Chapter 39
HackIt: A Real-Time Simulation Tool for Studying Real-World Cyberattacks in the Laboratory

Palvi Aggarwal, Cleotilde Gonzalez, and Varun Dutt

Abstract Computer-based simulation tools have an important role to play in helping us understand the behavior of people performing as attackers (people who launch cyberattacks) and defenders (people who protects computer networks against cyberattacks) in complex cyber situations. In this paper, we introduce a simulation tool called HackIt that could be used to build dynamic cyberattack scenarios. We used the HackIt tool to investigate the influence of timing of deception strategies involving honeypots (computers that pretend to be real, but those that are actually fake) on the decisions of participants performing as attackers. In a lab-based experiment, participants performing as attackers were randomly assigned to two between-subjects conditions, each involving six repeated games: early ($N = 20$) and late ($N = 20$). In early condition, deception was present via honeypots on the second and third games, whereas in late condition, deception was present via honeypots on the fourth and fifth games. Presence of deception meant that the honeypots were easy to exploit in deception rounds. In both conditions, the goal of attacker was to steal credit-card information for computers on the network. Results revealed that the proportion of honeypot attacks were higher in late condition compared to early condition. Similarly, we found that the proportion of regular attacks were lower in late condition compared to early condition. We highlight the potential of using the HackIt tool for creating realistic cyberscenarios and evaluating the effectiveness of different deception strategies in reducing cyberattacks.

Keywords Deception · Honeypots · Cyberattacks · Cybersecurity tools · Deception strategies

P. Aggarwal (✉) · C. Gonzalez
Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: coty@cmu.edu

V. Dutt
Indian Institute of Technology Mandi, Mandi, India
e-mail: varun@iitmandi.ac.in

B. B. Gupta et al. (eds.), *Handbook of Computer Networks and Cyber Security*,
https://doi.org/10.1007/978-3-030-22277-2_39

1 Introduction

Cyberattacks are targeting government, industries, banking, and e-commerce business at an alarming rate. The cyber criminals use advanced cyberattacks, e.g., SQL injection, phishing, Trojans, ransomwares, rootkits, and malware, to breach the network and gain access to information [1]. The growing threat of cyberattacks on critical cyber organizations reveal the urgent need for finding methods that enhance network security. Presently, there are numerous cyber defense mechanisms available to fight against various cyberattacks, but very few of these solutions can prevent zero-day cyberattacks, attacks that target publicly known but still unpatched vulnerabilities [2]. Deception is an art of persuasion where defenders intentionally mislead the hackers to something which is not true [3], and it may provide a promising real-time solution against cyberattacks [4, 5]. Deception technology may help reduce cyberattacks by shifting the cognitive, economic, and time costs of cyberattacks back onto the attacker. The objective of this chapter is to investigate how deception tools may help reduce cyberattacks.

The principles of deception have been around for years, and recently, they have become a secret weapon of purple teams and threat hunters worldwide [4–6]. Deception may involve communication between a target and a deceiver, in which the deceiver tries to convince target to trust the fake explanation of ground truth [3]. Deception can be implemented with existing tools, e.g., firewalls with blacklists, intrusion prevention, and URL filtering [7]. Several researchers have proposed deception's use via honeypots [8–10]. Honeypots are servers that mimic real server, but they are actually fake [11].

Prior research has proposed a number of tools for using deception in cybersecurity. Some of these tools include the deception tool kit (DTK), honeynets, honeytokens, thug, TrapX, and smokescreen [8–11]. The DTK is one of the earliest tools that applied deception methods for cyber protection. Using TCP wrappers, DTK process the incoming malicious requests that are usually blocked in the network. Researchers have also proposed honeynets [11], which are a group of honeypots that are used for real-time applications. The client-side honeypots were proposed to detect malicious Web-based activities that try to breach client application vulnerabilities [12].

Furthermore, some researchers have proposed noncooperative dynamic deception games for understanding hackers' cyberattack decisions via experimentation and mathematical modeling [13–15]. For example, Aggarwal et al. [16] investigated the role of the amount of deception and the timing of deception in a cyberattack scenario using a noncooperative deception game (DG). The DG involved two phases: probe and attack. During probe, participants performing as hackers could probe webservers, some of which were honeypots. Once hackers probed the network, they could attack one of the webservers for real. Results revealed that using late timing of deception and high amount of deception helped in reducing cyberattacks.

The focus of the past research was on developing tools for deception and understanding human factors in abstract environments. The results of lab-based experiments as part of the past research were based on abstraction of choices and outcomes, which may not be realistic enough to capture how computers are attacked by hackers in the real world. The primary objective of this research is to develop a cybersecurity tool which helps us understand the human factors influencing hacker's decisions in complex cyber situations involving deception. To accomplish the above objective, we developed a HackIt tool, i.e., a tool to simulate complex cyber situations and showcase the potential of this tool via certain cyberattack scenarios involving deception.

Using HackIt tool, we simulate two deception scenarios where the timing of deception is either early or late. The goal of the hacker in these scenarios is to steal the credit-card information from the network. The network consists of two webservers where one of the webservers is a real webserver and the other one is a fake honeypot webserver. In this chapter, using the HackIt game, we investigate whether an early or late timing of deception is effective in trapping hackers into honeypots. Furthermore, we compare our results with the findings presented by Aggarwal et al. [16].

In what follows, we first discuss the functioning of the HackIt tool. Next, we explain the methods of an experiment run using the HackIt tool. Furthermore, we detail the results from HackIt tool and discuss the implications of our results for decision-making of hackers in the real world.

2 HackIt Tool

Hacking websites generally may involve two phases: searching for vulnerabilities (probing) and attacking computer workstations (attacking). HackIt is a cybersecurity tool that allows researchers to create various cyber situations and to map real-world cyberattack scenarios by involving two phases: probe phase and attack phase. The probe phase involves scanning of webservers in the network for vulnerabilities, whereas the attack phase involves gaining access to different computers and stealing information or compromising computer systems. For example, using the HackIt tool, one can create networks of different sizes, use deception and configure different webservers as honeypots, and create any number of fictitious ports, services, fake operating systems, and fake files on honeypots. The HackIt tool can run various network commands that include nmap, use_exploit, ls, and scp. Nmap is a network utility that shows the open ports, operating system, and services on the specified webserver. The nmap utility provides the list of vulnerabilities on the corresponding webservers. The use_exploit command exploits vulnerabilities of a system and helps hacker to gain access to a webserver. Next, the ls command lists the files currently on the file system of the machine. The scp command transfers files to the remote machine. The probe phase involves scanning of webservers in the network using nmap command. Using nmap command, hacker may collect information about the

open ports, services, and vulnerabilities available in the network. Furthermore, the attack phase involves exploiting vulnerabilities and stealing information. To exploit any webserver, the hacker may use use_exploit command and gain access of the webserver. Next, the hacker could steal information from computer systems using the scp command. In the next section, we perform an experiment with the HackIt tool, where we evaluated the effectiveness of timing of deception via honeypots on reducing cyberattacks.

3 Experiment

3.1 Experiment Design

In this chapter, we analyzed the effect of timing of deception on hacker's actions. The timing of deception was manipulated across two between-subjects conditions: early deception ($N = 20$) and late deception ($N = 20$). In both conditions, participants playing as hackers were given 6 game rounds in a sequence (end point unknown to participants), where 2-game rounds possessed deception. Figure 39.1 shows the experimental design of the deception game implemented in HackIt tool. In this experiment, if the timing of deception was early, then deception was present on the second and third rounds in the sequence. However, if the timing of deception was late, then deception was present in the fourth and fifth rounds in the sequence. Presence of deception meant that the honeypots were easy to exploit via popular ports and vulnerability in the deception rounds compared to the nondeception rounds (more details ahead in this chapter). However, participants were not told that honeypots will involve easy-to-attack configurations in deception rounds. Also, participants were not disclosed of the number of rounds and on which deception was involved. Participants were told that there would be repeatedly interacting in games with different websites containing two new computer systems. To analyze human data, we looked at the proportion of honeypot attacks and proportion of regular attacks at the attack stage by the hacker across six rounds in each condition.

Condition/ Rounds	1	2	3	4	5	6
Early Deception	-	D	D	-	-	-
Late Deception	-	-	-	D	D	-

Fig. 39.1 Experiment design using deception game with six rounds and two conditions, i.e., early deception and late deception. *D* deception present; - deception not present

3.2 HackIt Task

In this chapter, we simulated a network of two webservers in HackIt tool where the objective of the hacker was to steal real credit-card information located on one of these webservers. Figure 39.2 shows the step-by-step HackIt task procedure. Step 1 of Fig. 39.2 shows a simulated network with two webservers where one of the webservers acts as a honeypot webserver and another one acts as a real webserver. Table 39.1 shows the information about the configuration of easy-to-attack and difficult-to-attack systems. For example, a system with Windows XP operating system, port 80/tcp, and service http will be easily exploitable. However, a system with Linux operating system, port 22/tcp, and service ssh will be difficult to attack. In deception rounds, honeypot webservers were configured in such a way that hackers could easily attack them. However, the regular webserver was difficult to attack in these deception rounds. In the experiment, participants were informed about the easy-to-exploit and the difficult-to-exploit configurations in Table 39.1.

Next, the hacker probes the network using nmap command to gain information about both the webservers. For example, in step 2 and step 3, the hacker probes webserver 1 and webserver 2, respectively (Fig. 39.2). Probing both the webservers gave the information about the operating system, open ports, services, and vulnerabilities. For example, probing webserver 1 gave the information to the hacker that webserver 1 is running on Solaris operating system. The open ports on webserver 1 are 80, 135, 21, and 111, where SQL injection, DOS attack, brute force attack, and DDOS attack are possible. The information provided to the hacker as a result of probing the systems gave him an idea of the possible success of an attack on that system. Once the hacker collects information about open ports and services, he/she could attack a webserver by using the "use_exploit" command. The use_exploit command exploits vulnerabilities of a system and helps the hacker to gain access to that webserver. For example, in step 4, the hacker used the use_exploit command to gain access of webserver 1 using DoS_attack vulnerability. Once the hacker exploited a system and gained access, he/she lists all the files by using the "ls" command. After this command, the hacker transfers required file, i.e., "pin.txt," using the "scp" command (step 5, Fig. 39.2). Once the hacker copies the file from the exploited system, he/she is informed whether he/she was successful or not in stealing a real credit-card file from the computer. Once the task is complete, the hacker is given the textual feedback about the success or failure of his action to copy the real credit-card file (step 6, Fig. 39.2).

3.3 Participants

A total of 40 participants were recruited through an e-mail advertisement to participate in an online cybersecurity study conducted using HackIt tool. About 68% of participants were males. The age of participants ranged from 18 years to 32 years

```
Welcome Hacker    Your IP: 172.22.31.31
=====================================
Your aim for the game is to attack any one of the 2 Systems, gain entry into
it and steal the credit card information from there. If you successfully steal
credit card information, you will win the game otherwise you may lose points.

Enter "start" to start the game
System you need to hack is "System1", or "System2"
--------------------------------------------------------------------------------
Probe the network -> Attack the System -> Steal the information

>> Remaining Time : 981 seconds >>
>
```

Step 1: Initial instructions to the participants

```
Probe Phase
----------------
Welcome to the Probe Phase. Probe means that you try to collect information
on whether a system is vulnerable or not. To probe a system, you need to run
the nmap command on each system once.
Nmap is a network utility is designed to check for open ports, Operating
System and services on a network connected system. In this game, the nmap
utility will also provide the list of vulnerabilities and the Operating
System available on the corresponding systems.

The format of nmap is: nmap [system-name].
Example: nmap System1
> nmap System1

Starting Nmap 6.47 ( http://nmap.org )
Nmap scan report for System1
Host is up (0.000011s latency).
Not shown: 996 closed ports

            PORT              STATE SERVICE Vulnerabilities
80/tcp                        open  http    sql_injection
135/tcp                       open  msrpc   DoS_attack
21/tcp                        open  ftp     brute_force
111/tcp                       open  rpcbind DDoS_attack
Device type: general purpose
Running: Solaris
OS details: Solaris
Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
Now you can probe another system...
```

Step 2: Scanning the webserver 1 using nmap command

Fig. 39.2 HackIt game environment

```
> nmap System2

Starting Nmap 6.47 ( http://nmap.org )
Nmap scan report for System2
Host is up (0.000011s latency).
Not shown: 996 closed ports

           PORT            STATE  SERVICE   Vulnerabilities
5800/tcp                   open   vncc http remote_auth
110/tcp                    open   pop3      pop3_version
53/tcp                     open   domain    DNS_zone_transfer
21/tcp                     open   ftp       brute_force
Device type: general purpose
Running: OpenBSD
OS details: OpenBSD
Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
You can exploit the above listed vulnerabilities.
```

Step3: Scanning the webserver 2 using nmap command

```
> use_exploit DoS_attack System1
You have gained entry into the system. Now use ls to see the files in the
system
```

Step 6: Scores Step 4: Exploiting one of the Webservers

```
> ls
pin.txt

Now use "scp" to transfer files
Example: scp "file_name" "your_ip"

>> Remaining Time : 941 seconds >1
You have been caught while attempting to steal the information.
Use "exit" command to view your score and exit the game>
>
```

Step 5: File transfer

Fig. 39.2 (continued)

Table 39.1 Configuration of honeypot and regular systems

Easy to attack	Difficult to attack
Operating systems: • Windows Server 2003 • Windows XP • HP-UX 11i • Solaris	*Operating systems*: • OpenBSD • Linux • Mac OS X • Windows 8
Services and ports: • 21/tcp—ftp • 25/tcp—smtp • 80/tcp—http • 111/tcp—rpcbind • 135/tcp—msrpc	*Services and ports:* • 22/tcp-ssh • 53/tcp-domain • 110/tcp-pop3 • 139/tcp-netbios • 443/tcp-https • 445/tcp-microsoft-ds • 3306/tcp-mysql • 5900/tcp-vncc http • 6112/tcp-dtspc • 8080/tcp-apache

(Mean = 22; SD = 4). About 60% of participants possess 4-year undergraduate college degree, and 40% participants have a graduate or a professional degree. Hackers were remunerated INR 50 for participation in the study. In addition, they won INR 5 for transfer of a real credit-card file and lost INR 5 for transfer of a fake credit-card file.

3.4 Procedure

Participants were given instructions about their objective in the HackIt task, and they were informed about their own action's payoffs. Specifically, human hackers were asked to maximize their payoff by stealing the real credit-card file from the network over several rounds of play (participants were not aware of the end point of the game). Each round had two stages: probe stage and attack stage. The hacker could probe two webservers in the network using "nmap" utility. After probing the webservers, he/she received information about open ports, operating systems, services, and vulnerabilities associated with each webserver. Next, the hacker had two alternatives to choose during attack stage: exploit webserver 1 or exploit webserver 2. Hacker participants had to choose between these alternatives presented to them and exploit one of the webservers using "use_exploit" command. Once the webserver was exploited, hackers transferred the credit-card file to their remote computer.

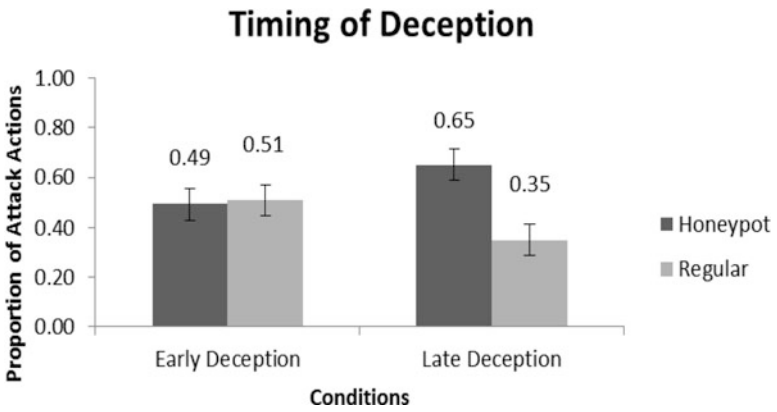

Fig. 39.3 Proportion of honeypot-/regular-attack actions for different timing of deception, early deception, and late deception

4 Results

We ran an experiment involving human participants and different between-subject conditions where deception occurred early or late in the HackIt environment. Figure 39.3 shows the proportion of regular-attack and honeypot-attack actions when the timing of deception was early and late.

The results revealed that the proportion of regular attacks was lower for late deception compared to early deception condition ($0.35 < 0.51$; $t(238) = 2.50$, $p < 0.05$, $\eta^2 = 0.21$). Thus, the simulated experiment conducted using HackIt tool helped us to validate the lab-based findings from the deception game obtained by Aggarwal et al. [16].

5 Discussion

In this chapter, we discussed HackIt, a simulation tool with a great potential to help cybersecurity researchers to investigate the decision-making of hackers and analysts in real-world cybersecurity scenarios. HackIt tool could provide a more realist task scenarios where the hackers were able to use real commands for network probing, exploiting, and transferring the information. Thus, the development of the HackIt tool could provide us a platform to migrate from abstract decision-making tasks to more applied and complex decision-making task. We showed a concrete research example of using HackIt to evaluate the effects of timing of deception on hacker's decisions. Results revealed that late timing of deception was more beneficial compared to early timing of deception in trapping hackers in honeypots. We believe that HackIt tool would be helpful in creating other cybersecurity

scenarios involving dynamic network sizes, dynamic network configurations, and various deception strategies.

In this chapter, we simulated the information-stealing scenario in HackIt tool where the hacker was supposed to steal a credit-card file from a regular webserver system. Researchers simulated a deception strategy in HackIt, which varied in time. We found that the proportion of honeypot attacks were higher when deception was late compared to early. Similarly, we found the proportion of regular attack actions were lower when deception was late compared to early. These results obtained using the HackIt tool agree with the findings obtained by [16]. The likely reason for this finding is that the hacker did not have any clue about the presence of deception in early rounds. According to IBLT, the recency and frequency of events play a role in decision-making. In the case of early deception condition, no instances were created in memory that tells about the presence of honeypots. Thus, the hackers attacked regular webservers more frequently. Due to trust build in early rounds, hacker's started relying on feedback in later rounds as well. Thus, they end up attacking more on honeypots compared to regular webservers.

In addition, results from the HackIt tool suggest that it would be beneficial for analysts to use deception late in a repeated cyberattack on a computer network. Thus, initial rounds of attack that lack deception will likely make hackers believe in the information provided by the network. Furthermore, late deception will likely help analysts to trap hackers effectively.

Currently, we investigated only one factor involving deception using HackIt tool, i.e., timing of deception. The network size was also limited to two webservers where one computer was acting as honeypot webserver and the other one was acting as regular webserver. In future, we wish to perform a series of experiments involving participants performing as hackers in other simulated network scenarios. We wish to extend our existing research work on creating deception in HackIt tool to investigate the effectiveness of deception in networks of different sizes. A network can be classified as small, medium, or large sized based on the number of computer systems present. The effectiveness of honeypots may vary with the network size. HackIt tool provides the flexibility to configure any number of webservers as honeypots and regular webservers. Honeypots are costly, and their installation and maintenance are also costly not only in terms of money and manpower but also in terms of time. Furthermore, we plan to investigate what proportion of honeypots in a network will be effective to trap hackers successfully in different ways. We also wish to investigate the effectiveness of deception technology via different cost models on hacker's probing actions and against other cyberattack types (like SQL injection, denial of service (DoS), and zero-day attacks).

Acknowledgment Palvi Aggarwal was supported by Visvesverya Ph.D. Scheme for Electronics and IT (IITM/DeitY-MLA/ASO/77), Department of Electronics and Information Technology, Ministry of Communication and IT, Government of India. Cleotilde Gonzalez was supported by the Army Research Laboratory under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA) to Cleotilde Gonzalez. Varun Dutt was supported by the Department of Science and Technology, Government of India award (award number: SR/CSRI/28/2013(G)) to Varun Dutt. The views and conclusions contained in this document are those of the authors and

should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the Indian or U.S. Government.

References

1. Trustwave global Security Report. (2015). Retrieved from https://www2.trustwave.com/rs/815-RFM-693/images/2015_TrustwaveGlobalSecurityReport.pdf
2. Symantec Corporation. (2014). *Internet security threat report*. Retrieved from http://www.symantec.com/content/en/us/enterprise/other_resources/bistr_main_report_v19_21291018.en-us.pdf
3. Whaley, B. (1982). Toward a general theory of deception. *Journal of Strategic Studies, 5*(1), 178–192.
4. Denning, D. (1999). *Information warfare and security*. New York: Addison Wesley.
5. Mitnick, K. D., & Simon, W. L. (2011). *The art of deception: Controlling the human element of security*. Indianapolis, IN: John Wiley & Sons.
6. Glantz, D. (1989). *Military deception in the second world war* (Cass series on soviet military theory and practice). London: Routledge. isbn:ISBN 978-0-714-63347-3.
7. Rowe, N. C., & Custy, E. J. (2008). Deception in cyber-attacks. In *Cyber warfare and cyber terrorism*. Hershey, PA: Information Science Reference.
8. Cohen, F. (1998). The deception toolkit. *Risks Digest, 19*.
9. Rowe, N. C. (2003, June). Counter planning deceptions to foil cyber-attack plans. In *IEEE Systems, Man and Cybernetics Society Information Assurance Workshop, 2003* (pp. 203–210). IEEE.
10. Heckman, K. E., Walsh, M. J., Stech, F. J., O'boyle, T. A., DiCato, S. R., & Herber, A. F. (2013). Active cyber defense with denial and deception: A cyber-wargame experiment. *Computers & Security, 37*, 72–77.
11. Spitzner, L. (2003). The honeynet project trapping the hackers. *IEEE Security & Privacy, 99*(2), 15–23.
12. Qassrawi, M. T., & Zhang, H. (2010). Client honeypots: Approaches and challenges. In *4th International Conference on New Trends in Information Science and Service Science (NISS) 2010* (pp. 19–25). IEEE.
13. Aggarwal, P., Gonzalez, C., & Dutt, V. (2017, June). Modeling the effects of amount and timing of deception in simulated network scenarios. In *2017 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA 2017)*, London, UK (pp. 1–7). IEEE.
14. Aggarwal, P., Gonzalez, C., & Dutt, V. (2016a, June). Looking from the hacker's perspective: Role of deceptive strategies in cyber security. In *2016 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA 2016)*, London, UK (pp. 1–6). IEEE.
15. Garg, N., & Daniel, G. (2007). Deception in honeynets: A game-theoretic analysis. In *IEEE SMC Information Assurance and Security Workshop. IAW'07*. IEEE.
16. Aggarwal, P., Gonzalez, C., & Dutt, V. (2016b). Cyber-security: Role of deception in cyber-attack detection. In D. Nicholson (Ed.), *Advances in human factors in cybersecurity* (pp. 85–96). Cham: Springer.