriate exploit. Participants received points if the exploit used matched with the true configuration; otherwise, they received zero points. Once they finished all the rounds, participants were asked to answer to a demographic questionnaire and asked for their feedback regarding the experiment.

## RESULTS

During the exploration phase, we observed that a majority of the participants (29 out of 30) scanned all the computers before launching an attack. Participants chose to scan by providing the full range of IP addresses available to them using the Nmap command. Thus, we only analyzed the exploit actions in the 5 real real rounds.

### Success Rate

We calculate the success rate, i.e., out of 5 rounds, on how many rounds participants used the correct exploit. Figure 3(left) shows the average success rate for the Optimal and Random conditions. Surprisingly, attackers appeared to be more successful when pit against the Optimal algorithm compared to Random condition (suggesting a loss for the defender). However, this difference was not statistically significant, $F(1, 28) = 2.8$, $p$=0.10). The success rate varied within rounds as shown in Figure 4(top). Participants appeared to be more successful in the Optimal condition compared to the Random in rounds 2 and 4. However, although there was a significant effect of the round, ($F(4, 112) = 4.45$, $p<0.05$), there was no significant interaction between condition and round, ($F(4, 112) = 1.37$, $p$=0.24).

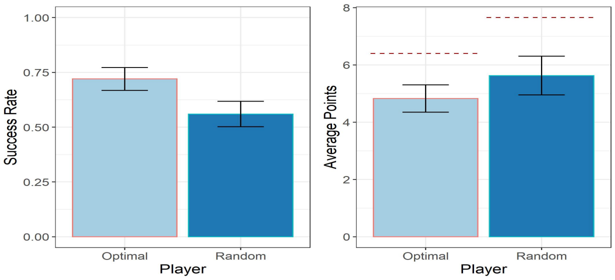

Figure 3: Success Rate (left) and Average Points (right)

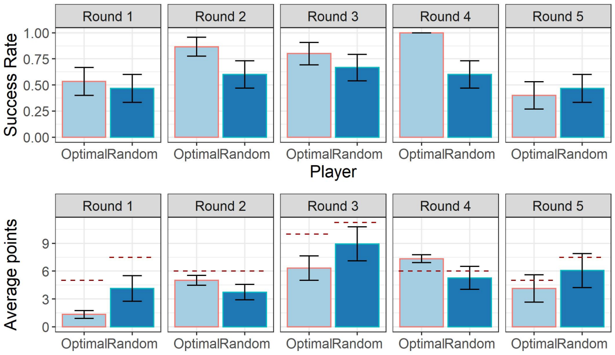

Figure 4: Round wise success rate (top) and average points (bottom)

### Average points

The average points per condition are shown in Figure 3(right). Interestingly, although attackers in the Optimal condition appeared to be more successful, they also obtained lower rewards compared to attackers in the Random condition. Again, this is only a tendency observed, as the statistical test revealed no significant difference between the masking conditions, $F(1, 28) = 0.65$, $p$=0.43), and no interaction of condition and round, $F(4, 112) = 2.02$, $p$=0.10). However, again, the round was significant, $F(4, 112) = 5.35$, $p<0.001$). The dotted lines in Figure 3(right) represent the expected utility of a rational attacker averaged over all the rounds. Overall, human attackers earned less points in both conditions compared to what a rational attacker would earn.

Figure 4(bottom) shows the average points per round. Participants earned more points in the optimal compared to the random condition in rounds 2 and 4; and except in round 4, human attackers earned less points in both conditions, compared to what a rational attacker would earn.

### Distribution of Attacks on True Configurations

To investigate the underlying reasons for the observed human behavior, we looked at the distribution of attacks on TCs in each round, according to their probability of success and their expected utility. Figure 5 presents the attack distributions on the TC machines, sorted by their expected utility (left-most bar in each round is the TC machine with the lowest expected utility and the right-most bar in each round is the machine with the highest expected utility). The payoffs and probability of success for each TC machine are shown on the top of the bars (payoff, probability).

We observe that frequently attacked TC machines were not necessarily those with the maximum expected expected utility. For example, in round 1 of the Optimal condition, the most commonly attacked TC machine was the one with the **lowest** expected utility. In other rounds the expected utility did not seem to matter significantly. For example, round 5 of the Optimal algorithm and round 1 of the Random algorithm, do not show any remarkable preference. The pattern emerging from this analysis is that, in many cases, participants preferred to attack TC machines that had higher probability of success even if the outcome of those machines was lower than the outcomes of other machines. For example, in round 1 of the optimal condition, most participants chose to attack on "slackware" machine which has a success probability of 1, but the lowest payoff (i.e., 2 points) among all the machines. In round 2, 3, and round 4 also attackers chose to attack the TC machines with high probability of success. A similar pattern in which participants tend to attack the machine with the highest probability of success is observed in the Random condition.

## DISCUSSION

To our knowledge, this is the first formal study conducted to verify the potential effectiveness of masking as a deception technique against human attackers in cybersecurity situations. A proposed masking algorithm of defense was compared to a random masking mapping. We observe that the Optimal algorithm tends to lead to more successful attacks but also to lower attacker's rewards compared to the Random algorithm. Also, generally human attackers' rewards are lower than the expected rational attackers' rewards.

A more detailed analysis of the attack decisions revealed that participants acted in agreement to a *certainty bias* (Baron et al., 1988), or risk aversion, as they tried

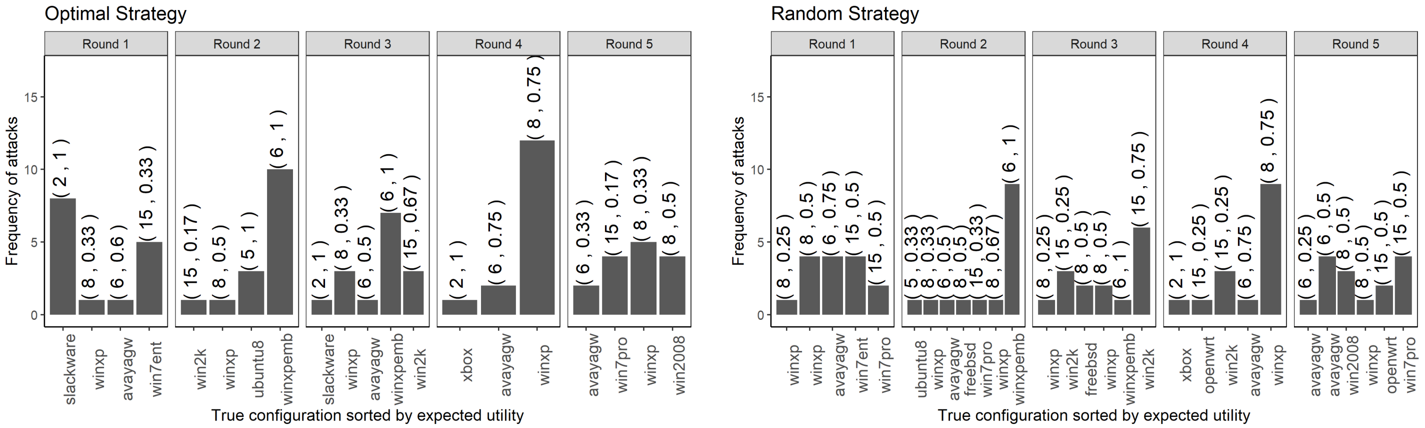

Figure 5: Attack Distribution on True Configuration: Sorted by Expected Utility

to attack machines where the probability of success was high, even when the potential reward was low. This concrete observation from our action data was supported by a post-experiment questionnaire where participants mentioned that they calculated probabilities before launching an attack.

Although the current study provides interesting insights, more research is required to generalize these observations to real-world cyberdefense applications. Realistic cyberdefense situations are extremely complicated, and often give more time to attackers for the reconnaissance. In this study the scenario is an oversimplification of realistic situations, and it relied on some unrealistic assumptions. For example, that the attackers had information about the mapping of true configurations to observable configurations of the machines, which real attackers are unlikely to have; we gave participants only 90 minutes to complete the task with a limited set of tools, and they could not use advanced commands and tools to verify if the Nmap output was correct or not. However, as it is expected from laboratory experiments, the real benefit is that we can study the cause-effect relationships between the treatment (e.g., masking algorithm) and the human behavior.

Our results provide some guidance to improve current state-of-the-art masking algorithms of cyberdefense. The masking strategies may be more effective by considering the risk-aversion tendency found in this study. We will also continue elaborating in this experimental approach by running laboratory experiments that address the various limitations of the current study. In the near future, we plan to develop a cognitive model that replicates attacker's behavior in this study. Using Instance Based Learning Theory (Gonzalez et al., 2003), we can generate a computational representation of attacker's decisions. Such model, could be used to generate large amounts of synthetic data, which could be useful to test the new masking strategies, and offset the costs of human experimentation in such complex scenarios.

## ACKNOWLEDGEMENTS

This research was sponsored by the Army Research Office and accomplished under MURI Grant Number W911NF-17-1-0370 and by the Army Research Laboratory under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The authors thank Jeffrey Flagg from the DDMLab at Carnegie Mellon University and Perspecta Labs for their help in conducting this experiment.

## REFERENCES

Aggarwal, P., Gonzalez, C., & Dutt, V. (2016). Cyber-security: role of deception in cyber-attack detection. In *Advances in human factors in cybersecurity* (pp. 85–96). Springer.

Al-Shaer, E., Wei, J., Kevin, W., & Wang, C. (2019). *Autonomous cyber deception.* Springer.

Baron, J., Beattie, J., & Hershey, J. C. (1988). Heuristics and biases in diagnostic reasoning: Ii. congruence, information, and certainty. *Organizational Behavior and Human Decision Processes*, *42*(1), 88–110.

Chadha, R., Bowen, T., Chiang, C.-Y. J., Gottlieb, Y. M., Poylisher, A., Sapello, A., . . . others (2016). Cybervan: A cyber security virtual assured network testbed. In *Milcom 2016-2016 ieee military communications conference* (pp. 1125–1130).

Ferguson-Walter, K. J., LaFon, D. S., & Shade, T. (2017). Friend or faux: deception for cyber defense. *Journal of Information Warfare*, *16*(2), 28–42.

Gonzalez, C., Aggarwal, P., Lebiere, C., & Cranford, E. (2020). Design of dynamic and personalized deception: A research framework and new insights. In *Proceedings of the 53rd hawaii international conference on system sciences.*

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, *27*(4), 591–635.

Griffith, S. B. (1963). *Sun tzu: The art of war* (Vol. 39). Oxford University Press London.

Provos, N. (2003). Honeyd-a virtual honeypot daemon. In *10th dfn-cert workshop, hamburg, germany* (Vol. 2, p. 4).

Schlenker, A., Thakoor, O., Xu, H., Fang, F., Tambe, M., Tran-Thanh, L., . . . Vorobeychik, Y. (2018). Deceiving cyber adversaries: A game theoretic approach. In *Proceedings of the 17th international conference on autonomous agents and multiagent systems* (pp. 892–900).

Spitzner, L. (2003). *Honeypots: tracking hackers* (Vol. 1). Addison-Wesley Reading.

Wang, W., & Zeng, B. (2018). A two-stage deception game for network defense. In *International conference on decision and game theory for security* (pp. 569–582).