The Moonraker Study: An Experimental Evaluation of Host-Based Deception

Temmie B. Shade, Andrew V. Rogers
Kimberly J. Ferguson-Walter
Laboratory for Advanced Cybersecurity Research
tbshade@tycho.ncsc.mil

Sara Beth Elson, Daniel K. Fayette
Kristin E. Heckman
The MITRE Corporation
selson@mitre.org

Abstract

Cyber deception has been discussed as providing enhanced cyber defense. This human subjects research, one of the first rigorously controlled studies on this topic, found that host-based deception was effective at preventing completion of a specific exfiltration task against a virtual network. In addition to impeding progress and preventing success, the deception resulted in increased confusion and surprise in the participants. This study provided the necessary rigor to scientifically attest to the effectiveness of cyber deception for cyber defense with computer specialists.

1. Introduction

Traditional network defense practices are proving to be increasingly ineffective at stopping the relentless and innovative offensive maneuvers of cyber attackers. Cyber deception is a growing part of the defender's arsenal aiming to slow down or prevent compromise by introducing confusion, frustration, or other psychological effects to the cyber attackers themselves. Most research on cyber deception has been on honeypots [1] or honeynets [2], decoy documents [3], or decoy network nodes [4]. However, there are many additional avenues defenders may take to muddle opponents and grant their systems an air of uncertainty.

In our study, we devised a novel approach of deceiving network intruders and measuring the effects on their campaign in terms of both success and cognition in an experiment. We have utilized a tool called Moonraker to intercept specific commands and react in deceptive ways. The experiment was conducted as a technical class teaching red team methodologies. Unbeknownst to the participants, Moonraker was utilized in the exercise portion of the class for half of the participants, configured to intercept the primary commands needed to complete the task. Completion required participants to enumerate a network's hosts, connect to one of the hosts, copy a malicious file to the host, execute the file, and retrieve its output. A post-exercise survey was given to capture participants' feelings and feedback on several fronts, including: emotional experiences (doubt, confusion, and frustration), past expertise in relevant technical skills, questions about their suspicion of the use of deception in the exercise, and, to further reinforce the cover story of the experiment being a training class, questions about the instructive material and exercise.

We hypothesized that the added deceptive actions would impede attacker progress and create a more frustrating and time-consuming attacker experience. While other researchers have made similar claims [5] [6], this is one of the first rigorously controlled experiments to examine the effectiveness of deception for cyber defense. To test our hypothesis, there were several metrics that were computed, including success on task, command success ratios, and self-reported emotions from the survey. It is generally not feasible to collect some of these metrics when an adversary is operating on a network, nor is it possible to control the environment as needed to attribute the cause to the experimental manipulation. As such, this research study provides a valuable contribution–scientific validation of the efficacy of cyber deception for defense.

2. Related Work

A variety of cyber deception techniques have been developed to thwart attackers, such as honeypots [1] and decoys [4]. Over the past several years, researchers have sought to determine the effectiveness of deceptive defenses by conducting studies with human participants. These studies have primarily focused on determining the realism of deception, measuring the difference in time on deceptive versus real assets, and assessing the abilities of deceptive techniques to detect attackers. Sample sizes were often small and most did not employ control conditions for comparison [6], thus they lacked

DISTRIBUTION A. Approved by AFRL for public release: distribution unlimited. Case Number 88ABW-2019-2863.

URI: https://hdl.handle.net/10125/63970
978-0-9981331-3-3

the necessary rigor to determine causative effects of the deception.

Other studies have entailed examining deception theoretically using game theory without human participants. These studies took a game theoretic approach, which provided a quantitative framework for reasoning about decisions given in scenarios where the players are either unaware or uncertain about the intent of opposing players [7].

One game theory study presented a new framework for autonomous cyber deception games in which each of the game's agents had their own perception of the game being played and the moves being taken [7]. In this framework, an agent may manipulate other players' perceived payoffs to convince them to take sub-optimal actions. The primary contribution of this work was a model of defensive cyber deception in which defender agents control attacker agents' perceptions of the cyber environment. The ultimate goal consisted of informing future cyber defense systems, enabling more sophisticated responses to attackers' behaviors, and improving defensive posture.

Another way to thwart attackers consists of exploiting their tendency to trust computer systems to tell them the truth [8]. Rowe provided a theory for defensive deception planning designed to encourage attackers to leave by convincing them that their plans could not succeed. Their theory included elements related to when and how to deceive attackers as well as how to monitor the attacker's acceptance of the deception. He closed the article by stating the need to test against human attackers. In a separate line of work, Rowe and colleagues conducted experiments with human participants to determine how much distortion they could detect in text [9]. They found that participants could detect text manipulations to a significant degree, but not perfectly. They concluded that randomly modified sentences can be detected despite a lack of familiarity with the subject matter, even though people found it difficult, in general, to detect deception. However, in a second experiment, participants could not detect fake directories. In this case, the context was sufficiently lacking, and context is a key resource in detecting deception. Along similar lines, Karuna et al. [10] presented a new framework which created false documents that were intentionally difficult to understand. They had participants answer comprehension questions after reading the documents and found statistically significant effects of the technique. Shu and Yan [11] asserted that one of the essential elements of deception is ensuring an internally consistent environment. They developed a technique to create an FTP file system free of inconsistency and asked students to determine whether or not the environment they were attacking was deceptive. The results revealed that the students underestimated the presence of deception.

Jafarian and colleagues [12] explored additional ways of interfering with attackers' abilities to conduct reconnaissance by designing a multi-dimensional deceptive technique to interfere with it. They demonstrated the effectiveness of this process by significantly increasing the workload for six expert participants. The Air Force Institute of Technology demonstrated the usefulness of manipulating system traffic to deceive an attacker's operating system (OS) fingerprinting as part of their network scanning efforts [13]. Since determination of the OS drives so much of the strategy in formulating attack vectors, Murphy found misleading network profilers to be a strong tactic, albeit one that could use some strengthening in the form of more developed tools and experienced configuration.

Lastly, in a set of experiments, Ferguson-Walter and colleagues [14] demonstrated that deceptive techniques, such as decoy systems, can increase attacker uncertainty regarding what is real and what is not when assessing a network. Their findings also showed that decoy systems distracted attackers from real assets and content, as well as slowed attacker behavior and disrupted their progress. Following up on these findings, Ferguson-Walter and colleagues [4] conducted a larger-scale study including over 130 red teamers to understand how defensive deception, both cyber and psychological, affects cyber attackers. To date, this (Tularosa study) is the largest study utilizing a professional red team population. The red teamers took part in a network penetration task over two days in which the researchers controlled both the presence of and explicit mention of deceptive defensive techniques. In addition to collecting extensive cyber data, the study collected psychological and physiological data to assess attackers' reactions to the exercise. While there are similarities, the Moonraker study was designed to determine the effectiveness of deceptive responses to attacker commands, thus concentrating on host-based deception; while the Tularosa study focused on network deception. In addition, thie Moonraker study examined the feasibility of using computer specialists rather than red teamers as participants for such experiments.

3. Design

3.1. Conditions

We examined the effectiveness of cyber deception at degrading network attacks and affecting attackers'

cognitive processes. The experiment varied the conditions of a hands-on cyber attack task. Participants were randomly assigned to one of two conditions in a between-subjects design: the Deception condition, in which unwitting participants encountered deceptive responses to commands, or the Control condition, in which they encountered no deceptive responses.

Participants worked individually to attack a virtual network and were instructed to operate with the mindset of a red teamer. Due to the use of generic computer specialists rather than professional red teamers, training was provided to introduce or review commands that would be needed in the task and help them adopt an adversarial mindset. The task-specific techniques were intermixed with common techniques not specifically needed for completion of the task. In order to introduce the prerequisite training, participants were told that they were evaluating the contents of a training video designed to teach cyber red teaming skills. To help gauge the effectiveness of the training, a hands-on network attack exercise using the skills learned followed the video. This cover story was used during the participant recruitment process and supported throughout the course of the seven-month, ten session study.

3.2. Cyber Range

For the hands-on exercise, each participant was presented with an identical view of the environment. The network was designed to be a small cluster of workstations running simulated SCADA software. The network setup (including hostnames, IP addresses, and MAC addresses) was identical in all environments, though between the Control and Deception conditions the machines behind some network nodes had key differences. The deceptive responses were provided by a capability called Moonraker, a product of the Air Force Research Laboratory's Firestarter Program, which was adapted for use in this study.

In the Control condition there were five Windows workstations and a single Linux system. The Linux system was used to isolate traffic within the environment from the rest of the network by only allowing specific connections into and out of the environment. Outbound connections were limited to the command and control channel required for the Moonraker client to receive commands from and provide logs to the Moonraker server. Inbound connections were limited to Remote Desktop Protocol (RDP) from the corporate network to an individual system which acted as the foothold into the experiment network. Of the five Windows systems, one was set up to be the participant's initial RDP system (the environment acting as the first foothold on the network they're attacking) and four were other remote systems on the network.

For the Deception condition, only three Windows workstations and a single Linux machine were present. The Linux system and participants' foothold environments were configured the same in both conditions. Moonraker intercepted calls to certain commands that revealed available hosts on the network and added on two decoys to the results. It additionally intercepted specific commands destined to these decoys and returned results as if they were valid machines on the network. Moonraker further manipulated the input to and results of certain commands targeted to the true remote Windows machines to provide additional deception for the experiment. In the remainder of this paper, these machines are referred to as responsive hosts to differentiate them from both the decoys and the real hosts from the Control condition. Usernames and passwords for all Windows systems were provided to participants. The username was the same on all systems; however the password was different between the initial RDP system (their first foothold environment) and the other Windows systems on the network. This was done in order to prevent Windows from using the local credentials when issuing commands requiring authentication. In this way, participants were required to include passwords for such authenticated commands, emulating normal attacker experiences during their campaigns.

Since the systems in use were basic Microsoft Windows installations and not actual SCADA systems, we designed a set of services to be the target of attacks. A pair of services was installed on each of the real workstations hosts that were not the initial RDP system. Since the services were not installed on the starting system, participants were required to move laterally to a second system to be successful in their attack. These services did not perform any additional function.

Each environment was prevented from accessing the internet or other corporate resources in order to reduce potential confounding variables. This also prevented the use of external tools to bypass the deception techniques of Moonraker. Participants were required to use built-in Windows command-line-only tools that were found in a default installation. Two exceptions to this rule existed: participants were instructed not to use PowerShell, and both *xcopy* and *move* were administratively disabled to prevent participants from using tools for which Moonraker manipulations were unavailable.

3.3. Moonraker

Moonraker provided a framework for monitoring and actively manipulating memory and processes. It can be customized to remotely hook and intercept system application programming interface and internal function calls, providing the ability to inject code, observe, and gain control of memory and processes running on a system. Moonraker was used in this study to manipulate behavior of participants' commands to implement deceptive responses. For this study, Moonraker was adapted to remotely intercept a specific set of commands that participants were most likely to use to execute six specific tactics, techniques, and procedures (TTPs) listed in the ATT&CK^{TM} framework [15] to successfully execute their objective. For all commands, the Control condition responded normally. Each TTP is outlined below, followed by an explanation of the response in the Deceptive condition. There are two types of hosts in the Deception condition: decoys and responsive hosts. Decoy hosts are false hosts that, when selected by participants, do not allow progress beyond TTP2. Responsive hosts are responsive to all TTPs, but the commands are intercepted by Moonraker and a deceptive response is given for certain predefined commands.

TTP1: Local network enumeration via the *net view* command. In the Deception condition, two of the hosts (the first and last in the list) were decoys thus obfuscating the true configuration of the network. A target system is selected.

TTP2: Connect from their local host to an admin share on the host they selected to target in the previous TTP. In the Deception condition, if the participant selected a responsive host from the list of four possible targets, the participants' commands produced results as expected. If participants selected a decoy host, the *net use* command was intercepted by Moonraker and reported as success when, in fact, it failed. Participants were unable to successfully complete the next step in the TTP sequence on decoy hosts. The only way forward was to return to TTP1 to restart the attack and select a responsive host.

TTP3: Copy an executable file ($process\_dump.exe$) to the system chosen in the previous TTP. In the Deception condition, various Windows commands were intercepted and a deceptive response was given. In instances where the participant chose a decoy host during TTP2, their attempts to copy this file resulted in standard Windows errors stating that the drive or path was not valid. This forced participants to go back to prior TTPs and reattempt their commands which continued to report success. When a responsive host was selected in TTP2, the first copy command submitted resulted in copying a *different* executable file but retaining the filename specified by the participant. This was often undetected until later in the attack sequence, typically discovered by the observing that the executable did not create an output file as it should.

TTP4: Schedule a task to run the executable file copied in the previous step to generate the SCADA process list. In the Deception condition, the *schtasks* command was manipulated using Moonraker to schedule the task five minutes later than the participant intended. The intent of this deception was to waste additional time (above the five minute delay) in activities such as debugging to determine why no output had been produced, querying the list of scheduled tasks, or re-executing TTP4.

TTP5: Stage the process list file for exfiltration. Participants were required to copy the process list file from the remote host to their local host. The Windows *copy* and *type* commands were manipulated to implement the deceptive response, corrupting the file. Participants who checked the file contents after copying the file would be made aware of the corruption and could remedy it through repeated attempts to copy the file. Those who did not check the contents of the copied file were unaware of the manipulation, positioning themselves for failure in their task to exfiltrate the process list.

TTP6: Exfiltrate the process list file to a designated server. Participants were reminded to use the pre-staged *pscp*, the Putty Secure Copy client tool, in order to simplify the exfiltration of the data. No deception was introduced in this step.

3.4. Cyber Data

In order to capture the full set of data generated by participants and the systems within the environments, four types of log collection were enabled. Since traditional key loggers only capture keystrokes, a custom key logger was developed that also recorded the date and time of the first key press as well as when the ENTER key was pressed. This allowed us to record the time between commands and the time a participant spent typing a command. The command prompt was modified in order to record the commands typed by the participant and the text response displayed to the participant as a result of that command. Two additional data sources were used as backups. Moonraker's internal logging mechanism recorded the commands it received at the server and video recording software captured a recording of the participants' desktops during the experiment.

Several timestamps were collected to track subject activity. These were used to calculate how long a subject took to complete a command and to start the next command. Each of the task's six TTPs had a list of commands that were defined as commands that could be used to successfully complete that part of the task. These TTP commands were the primary means of gauging subject progress and used during data coding to determine whether the command was successful in completing the specific TTP's task or not. Depending on the command, specific flags or arguments were required for success to be recorded. Additional flags or arguments did not prevent a label of successful if it still performed its noted TTP task. Typographical errors, either in the command itself or any of its required arguments, caused a label of unsuccessful. False success reported to subjects in some deception responses was also labeled as unsuccessful, as the subject had not truly completed the TTP.

Using the TTP commands alongside their noted success or failure, we tracked TTP progression in two ways: overall progress and on a per-host basis. Once a given TTP was successfully completed, the overall progress was updated to note the transition to the next TTP, marking anything past TTP6 completion as *post-success*. As subjects were able to switch targeted hosts, the per-host progress tracked completion of TTPs in a similar way for each given host.

3.5. Individual Measures

Participants completed surveys after training and after the exercise to gather demographic data, personality characteristics, and self-reported reactions to the task. Prior to the exercise, participants were asked to describe their experience in system administration, network defense, red teaming, and their appraisal of the training video. Following the exercise, personality was assessed using the BFI-44 [16] to assess their openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism. This personality assessment has been used in previous studies of defensive deception [4]. A post-exercise survey collected participants' self-reports of thoughts, actions, and emotional reactions during the task on Likert scales.

4. Implementation

4.1. Participants

The participants were selected based on a combination of relevant self-identified skills such as red teaming, cyber operations, and systems administration. Advanced adversaries often maliciously utilize legitimate system administration tools as a way of remaining undetected while illicitly accessing systems. The intent of recruiting participants with these experiences was to expand the pool of potential participants to include those who, with cyber red team training, could apply their skills successfully in a cyber context.

Potential participants were required to complete a prescreen test designed to identify those with the baseline technical skills for the hands-on attack exercise component of the study. The prescreen test included questions with answers collected via free-form text which allowed us to assess potential participants' levels of familiarity with commands and their ability to situationally apply concepts that would be necessary to successfully complete the task. Fifty-nine employees of a mid-sized, east coast technical company participated in the experiment.

4.2. Procedures

Prior to data collection, the study design was approved by the appropriate Internal Review Board to ensure all necessary human participant protections were in place. Study sessions were conducted classroom-style with one condition per session and five to thirteen randomly-assigned participants and a team of three to four proctors. Upon arrival, the lead proctor reiterated the cover story and introduced the participants to the proctors, each of whom was assigned a subset of participants to assist during the study and administer participant consent.

Study sessions had two main components consistent with the study's cover story: training and a hands-on cyber attack exercise. During the training component, participants watched a 1.5-hour red team training video and were provided with handouts for later reference. After the video, participants completed a post-training survey that included the Big Five personality inventory, questions about their technical background, and questions about the understandability of the training topics (in support of the cover story). During the 2.5-hour hands-on attack exercise, participants were provided with instructions and told to work individually to attack a network. They were given initial access to allow them to focus on post-exploit TTPs. Their primary objective was to find a host that was operating SCADA-related software and exfiltrate a zip archive containing two files: the process list from the host running SCADA software and the specific process name and process ID of all processes on that host that were related to the SCADA control systems.

Following the exercise, participants completed a post-task survey which contained questions to gauge their thoughts and self-reported actions during the attack, to identify their emotional reactions to the attack, to understand prior professional experiences, and questions designed to further support the cover story. Participants were then debriefed and could leave.

All participants were provided with a small set of pre-staged tools and told there were two specific tools they would need to use: $process\_dump.exe$, which produced the process list from a host as output, and *pscp* for exfiltrating the zip file. Participants were instructed to not use comparable system tools in lieu of the specific pre-staged tools. Participants were told that to achieve their objective, they needed to adopt a red teamer mindset, use the pre-staged tools, and use any other approaches they may have learned from the training portion of the study. The TTPs required for success were detailed in Section 3.3. Participants were instructed to choose any host to attack, and, in the event of failure, they could choose to attack the same host again or a different host within the time allotted.

5. Sample Characteristics

Participants were asked to self-report their areas of experience and skill level. The Control condition had 30 participants, while the Deception condition had 29. There were no significant differences between conditions on any of these questions. Of the participants, 73% had no experience with red teaming, 47% had never performed a penetration test, and 34% had no experience in network defense. The average years of experience in each of the previous activities is given in Figure 1 for all participants, followed by the number of participants who reported no experience and the average years of experience for those who listed any experience. While the lack of experience is a limitation of the experimental design, there were no significant correlations found between the experience factors and success on the cyber task. The strongest correlation with experience in network defense accounted for less than 6% ($r^2 = .058$) of the variance in success.

Participants were also asked to complete the BFI-44, a test of the Five Factor Personality model. The participants, when compared to a large normative sample [18], scored higher on Agreeableness and Conscientiousness and lower on Neuroticism. This is consistent with results from another cyber deception experiment using professional red teamers as participants [4]. In both cases, while the scores for the samples were slightly higher they remained within the average range for each scale (Figure 2).

6. Results

To test the major hypothesis, that deceptive responses will impede attacker progress, we compared the number of participants in the two conditions who were able to successfully complete the task assigned to determine whether or not the deceptive command line responses impeded progress. The participants were provided the following instruction: "To achieve your objective, you must exfiltrate a zip archive containing two files. The first is the process list from a host that is running SCADA software. The second is a file that you generate from analyzing the process list and identifying the process name and process ID (PID) of all processes on that host that are related to the SCADA control systems. All processes related to SCADA control systems in this exercise will have the word *scada* somewhere in the process name. For example, $log\_scada.exe$ would be a SCADA related process."

Success at this task was determined by examining the submitted zip archive for the requisite files. If the file contained the complete process list or a summary document with the SCADA-related files identified, as described earlier, the task was considered successful. The control condition was significantly more successful than the deception condition as indicated by a $\chi^2(1) = 7.03$, $p = 0.012$ (Figure 3).

Another measure of effectiveness was the proportion of TTP commands that were successful. The control and deception condition were compared on the proportion of TTP commands that were labeled successful. A significant difference was found between the two condition means ($t(57) = 1.40$, $p = 0.013$; Figure 4). Overall, 67% of the TTP commands submitted by the control condition were successful compared to 55% successful for the deception condition. This provides one measure of impeded progress in the attack scenario.

In addition to preventing a successful attack, deception can also impede progress of attackers through the system. To test this, we compared participants in both groups who successfully completed the task on the total number of minutes they took from start to end for the first success. Many participants actually completed the task on multiple hosts. The summary statistics for the conditions are provided in Figure 5. There was a significant difference ($t(26) = 2.29$, $p = 0.033$) between the two conditions. The participants in the Deception condition took on average 106 minutes while the participants in the Control condition took only 76 minutes. While a portion of this time can be accounted for by the additional steps needed for success in the Deception condition, the difference is larger than necessary to accomplish those steps. To account for

Characteristic	Average All	Amt. with No Experience	Average Experienced	Correlation with Success
Years of experience in Network Defense	5.3	20	8.0	r = 0.24 (ns)
Years of experience in Red Teaming	0.8	43	3.2	r = 0.14 (ns)
Number of times performed penetration test	2.3	28	4.4	r = 0.01 (ns)

Figure 1. Self-Reported Experience of Participants

	Moonraker			Normative Sample			Mean Difference		
BFI-44	Mean	SD	N	Mean	SD	N	Moonraker - Normed	t - value	Significance
Agreeableness	72.74	11.68	59	66.40	17.79	132515	6.34	4.1691	<.01
Conscientiousness	75.08	11.68	59	63.84	18.02	132515	11.24	7.3914	<.01
Neuroticism	34.78	17.55	59	51.02	21.34	132515	-16.23	-7.1076	<.001
Openness	74.85	10.17	59	74.51	16.29	132515	0.34	0.2568	ns
Extraversion	54.22	21.10	59	54.61	22.49	132515	-0.39	0.1528	ns

Figure 2. Five Factor Personality Scores of Participants

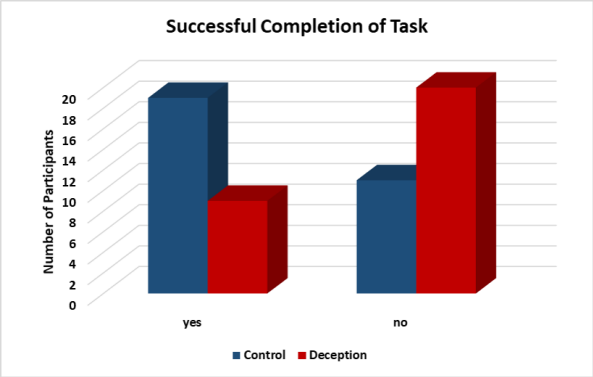

Figure 3. Participants performance on exfiltration task

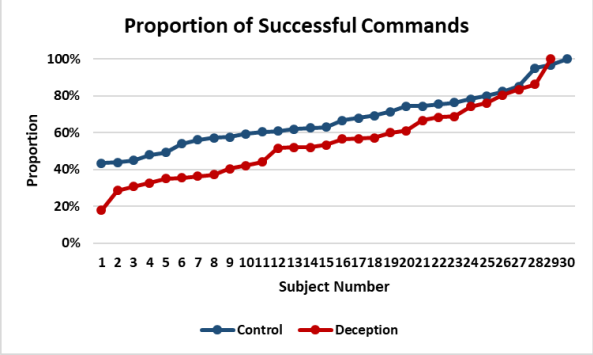

Figure 4. Proportion of successful commands

	Control	Deception	t value	p
Minimum	20.22	64.62		
Maximum	145.98	143.49		
Mean	**75.71**	**106.49**	2.29	0.033
St Dev	37.56	30.87		
N	19	9		

Figure 5. Time to Task Completion

the five-minute delay included in TTP4 in the deception condition, 5 minutes was subtracted from the time to complete TTP4 for each participant in the deceptive condition before calculating total time. The fastest time for the Deception participants was 64 minutes, more than three times the mean for Control participants.

As you can see in Figure 6, almost a third (9) of the Control participants completed the task faster than any of the Deception participants. The quickest completion rate for those who successfully completed the exfiltration for TTP6 the first time was 20 minutes for Control and 64 minutes for Deception. This provides additional evidence that the deceptive responses significantly impeded progress as measured by time necessary to attack the system. An additional seven of the participants in the Deception condition succeeded on TTP6, the last step, but did not do the additional steps necessary to successfully complete the task. This indicates they may have believed they actually succeeded when in fact they did not. Not only did the deceptive responses significantly impede progress

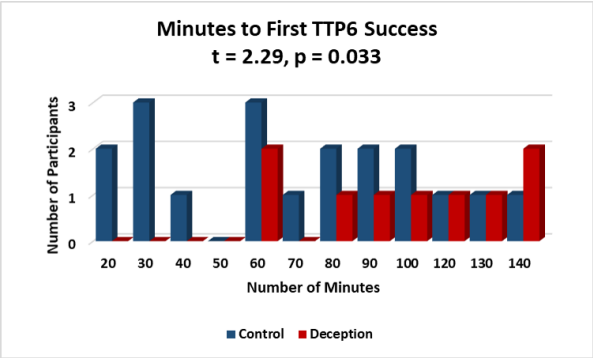

Figure 6. Total Time to Complete TTPs 1-6

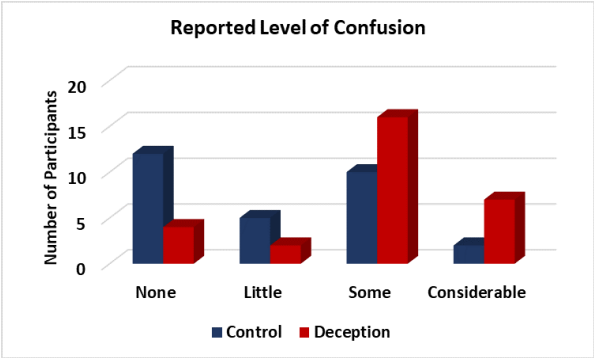

Figure 8. Reported Confusion.

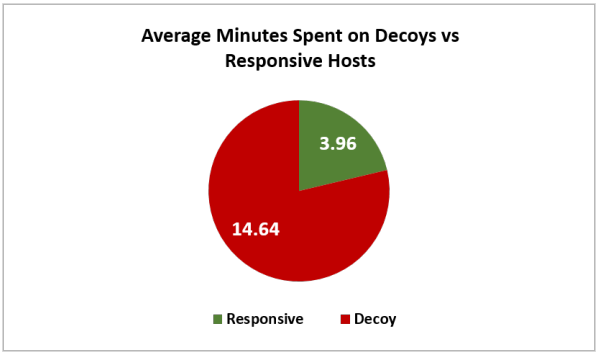

Figure 7. Time wasted on decoys

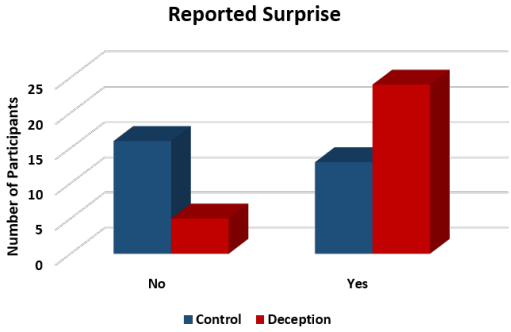

Figure 9. Reported Surprise

as measured by the time necessary to be successful but also provided a false sense of success for those in the Deception condition. Slowing progress of the attackers and instilling a false sense of success provide a significant defensive advantage.

For participants in the Deception condition, the total amount of time spent attempting TTP2 on decoys vs responsive hosts was calculated (See Figure 7). From the participants perspective, when the target selected is a decoy, TTP2 reports a deceptive success and TTP3 provides a failure message. This reflects the average amount of time continuing to attempt to progress on a decoy compared to the average amount of time to complete TTP2 on a Responsive Host. Participants spent significantly more time, 3.5 times the number of minutes, on decoys than on responsive hosts (paired samples $t(58) = 4.45$, $p = .00007$). This supports previous research that demonstrated participants tend to expend more resources on decoys [14]. We noted that overall 97% of participants started by targeting either the first or last IP numerically (which were decoys in the Deception condition); 43% progressed through the IPs in numeric order, starting either at the first or the last.

After completion of the cyber task, participants were asked to rate their feelings of frustration, confusion, doubt, and surprise related to their experience of the task. While there was no significant difference found between conditions on feelings of frustration or doubt, significant differences were noted between the control and the deception condition on the amount of confusion felt due to the task ($\chi^2(3) = 9.45$, $p = 0.024$) and whether or not participants were surprised by the task ($\chi^2(1) = 9.03$, $p = 0.003$). (Figures 8 and 9, respectively)

After the exercise, participants were asked if they thought the system had tried to deceive them. There was a significant difference in the answers between the deception and control condition ($\chi^2(1) = 7.28$, $p = 0.007$). Interestingly, less than half (41%) of those in the deceptive condition thought that deception was involved (Figure 10). This might reflect the operation of an innate cognitive bias, in which, confirmation bias influenced the attacker's perception that the network's representation was accurate and that their commands executed as expected, even after being presented with an alternative hypothesis in the form of the question asked. Previous researchers have discussed a similar effect in decision-making biases in the cyber domain [17] as well as how understanding these biases can aid in cyber

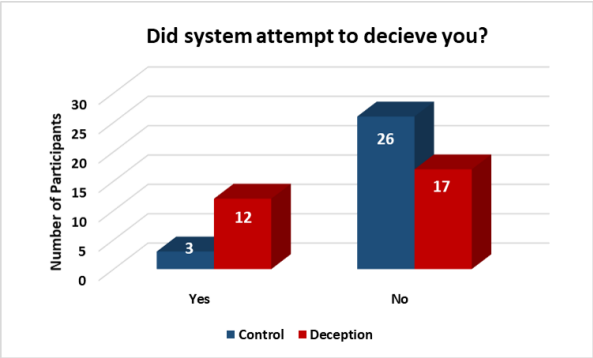

Figure 10. System attempt to deceive

defense [18].

To examine the extent to which our participants–computer specialists given minimal red team training–might resemble the broader population of interest, we examined the performance of the participants in the control condition. This condition did not encounter any deception and, therefore, their performance provides an estimate of the success rate for individuals who were not considered red teamers. Only 63% of the control condition was able to successfully complete the task after the training. Thus, it appears that individuals without specific experience related to the task at hand are not ideal participants for testing of cyber defensive deception because the success rates will be lowered due to lack of experience rather than the experimental treatment.

Given this difficulty with the task attributed to inexperience, we must question if some of the differences found were not attributed to deception but rather to inexperience. However, participants of differing skill levels were equally distributed between the conditions and the relation between experience and successful completion of the task was tested and found to be non-significant for all conditions as indicated in Figure 1. Therefore, it is likely that the effects noted are correctly attributed to deception.

It has previously been assumed that red teamers are a valid representation of adversaries [19]. However, the lack of availability and the expense of hiring them makes it difficult to use them for participants. As such, we used computer professionals and gave them red team training so that they could stand in for professional red teamers on a simple task. In the process, we discovered that individuals without specific experience related to the task at hand are not appropriate participants for testing of cyber defensive deception because the success rates will be lowered due to lack of experience rather than the experimental treatment. Given the reduction in success rates due to inexperience noted earlier, we concluded that to adequately test host or network defenses requires participants that have expert knowledge and experience with penetration testing. We consider this knowledge as a valuable lesson that can benefit future studies.

7. Discussion

7.1. Limitations

There were some limitations to the study design. Internal validity was limited by the deceptive methodology used which only provided deceptive responses to a limited number of predetermined commands. Participants could bypass deception by using commands that performed similar functions but were not on the bypass list. We mitigated this possibility by constraining the participants' command and program usage through detailed instructions and technical solutions. However this effects the external validity by artificially influencing the attack behavior.

External validity was limited due to the population from which the participants were drawn. While the ideal participant pool would have been individuals whose skills and experience more closely resembled cyber adversaries, it was decided to employ computer specialists who did not necessarily have such experience to give us a large enough pool of participants. However, even with these limitations, the results indicated that deception significantly degraded participants' performance.

Construct validity in any timing analysis was a concern due to the extra time forced by the responsive hosts in the deceptive condition. We addressed this by adjusting the data to account for the delay imposed in the deception condition. Results supported the hypothesis that in addition to impeding attack, deceptive responses can cause a critical delay (allowing time for cyber defenders to mitigate the threat).

7.2. Conclusions

To summarize, deception reduced participants' ability to attack a network and significantly slowed the progress of those who were able to attack it. Even given the difficulty that some participants in the control condition had in completing the task, the control condition performed significantly better than the deception condition in accomplishing the exfiltration objective. Host-based deception effectively impeded progress, prevented task completion and induced increased confusion and surprise in those attempting to exfiltrate targeted information. Slowing the progress of attackers and instilling a false sense of success provided a significant defensive advantage. As one participant in

the deception condition stated in the interview following the task, "[the] most challenging [part of the task] was I couldn't tell if it was me or the network."

Previous research [14] has found similar interference with task completion and effect on cognitive processes using network deception. This alteration of cognitions, i.e., increased confusion and surprise, may have a more persistent effect as it potentially influences future attacks or reduces the attacker's confidence in the validity of any information obtained. Defensive cyber deception appears to be a promising method to re-balance the asymmetry of cyber defense not only in delaying or impeding attacks but also in affecting the cognitive processes of the attacker. Very few published studies have utilized an experimental design that includes human experts. This research, one of the first of its kind, provided the necessary rigor to scientifically attest to the effectiveness of cyber deception for cyber defense with computer specialists.

Acknowledgments

The authors would like to thank all the professionals who volunteered to participate in this study and gratefully acknowledge the following people for the roles they played in the success of this research: Dr. Dana LaFon, Dr. Ahmad Ridley, Rick Van Tassel, Mary Berlage, Steve Danko, Adam Pennington, Frank Duff, Dr. Deanna Caputo, Dr. Nick Kohn, Cory Minter, Chris Alicea. We'd also like to thank Thomas Parisi and Dr. Edward "Paul" Ratazzi from the Air Force Research Laboratory as well as the Moonraker development team.

References

[1] N. Provos, "A Virtual Honeypot Framework," in *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, SSYM'04, (Berkeley, CA, USA), pp. 1–1, USENIX Association, 2004.

[2] L. Spitzner, "The honeynet project: Trapping the hackers," *IEEE Security and Privacy*, vol. 1, pp. 15–23, Mar. 2003.

[3] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting Inside Attackers Using Decoy Documents," in *Security and Privacy in Communication Networks* (Y. Chen, T. D. Dimitriou, and J. Zhou, eds.), Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 51–70, Springer Berlin Heidelberg, 2009.

[4] K. J. Ferguson-Walter, T. B. Shade, A. V. Rogers, E. M. Niedbala, M. C. Trumbo, K. Nauer, K. M. Divis, A. P. Jones, A. Combs, and R. G. Abbott, "The Tularosa Study: An Experimental Design and Implementation to Quantify the Effectiveness of Cyber Deception," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, p. 10, Jan. 2019.

[5] S. Achleitner, T. La Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, and R. Chadha, "Cyber deception: Virtual networks to defend insider reconnaissance," in *Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats*, MIST '16, (New York, NY, USA), pp. 57–68, ACM, 2016.

[6] X. Han, N. Kheir, and D. Balzarotti, "Deception techniques in computer security: A research perspective," *ACM Comput. Surv.*, vol. 51, pp. 80:1–80:36, July 2018.

[7] K. Ferguson-Walter, S. Fugate, J. Mauger, and M. Major, "Game theory for adaptive defensive cyber deception," in *Hot Topics in Science of Security (HoTSoS)*, (Nashville, Tennessee), ACM, Apr. 2019.

[8] N. C. Rowe, "Designing good deceptions in defense of information systems," in *20th Annual Computer Security Applications Conference*, pp. 418–427, Dec 2004.

[9] N. Rowe and J. Rrushi, *Introduction to Cyberdeception*. Springer International Publishing, 2016.

[10] P. Karuna, H. Purohit, R. Ganesan, and S. Jajodia, "Generating Hard to Comprehend Fake Documents for Defensive Cyber Deception," *IEEE Intelligent Systems*, vol. 33, pp. 16–25, 2018.

[11] Z. Shu and G. Yan, "Ensuring Deception Consistency for FTP Services Hardened Against Advanced Persistent Threats," in *Proceedings of the 5th ACM Workshop on Moving Target Defense*, MTD '18, (New York, NY, USA), pp. 69–79, ACM, 2018. event-place: Toronto, Canada.

[12] J. H. Jafarian, A. Niakanlahiji, E. Al-Shaer, and Q. Duan, "Multi-dimensional Host Identity Anonymization for Defeating Skilled Attackers," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, MTD '16, (New York, NY, USA), pp. 47–58, ACM, 2016. event-place: Vienna, Austria.

[13] S. B. Murphy, *Deceiving Adversary Network Scanning Efforts Using Host-Based Deception*. Air Force Institute of Technology, Defense Technical Information Center (DTIC), 2009.

[14] K. J. Ferguson-Walter, D. S. LaFon, and T. B. Shade, "Friend or Faux: Deception for Cyber Defense," *Journal of Information Warfare*, vol. 16, no. 2, pp. 28–42, 2017.

[15] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK^{TM}: Design and Philosophy," *MITRE Coperation Technical Report*, 2018.

[16] O. P. John and S. Srivastava, "The Big-Five trait taxonomy: History, measurement, and theoretical perspectives.," in *Handbook of Personality: Theory and Research*, vol. 2, pp. 102–138, New York, NY, USA: Guilford Press, l. a. pervin & o. p. john ed., 1999.

[17] R. S. Gutzwiller, K. J. Ferguson-Walter, and S. J. Fugate, "Are cyber attackers thinking fast and slow? Evidence for cognitive biases in red teamers reveals a method for disruption," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Oct. 2019.

[18] R. Gutzwiller, K. J. Ferguson-Walter, S. Fugate, and A. Rogers, "'Oh, Look, A butterfly!' A framework for distracting attackers to improve cyber defense," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Oct. 2018.