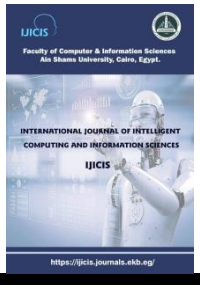
IJICIS, Vol.23, No.4, 50-65
DOI: 10.21608/ijicis.2023.247380.1306

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/



Navigating the Deception Stack: In-Depth Analysis and Application of Comprehensive Cyber Defense Solutions

Abdelrahman Aly*

Computer Systems Department,
Faculty of Computer and Information Science, Ain Shams University,
Cairo, Egypt
abdlrhmn.ali@cis.asu.edu.eg

Mahmoud Fayez

Computer Systems Department,
Faculty of Computer and Information Science, Ain Shams University,
Cairo, Egypt
mahmoud.fayez@cis.asu.edu.eg

Mirvat Al-Qutt

Computer Systems Department,
Faculty of Computer and Information Science, Ain Shams University,
Cairo, Egypt
mmalqutt@cis.asu.edu.eg

Ahmed M. Hamad

Computer Systems Department,
Faculty of Computer and Information Sciences, Ain Shams University
Cairo, Egypt
ahmed.hamad@cis.asu.edu.eg

Received 2023-11-08; Revised 2023-12-05; Accepted 2023-12-10

Abstract *— Deception techniques play a crucial role in enhancing cybersecurity by misleading attackers and safeguarding critical systems. The strategic placement of decoys constructs an elaborate defense architecture that can effectively thwart unauthorized access. This paper presents a comprehensive survey of deception techniques from a research perspective, highlighting their classification, modeling methodologies, and deployment strategies. Additionally, it explores the major research issues associated with these techniques, including the challenge of maintaining the believability of decoys and the ethical implications of their use. Moreover, the paper investigates the role of Moving Target Defense (MTD) in cyber deception, emphasizing its dynamic nature and specifically the network environments. This paper goes beyond theoretical discussions and digs into the implementation details of real frameworks operating at four layers of the deception stack composed of the network, system, software, and data layers. The seamless integration between these layers is essential for creating a convincing deceptive environment. It categorizes these implementations into four main approaches and highlights the corresponding systems that have been developed, thus offering a roadmap for future research and development in this critical field of cybersecurity.*

Keywords: *Cyber Deception, MTD, MITRE, Adversarial techniques, Honeypots*

1. Introduction

***Corresponding Author**: Abdelrahman Aly

Computer Systems Department, Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt

Email address: abdlrhmn.ali@cis.asu.edu.eg

Cyber intrusion [1] refers to unauthorized access or attack on a computer system or network, often to steal or manipulate sensitive information or cause harm to the system. This can include hacking, malware attacks [2], phishing scams [3], and other forms of cybercrime. Cyber intrusions can be carried out by individuals or organized groups and can have serious consequences for individuals and organizations, including financial loss, reputational damage, and legal repercussions.

Cyberreactive activities refer to actions taken in response to a cyber security incident or threat [4]. This could include things like identifying the source of the threat, implementing measures to prevent further damage or attacks, and recovering from the incident.

Proactive cyber activities [5], on the other hand, are actions taken to prevent cyber security incidents from occurring in the first place. This could include things like implementing strong passwords, regularly updating software and security protocols, and conducting regular security assessments. Proactive cyber activities are typically focused on building a strong and resilient cyber security infrastructure to protect against future threats.

Various deception techniques have been developed and implemented to address different aspects of cybersecurity.

Honeypots [6], for instance, act as systems or networks specifically designed to attract and trap attackers, serving the dual purpose of gathering intelligence on their tactics while distracting them from more critical systems. Decoy servers [7], on the other hand, mimic real servers, leading attackers to believe they have successfully compromised a system, thereby gathering valuable insights into their behavior. Spoofed websites are designed to deceive users into thinking they are accessing legitimate websites, often used for phishing sensitive information.

Deception networks[8] replicate real networks, tricking attackers into believing they have gained control, and enabling defenders to gather intelligence on their techniques. Decoy documents, meanwhile, entice attackers to reveal their tactics and techniques through hidden tracking mechanisms, allowing defenders to monitor and collect valuable information.

These different deception implementations provide organizations with a range of tools to combat cyber threats. By leveraging deception techniques, organizations can proactively defend their networks and critical assets, making it more challenging for attackers to succeed in their malicious endeavors.

In the field of deception techniques in computer security, survey papers typically provide a theoretical exploration of the differences between various approaches.

Xiao Han et.al [5]surveyed from a research perspective, focusing on deception techniques. Their paper extensively discussed the classification, modeling, and deployment methodologies employed in this domain. Another survey conducted by Zhuo Lu highlighted high-level models and listed current implementations of deception techniques [9] but without further technical details or real-life examples. Additionally, Lu identified major research issues in the field.

Building upon the existing research, Sailik Sengupta [10] aims to illustrate different implementations of real frameworks that operate on various deception stack layers. The focus is on practical applications rather than theoretical discussions.

The organization of this paper is as follows. The introduction, in Section I, provides a brief overview of different types of deception techniques used in cyber defense. In Section II, network-based deception is explored with the MTDCD (MTD Enhanced Cyber Deception Defense System) as a specific example. Section III focuses on system-based deception Section IV investigates malware-based deception and discusses the implementation of SODA (System for Cyber Deception Orchestration and Automation). Section V examines web-based deception strategies, highlighting practical examples. The paper concludes in Section VI with reflections and future directions for defensive deception research.

2. Network-based Deception (Network Layer)

Gao, Wang et.al [8] developed the MTDCD system as an enhanced cyber deception defense mechanism. The study discussed the drawback of the defensive strategy that involves deploying scams or other misleading elements within a network information system to interfere with an adversary's perception of the system and achieve the goal of detecting, delaying, or blocking their activities [11].
The mentioned approach to defense has some limitations. One problem is that even after deploying cyber deception, the network system configuration remains static, which allows sophisticated adversaries to potentially bypass the defense mechanism after careful detection and analysis. Another issue is that it can be complicated to deploy network deception in traditional networks.
Gao, Wang et.al [8] developed a network deception system that uses techniques such as MTD [12] (moving target defense) and software-defined networking (SDN) [13] to defend against advanced persistent threats (APT) [14]. This system would use data packet header rewriting to create decoy nodes and virtual network topology in order to mislead and increase the time cost for attackers.
This system also uses IP randomization technology [15] to further defend against APT attackers. By regularly changing the IP addresses of nodes in the network, the system makes it more difficult for attackers to gather accurate information about the network and forces them to reprobe the network, which increases their time cost and makes it harder for them to maintain a foothold in the network. By combining cyber deception and MTD, this system aims to effectively resist continuous network reconnaissance attacks and protect the intranet from APT intrusions.

2.1. Implementation

The system is designed to provide security and protect against network attacks and consists of three main modules. The virtual network topology module is responsible for creating and managing a virtual network, which can be used to simulate a network environment to test network security measures or to provide an additional layer of protection for a real network.
The IP randomization module is used to randomize the IP addresses of devices on the network, making it more difficult for attackers to target specific devices.
The deception server is used to lure attackers away from the real network and towards a simulated network environment, where their activities can be monitored and analyzed.

2.1.1. The Virtual Network Topology Module:

The virtual network topology module of this system is responsible for creating and managing a virtual network that can be used for security purposes and is divided into three submodules:
The virtual network topology generation module specifies the components of the virtual network and their connectivity, including the real and virtual address information of host and decoy nodes [16], as well as the connectivity between them.
The decoy node generation module creates a large number of decoy nodes that can be used to lure attackers away from the real network and towards a simulated environment where their activities can be monitored. The flow table distribution module generates flow tables based on the virtual network topology and pushes them to the network switches to control transmission.

2.1.2. IP Randomization Module

The IP randomization module of this system is designed to coordinate the address conversion between the host and decoy nodes in the network and is divided into three submodules. The address management module is responsible for tracking the IP addresses of the host and decoy nodes, as well as allocating new IP addresses to them to ensure that they do not interact with each other.
The conversion decision module sets the parameters for IP address randomization and determines the construction of the virtual network topology, including the size of the network, the number of subnets, and the number of decoy nodes. The connection maintenance module ensures that the end-to-end connection is not interrupted when the address change occurs, making the address change transparent to the user.

2.1.3. Deception Server

The deception server in this system is responsible for making responses based on the specifications of the virtual network view in order to deceive malicious scanners [17]. It includes several modules, including the address management module, the message processing module, the DHCP processing module, the ARP processing module, the ICMP processing module, and the routing simulation module. The address management module ensures that the virtual network topology is maintained, while the message processing module parses incoming data packets and sends them to the appropriate module for processing based on their type. The DHCP, ARP, and ICMP processing modules deceive malicious scanners by responding to specific requests, and the routing simulation module simulates the multi-hop path between two nodes by sending ICMP timeout and port unreachable messages.

2.2. Evaluation

The case described in this study evaluates the effectiveness of deploying virtual network topologies (VNTs) to extend the time it takes for attackers to discover vulnerable hosts in a network. The study compares the time taken for an attacker to scan for vulnerable hosts under a network with no VNT to one with a VNT.
The results show that the deployment of a VNT increases the maximum and the average time it takes for an attacker to discover vulnerable hosts by an average of seven times. The study also investigates the impact of VNTs on the number of hosts attacked by an attacker over time. The results show that the deployment of a VNT extends the time it takes for an attacker to attack a vulnerable host by an average of eight times.
The study also examines the impact of VNTs on system overhead, specifically network latency and flow table reinstallation frequency. The results show that the network latency increases by 2.2%-11.6% with the deployment of VNTs and dynamic VNTs, and that in the case of adding IP randomization frequency increases by 3.1%-26.8% with the deployment of dynamic VNTs.

2.3. Gaps, Challenges, and Future Work

2.3.1. Gaps and Challenges

The dynamic cyber deception system based on SDN shows promise but also faces several challenges. One key issue is ensuring the seamless coordination between real and virtual entities within the network, a pivotal aspect that ensures the system's believability and effectiveness. Further, maintaining the integrity and responsiveness of the deception mechanisms, such as IP randomization and virtual network topology, under varying attack scenarios remains a constant challenge.

2.3.2. Advancements and Positive Aspects

The tool exhibits considerable advancements in leveraging SDN for cyber deception, notably in the realms of virtual network topology and IP randomization. These modules collaborate to create a dynamic and adaptive deception environment, enhancing the system's resilience against potential attackers. For instance, the virtual network topology module adeptly manages the generation of decoy nodes and flow table distributions, thereby enriching the deception layer. IP randomization further augments defense mechanisms by introducing variability, making it difficult for attackers to discern the actual network entities.

2.3.3. Potential Areas for Improvement

While the system exhibits robust attributes, there is room for enhancement. Improving the deception server's capabilities to more convincingly emulate real network behavior could be instrumental in bolstering defense mechanisms. Additionally, exploring more adaptive and intelligent algorithms, which allow the system to autonomously evolve its deception strategies based on real-time threat analysis, could further strengthen the system's ability to thwart sophisticated attacks. Enhanced mechanisms to ensure uninterrupted and seamless communication, even in the face of IP changes and other dynamic adjustments, can also augment the system's reliability and effectiveness.

3. System-based Deception (System Layer)

Moonraker was first introduced by Shade et.al [18], allowing the researchers to monitor the participants' actions in real-time, allowing them to intervene and manipulate the outcome of the TTP [19]being executed. This allowed the researchers to evaluate the effectiveness of the deceptive responses in disrupting the participants' ability to successfully execute their objectives.
Moonraker intercepts a specific set of commands that were likely to be used by participants to execute certain tactics, techniques, and procedures (TTPs) as part of their objective. In the control condition, the system responded normally to all commands. In the deception condition, there were two types of hosts: decoys and responsive hosts. Decoy hosts aren't real mechanisms, they do not allow participants to proceed in any TTP after executing. Responsive machines, on the other hand, return results for all executed commands until TTP6 via Moonraker predefined commands/responses.
Overall, Moonraker provided a useful tool for monitoring and manipulating memory and processes to study the behavior of participants and their ability to execute specific tactics, techniques, and procedures. It allowed the researchers to remotely intercept and manipulate system functions and inject code, giving them the ability to gain control of the system and observe the behavior of the participants. Overall, Moonraker proved to be a valuable tool in assessing the effectiveness of deceptive responses in disrupting cyber threats.

3.1. Implementation

The Deceptive condition of Moonraker aims to mislead participants about the true nature of the network and the target host, potentially causing them to make incorrect decisions and assumptions throughout different TTPs (Tactics, Techniques, and Procedures). Moonraker was developed to adapt with the participants' TTPs to exfiltrate the system process. The used TTPs were consecutive to list the network hosts and copy to them the process_dump executable that is responsible for exfiltrating the running

processes on the system according to the predefined schedule. Also, Moonraker intercepted the used commands for scheduling the execution and delayed it to waste additional time and power. Moonraker is targeting to hook all the used commands and modify the normal response to mislead participants.

3.1.1. The Technical Approach for Attack Execution and Deception

In this research, a new technique was followed to improve the actions against system-based attacks. The researcher works to hook the critical binaries that are used for exfiltrating sensitive information by changing their behavior to return controlled output.
First, Moonraker intercepts the `net use` command that is responsible for mapping network drives and returns a successful result in case the decoy host is selected via an attacker After that, If the attacker selects a controlled host via Moonraker and starts to copy a malicious executable `process_dump.exe`, Moonraker will intercept the copy process and replace it with another fully controlled executable that does the same function but return a different output.
In the mentioned TTPs, after copying the malicious file, the participant should schedule this attack using schtasks command which is also manipulated using Moonraker to delay the task output by five minutes.
Then, the participants should copy the output of process_dump which is a list of running processes on the host to their machines. To conduct such an activity, participants should use copy and type commands. This command has been manipulated using Moonraker to corrupt the expected output. This corruption needs investigation of the output file to be detected.
The Technical Approach to Log Participants' Actions:
Moonraker’s collected data from various sources in order to track the behavior of participants and systems within the environment. The custom key logger you mentioned captures keystrokes, as well as the date and time of key press and the pressing of the ENTER key, which allows the track of the time between commands and the time spent typing a command. Also, there was a modified command prompt to record the commands typed by participants and the responses displayed to them and used two additional data sources as backups.

3.2. Evaluation

To test the effectiveness of deceptive command line responses in impeding the progress of attackers, the study compared the number of participants in two conditions[20] (one with deceptive responses, and one without) who were able to successfully complete a task. Success was determined by examining a submitted zip archive for certain files. The results showed that the control condition (without deceptive responses) was significantly more successful than the deception condition (with deceptive responses).
Another measure of effectiveness used in the study was the proportion of TTP (tactics, techniques, and procedures) commands that were successful in each condition. The results showed that a significantly higher proportion of TTP commands were successful in the control condition compared to the deception condition. This suggests that the deceptive responses did impede the progress of the attackers.
In this deceptive methodology, participants were given deceptive responses to a limited set of predetermined commands. This could potentially limit the internal validity of the study, as participants could bypass the deception by using different commands. To address this, the study tried to constrain the participants' command and program usage through instructions and technical solutions. However, this could also potentially limit the external validity of the study, as it may have artificially influenced the attack behavior of the participants.

3.3. Gaps, Challenges, and Future Work

3.3.1. Gaps and Challenges

Despite the innovative approach of the Moonraker framework in system-based deception, there remain several gaps and challenges to address. A gap lies in predicting how attackers adapt to deceptive techniques in real-time. Current implementations are static, focusing mainly on predefined deceptive responses such as altering command outputs. This rigidity could limit the effectiveness of the deception, as adaptive adversaries might quickly recognize and circumvent the deceptive measures. Furthermore, it might face challenges in adapting to various system architectures and environments. Its effectiveness could be limited if it can't seamlessly integrate and operate across diverse systems.

3.3.2. Advancements and Positive Aspects

Moonraker brings notable advancements to the realm of cybersecurity, particularly in crafting deceptive environments to mislead attackers. Its ability to monitor and actively manipulate memory and processes showcases a robust approach toward creating uncertainty and confusion for attackers. The framework allows for customization, offering flexibility in intercepting and manipulating system application programming interfaces and internal function calls, thereby enabling the injection of code and altering processes to implement deceptive responses effectively. This level of sophistication enhances the resilience of systems against attacks by making the attack pathways more convoluted and challenging for adversaries to navigate.

3.3.3. Potential Areas for Improvement

To enhance Moonraker's effectiveness further, research could focus on making the deceptive techniques more dynamic and adaptable. Incorporating machine learning and artificial intelligence could be pivotal, allowing the system to analyze attackers' behaviors in real time and tailor the deception strategies accordingly. This could make the deceptive environment more resilient against various types of attackers, including those who quickly adapt to recognized deceptive patterns. Additionally, exploring the psychological dimensions of deception could also be beneficial. Understanding the attackers' decision-making processes, tolerance to uncertainties, and reactions to anomalies could provide invaluable insights, enabling the crafting of deception strategies that are more psychologically compelling and challenging for attackers to discern and navigate.

4. Malware-based Deception (Software Layer)

SODA by Sajid, Wei, Abdeen et.al [21], is an autonomous cyber deception system that aims to address the limitations of existing approaches by providing dynamic and automated orchestration of deception[22]. It does this by analyzing malware to extract malicious subgraphs (MSGs) that represent sequences of API calls [23]that work together to perform a malicious task. SODA maps these MSGs to the MITRE ATT&CK framework to understand the malware's behaviors at the kill chain tactical level. This knowledge is used to create a deception playbook, a set of deception course-of-actions that can be used to deceive specific malicious behaviors with a given deception goal and strategy[24]. Users can choose from pre-built deception playbook profiles or create their own, and SODA will automatically orchestrate the deception in real time using API hooking[25]. SODA has been evaluated with recent malware, including information stealers, ransomware, and remote access trojans, and has shown high

accuracy and scalability, with an accuracy of 95% in deceiving malware and a recall value of 97% for MSG extraction.
SODA is a system for creating deception playbooks, which are used to deceive and defend against malicious software. The playbooks are created by extracting "malicious sub-graphs" (MSGs) from malware repositories and mapping them to behaviors. The mapped MSGs are then used to synthesize deception playbooks, which can be used to defend against attacks from specific types of malware. The system also includes a template code generator that can be used to create deception ploys for various types of malicious behavior. The deception playbooks can be created with minimal errors and can be customized for specific types of attacks.

4.1. Implementation

SODA is a system designed to provide dynamic orchestration of cyber deception at runtime. It aims to mislead and confuse malicious software, also known as malware, to protect a system from being compromised. The system works in two phases: the Deception Playbook Creation phase and the Real-time Deception phase.
In the Deception Playbook Creation phase, SODA first uses a malware repository [26], [27]to extract malicious sub-graphs (MSGs) from thousands of malware samples. These MSGs represent sequences of Windows API calls that achieve specific malicious objectives and can be represented as graphs. The system then maps the extracted MSGs to specific malicious behaviors and synthesizes a Deception Playbook, which consists of deception ploys and strategies for different 4D goals (i.e., Deception, Denial, Detection, and Depletion).
In the Real-time Deception phase, SODA deploys the Deception Playbook to deceive malware in real time through the use of API hooking. A Detection Agent is used to detect the presence of malware, and the Orchestration Engine Client (OEC) injects an End-Point Dynamic Link Library (DLL) into the malware process to enable real-time orchestration. The OEC communicates with the Orchestration Engine Server (OES) to select the appropriate deception ploys and deploys a HoneyFactory, which consists of scripts and fake resources, to execute the selected deception strategies and achieve the desired 4D goals.

4.1.1. Deception Playbook Creation

The deception playbook creation phase is a process for creating and storing playbooks that can be used to deceive and defend against specific types of malware. The playbooks are created by extracting MSGs (malicious sub-graphs) from real-world malware and mapping them to MITRE techniques and defined malicious behaviors. The mapped MSGs are then used to synthesize deception ploys, which are specific actions taken to deceive the malicious behaviors of the malware. The deception ploys are organized into deception playbooks, which are stored and used by the deception factory to perform the actions defined within the playbooks when needed. The deception factory includes hooks, REST APIs, and HoneyFactories (HFs), which are used to execute the actions defined in the playbooks. In this phase, there are three submodules which are mentioned below.

4.1.2. Malicious Sub-graphs (MSG) Extraction

This component of the SODA system is used to collect execution traces from malware and extract MSGs (graphs of Windows API calls and the data flow between them). The API Call Tracer is based on the Cuckoo sandbox and is customized to monitor a specific set of Windows APIs with all their

parameter values. The set of APIs to be monitored is determined by running representative malware samples through gExtractor[28], a dynamic analysis tool, and identifying the unique APIs invoked by the malware. The API Call Tracer is automated by using scripts to generate template code for each API based on rules provided by the Cuckoo sandbox [29]and the API definitions from the Microsoft Developer Network (MSDN) website. The extracted MSGs can be used to understand the malware's execution flow and design an accurate deception plan.

4.1.3. MSG Classifier

The MSG Classifier is a component of the SODA system that maps MSGs (low-level implementation details of malware) to MITRE ATT&CK techniques (structured knowledge base of adversary tactics and techniques that illustrate the attack lifecycle of an adversary). The MSG Classifier does this by representing each MITRE technique and API as a vector and using these vectors for real-time MSG-to-MITRE classification. The MSG Classifier uses descriptions of MITRE techniques, APIs, and Stack Overflow questions and answers to create these vectors. The descriptions are preprocessed to remove unnecessary words and low-occurring words, and the verbs are lemmatized to their base form. The vectors for the MITRE techniques and APIs are then enriched using the term frequency-inverse document frequency (TF-IDF) to highlight important words. The MSG-to-MITRE classification process involves finding the cosine similarity between the vectors for the MSGs and the vectors for the MITRE techniques and selecting the technique with the highest similarity score as the mapped technique.

4.1.4. Deception Factory Synthesis

Deception ploys are actions that can be taken to deceive a specific malicious behavior of malware, intending to achieve a certain deception strategy or meet certain 4D deception goals (i.e., deceive, deter, disrupt, and degrade). These ploys can be grouped into profiles based on the co-occurrence of different behaviors, and a Deception Playbook can be created that includes all the necessary ploys for a given malware. The Deception Playbook can then be used to develop WinAPI hooks, which are used to intercept and alter the execution of the malware as needed to implement the deception ploys.

4.1.5. Real-Time Orchestration

SODA is a cyber deception system that is designed to deceive malware at runtime to disrupt its execution and better understand its behavior.

It consists of two main phases(offline and online phases):
The offline phase: During the offline phase, SODA develops deception profiles that outline specific deceptive tactics to use against different types of malware behaviors. These profiles are created by mapping malicious subgraphs (MSGs) to MITRE ATT&CK techniques and identifying the most effective deception ploys to use against them.
The online phase: During the online phase, SODA deploys these deception profiles in real-time using a detection agent, an orchestration engine server and client, and honeypots. The detection agent is responsible for detecting the presence of malware and triggering the orchestration process. The orchestration engine server and client facilitate communication between the victim's system and the honeypots[30], and the honeypots execute the chosen deception tactics. SODA is designed to be flexible and customizable, allowing users to create their deception profiles or select from pre-built ones.

4.2. Evaluation

The overall accuracy of SODA in terms of deceiving malware was evaluated by using four types of malware (RATs, InfoStealers, Ransomware, and Spyware) for testing. The evaluation metrics used were the number of ploys that SODA was able to use to deceive the malware out of the number of ploys that the user selected. The observation criteria to consider a deception ploy as successful varied based on the type of malware being tested. For RATs, the effectiveness of the deception ploys was observed via the Command and Control (C2) server. For InfoStealers, Wireshark was used to examine exfiltrated credentials to determine whether the deception ploys were working. For Ransomware, the successful indication of deception was fooling the malware into creating a ransom note, even if the encryption did not take place. For Spyware, the IP address where the malware was supposed to upload the collected information was identified using Wireshark and ApateDNS was used to redirect the packet to a hosted FTP server. In the experiments, SODA achieved an overall accuracy of 95%, with 224 out of 237 ploys being successful in deceiving the malware.

4.3. Gaps, Challenges, and Future Work

4.3.1. Gaps and Challenges

The document highlights a significant "semantic gap" in the language used by attackers and defenders, making the identification and response to attacks cumbersome. Existing deception techniques, being mostly static, are easily detected and bypassed by attackers, pointing towards a glaring inadequacy in current deception methodologies. Additionally, the challenges lie in understanding the attacker's behavior intricately and grappling with the limitations posed by the existing tools. Moreover, evaluating the real-time effectiveness of these deception techniques presents a considerable challenge.

4.3.2. Advancements and Positive Aspects

Advancements in the field of Active Cyber Deception have been notable. There is a marked utilization of machine learning, automating the creation of Deception Playbooks, signifying a positive stride toward sophisticated deception techniques. The inception of the SODA system embodies a pivotal advancement, designed meticulously to address and navigate through the prevalent gaps and challenges. SODA's architecture, promoting automation and dynamic orchestration, heralds a new era of agility and robustness in deploying deception strategies against cyber threats.

4.3.3. Potential Areas for Improvement

Future work avenues beckon a focused improvement in components of the SODA system, like the MSG Classifier, enhancing its overall precision and reliability. A call for expansive research resonates, aiming to delve deeper into the effectiveness of deception techniques, envisaging their seamless integration with an array of other cybersecurity tools and techniques. This approach seeks to foster a comprehensive and multifaceted defense mechanism, bolstering cybersecurity realms against evolving threats.

5. Web-based Deception

According to (Han, Kheir, & Balzarotti, 2017),[31] the author uses deception techniques to detect web attacks by using fake or misleading information to lure attackers into revealing their presence. These techniques are based on the concept of honeytokens, which are fake elements inserted into a web application that is meant to be detected if accessed.
The used deception techniques can be classified into three categories: alteration to honey trap data, such as fake hidden form fields or additional URL parameters; honey trap resources requested, such as fake pages or directories listed in the robot.txt file; and honey trap data collected and used by the attacker, such as fake accounts only visible in the source HTML code. Han, Kheir, et al. [32]propose two additional deception techniques: a fake protected area that prompts the client for authentication, and fake vulnerabilities that return realistic error messages when tampered with by an attacker. These techniques can be used to detect and deter attackers by keeping them busy trying to exploit fake vulnerabilities.

5.1. Implementation:

The authors have designed and implemented a deception framework for detecting web attacks. The framework acts as a reverse proxy in front of a web application, allowing the transparent insertion of deceptive elements such as fake cookies, hidden form fields, and fake protected areas[33]. These elements are added to the HTTP protocol and HTML content of incoming and outgoing traffic through the use of regular expression rules. The framework is implemented using the open-source HTTP hacking tool Hoxy[34], which allows for the interception and modification of HTTP requests and responses. The authors also describe a method for deploying deception techniques, which involves deploying the framework on a separate server and redirecting traffic to it. This allows the deception techniques to be tested and evaluated in a real-world deployment without modifying the target web application.

5.1.1. Use of Deception in a Real Content Management System

This experiment involved the deployment of deception techniques in a Content Management System (CMS) in order to evaluate their false positive rate in the presence of legitimate users. The CMS was based on Open Atrium and was customized to allow research project members to manage publicly accessible websites. The deception techniques were placed in the public and private spaces of the CMS in order to resemble the deployment used in a capture the flag (CTF) exercise. The goal of this experiment was to determine how often the deception techniques generated false alarms when used by legitimate users.

5.1.2. *Use of Deception in a Capture-The-Flag Competition*

In the second experiment, deception techniques were integrated into a Capture the Flag (CTF) [35]exercise, in which participants are presented with a specific environment where vulnerabilities are purposely planted. The goal of this experiment was to evaluate the ability of deception techniques to detect web attacks in their early stages, by using a CTF competition to mimic what users would do to discover vulnerabilities in an unknown piece of software. The CTF exercise was organized by Orange Labs and simulated a situation where participants audited the security of an e-commerce application in a black-box approach. Many classic vulnerabilities such as cross site scripting, local file inclusion, SQL injection, and remote code execution were planted at different locations in the application[36]. Deception techniques were placed in the application to resemble as closely as possible the deployment

used in the first experiment. The results of the experiment showed that deception techniques were effective at detecting attacks in their early stages and that they had a low false positive rate.

5.2. Evaluation:

The results of the experiments showed that deception techniques were effective at detecting web attacks in both a real CMS application and a CTF exercise. In the CMS experiment, four alerts were triggered by the honey trap resources placed in the robots.txt file, which were caused by a scan attempt to test the system and check for known vulnerabilities. No alerts were triggered by the deception elements placed in the private space of the CMS.
In the CTF experiment, 84 out of 150 participants triggered at least one of the 12 deception traps, while only 25 participants successfully discovered at least one flag. The results suggest that the deceptive elements were easier to trigger than the real vulnerabilities and that participants who discovered real vulnerabilities were also likely to trigger deception traps. However, not all participants who triggered deception traps discovered real vulnerabilities. The results also showed that the majority of the deception traps were triggered manually, rather than by automated tools. Overall, the experiments demonstrated the effectiveness of deception techniques in detecting web attacks in both real and simulated environments.

5.3. Gaps, Challenges, and Future Work

5.3.1. Gaps and Challenges

The paper mentioned a notable gap in the research concerning the efficacy of deception techniques in detecting web attacks, emphasizing a profound need for a richer diversity and realism in datasets for evaluating these methodologies. In terms of challenges, the sophistication of attackers, who might recognize and evade deception techniques, necessitates their continuous refinement and enhancement. Another formidable challenge accentuated is the careful navigation through legal and ethical landscapes when deploying deception techniques in a production environment, ensuring that the implementation aligns with requisite legal and moral codes.

5.3.2. Advancements and Positive Aspects

A beacon of advancement highlighted in the paper is the introduction of a novel framework geared towards the implementation of deception techniques in web applications. This framework is imbued with a variety of innovative traps and decoys meticulously designed to detect and thwart attackers effectively. The illustration of this framework's effectiveness, underscored through the conduit of two experiments, further substantiates its practical applicability and proficiency in a real-world setting.

5.3.3. Potential Areas for Improvement

For future endeavors, the paper underscores the enhancement of the realism and diversity of datasets used for the evaluative analysis of deception techniques as a pivotal area of focus. Additionally, a foray into exploring innovative traps and decoys opens avenues for refinement and enhancement of the deception framework. A resonating emphasis is also placed on the necessity to delve deeper into researching the multifaceted legal and ethical implications intrinsic to the deployment of deception

techniques within web applications, ensuring a balanced alignment with prevailing legal and ethical standards.

.

6. Conclusion

In this paper, we have presented a comprehensive survey of deception techniques in computer security, examining them from a research perspective. We explored the classification, modeling methodologies, and deployment strategies associated with deception techniques. Additionally, we investigated the role of Moving Target Defense (MTD) in cyber deception and its effectiveness against Advanced Persistent Threats (APTs) and different attack representation models.

Moving beyond theoretical discussions, our paper delved into the implementation details of real frameworks operating at various layers of the deception stack. We categorized these implementations into four main approaches: system-based deception, network-based deception, malware-based deception, and web-based deception. Each approach was accompanied by an example system that has been developed to effectively deceive attackers within the respective domain.

Among the implementations discussed, Moonraker showcased the effectiveness of system-based deception in misleading attackers and disrupting their objectives. Additionally, we examined two other deception techniques: MTDCD, a network-based deception approach, and SODA, a malware-based deception system. These implementations demonstrated their ability to mislead attackers in their respective domains and protect critical assets. Moreover, we explored the realm of web-based deception, encompassing various implementations that manipulate web content, session management, and user interaction to misdirect potential adversaries.

Through our survey, we have provided practical insights into the effectiveness and real-world applicability of deception techniques in computer security. By leveraging deception strategies in different layers of the deception stack, organizations can enhance their cybersecurity defenses and thwart malicious actors. These techniques, such as honeypots, decoy servers, spoofed websites, deception networks, and decoy documents, enable defenders to gather intelligence, distract attackers, and gain an upper hand in the cybersecurity landscape.

It is crucial to acknowledge the ongoing evolution of cyber threats and the dynamic nature of the cybersecurity landscape. Future research should focus on advancing deception techniques, addressing their limitations, and staying ahead of sophisticated attackers. By continuously improving and adapting deception strategies, organizations can effectively counter emerging cyber threats and protect their critical systems and assets.

In conclusion, deception techniques serve as valuable tools in enhancing cybersecurity by misleading attackers, protecting critical assets and mitigating the impact of cyber threats. By combining theoretical understanding with practical implementation examples, this paper contributes to the growing body of knowledge in the field of deception techniques in computer security, paving the way for further advancements and innovations in the realm of cyber deception.

7. Acknowledgments

I extend my deepest appreciation to my supervisors for their invaluable guidance and expertise, which have been crucial in the progression of this research. Your support and insightful feedback have profoundly shaped my academic journey, and for that, I am eternally grateful.

Special thanks go to my wife and family, whose love, understanding, and unwavering belief in me have been my greatest source of strength and inspiration. Your support has been the bedrock of my perseverance and success.

I am also indebted to Loay Abdelrazik for his constant technical support and mentorship. Your contributions have been instrumental in my work. My gratitude also extends to my colleagues for their collaborative spirit, which has greatly enriched my professional development.
Thank you all for your significant contributions and for being an integral part of this journey.

References

[1] W. A. H. M. Ghanem et al., "Cyber Intrusion Detection System Based on a Multiobjective Binary Bat Algorithm for Feature Selection and Enhanced Bat Algorithm for Parameter Optimization in Neural Networks," IEEE Access, vol. 10, pp. 76318–76339, 2022, doi: 10.1109/ACCESS.2022.3192472.
[2] "A Survey of Stealth Malware Attacks, Mitigation Measures, and Steps Toward Autonomous Open World Solutions | IEEE Journals & Magazine | IEEE Xplore." Accessed: Jun. 21, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7778160
[3] J. Wu et al., "Who Are the Phishers? Phishing Scam Detection on Ethereum via Network Embedding," IEEE Trans Syst Man Cybern Syst, vol. 52, no. 2, pp. 1156–1166, Feb. 2022, doi: 10.1109/TSMC.2020.3016821.
[4] J. Kotsias, A. Ahmad, and R. Scheepers, "Adopting and integrating cyber-threat intelligence in a commercial organisation," https://doi.org/10.1080/0960085X.2022.2088414, vol. 32, no. 1, pp. 35–51, 2022, doi: 10.1080/0960085X.2022.2088414.
[5] N. Sun et al., "Cyber Threat Intelligence Mining for Proactive Cybersecurity Defense: A Survey and New Perspectives," IEEE Communications Surveys and Tutorials, 2023, doi: 10.1109/COMST.2023.3273282.
[6] H. Wang, H. He, W. Zhang, W. Liu, P. Liu, and A. Javadpour, "Using honeypots to model botnet attacks on the internet of medical things," Computers and Electrical Engineering, vol. 102, p. 108212, Sep. 2022, doi: 10.1016/J.COMPELECENG.2022.108212.
[7] W. Tounsi, "Cyber Deception, the Ultimate Piece of a Defensive Strategy-Proof of Concept," 2022 6th Cyber Security in Networking Conference, CSNet 2022, 2022, doi: 10.1109/CSNET56116.2022.9955605.
[8] C. Gao, Y. Wang, X. Xiong, and W. Zhao, "MTDCD: An MTD Enhanced Cyber Deception Defense System," in IMCEC 2021 - IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference, Institute of Electrical and Electronics Engineers Inc., Jun. 2021, pp. 1412–1417. doi: 10.1109/IMCEC51613.2021.9482133.
[9] L. Zhang and V. L. L. Thing, "Three decades of deception techniques in active cyber defense - Retrospect and outlook," Comput Secur, vol. 106, p. 102288, Jul. 2021, doi: 10.1016/J.COSE.2021.102288.
[10] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A Survey of Moving Target Defenses for Network Security," IEEE Communications Surveys and Tutorials, vol. 22, no. 3, pp. 1909–1941, Jul. 2020, doi: 10.1109/COMST.2020.2982955.
[11] K. D. Bowers et al., "Defending Against the Unknown Enemy: Applying FlipIt to System Security," Cryptology ePrint Archive, 2012.
[12] T. A. Nguyen, M. Kim, J. Lee, D. Min, J. W. Lee, and D. Kim, "Performability evaluation of switch-over Moving Target Defence mechanisms in a Software Defined Networking using stochastic reward nets," Journal of Network and Computer Applications, vol. 199, Mar. 2022, doi: 10.1016/J.JNCA.2021.103267.
[13] "IEEE Xplore Full-Text PDF:" Accessed: Jun. 21, 2023. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9943548

[14] M. N. A. ; Khalid et al., “Recent Developments in Game-Theory Approaches for the Detection and Defense against Advanced Persistent Threats (APTs): A Systematic Review,” Mathematics 2023, Vol. 11, Page 1353, vol. 11, no. 6, p. 1353, Mar. 2023, doi: 10.3390/MATH11061353.
[15] J. Narantuya et al., “SDN-based IP shuffling moving target defense with multiple SDN controllers,” Proceedings - 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume, DSN-S 2019, vol. 00, pp. 15–16, Jan. 2019, doi: 10.1109/DSN-S.2019.00013.
[16] J. Sun and K. Sun, “DESIR: Decoy-enhanced seamless IP randomization,” Proceedings - IEEE INFOCOM, vol. 2016-July, Jul. 2016, doi: 10.1109/INFOCOM.2016.7524602.
[17] M. Dandotiya, “A Secure Detection Framework for ARP, DHCP, and DoS Attacks on Kali Linux,” Int J Res Appl Sci Eng Technol, vol. 10, no. 7, pp. 3044–3053, Jul. 2022, doi: 10.22214/IJRASET.2022.42176.
[18] T. B. Shade, A. V. Rogers, K. J. Ferguson-Walter, S. B. Elson, D. K. Fayette, and K. E. Heckman, “The MoonRaker study: An experimental evaluation of host-based deception,” in Proceedings of the Annual Hawaii International Conference on System Sciences, IEEE Computer Society, 2020, pp. 1875–1884. doi: 10.24251/hicss.2020.231.
[19] CSCI 2022 2022 International Conference on Computational Science and Computational Intelligence : proceedings : 14-16 December 2022, Las Vegas. The Institute of Electrical and Electronics Engineers, 2022.
[20] “(PDF) “Oh, look, a butterfly!" A framework for distracting attackers to improve cyber defense.” Accessed: Jun. 21, 2023. [Online]. Available: https://www.researchgate.net/publication/326561335_Oh_look_a_butterfly_A_framework_for_distracting_attackers_to_improve_cyber_defense
[21] M. S. I. Sajid et al., “SODA: A System for Cyber Deception Orchestration and Automation,” in ACM International Conference Proceeding Series, Association for Computing Machinery, Dec. 2021, pp. 675–689. doi: 10.1145/3485832.3485918.
[22] M. M. Islam, A. Dutta, M. S. I. Sajid, E. Al-Shaer, J. Wei, and S. Farhang, “CHIMERA: Autonomous Planning and Orchestration for Malware Deception,” 2021 IEEE Conference on Communications and Network Security, CNS 2021, pp. 173–181, 2021, doi: 10.1109/CNS53000.2021.9705030.
[23] Y. T. Huang, Y. S. Sun, and M. C. Chen, “TagSeq: Malicious behavior discovery using dynamic analysis,” PLoS One, vol. 17, no. 5, p. e0263644, May 2022, doi: 10.1371/JOURNAL.PONE.0263644.
[24] Q. Duan, E. Al-Shaer, M. Islam, and H. Jafarian, “CONCEAL: A strategy composition for resilient cyber deception-framework, metrics and deployment,” 2018 IEEE Conference on Communications and Network Security, CNS 2018, Aug. 2018, doi: 10.1109/CNS.2018.8433196.
[25] “GitHub - EasyHook/EasyHook: EasyHook - The reinvention of Windows API Hooking.” Accessed: Jun. 21, 2023. [Online]. Available: https://github.com/EasyHook/EasyHook
[26] “VirusTotal API v3 Overview.” Accessed: Jun. 21, 2023. [Online]. Available: https://developers.virustotal.com/reference/overview
[27] “MalShare.” Accessed: Jun. 21, 2023. [Online]. Available: https://malshare.com/
[28] M. N. Alsaleh, J. Wei, E. Al-Shaer, and M. Ahmed, “gExtractor: Towards Automated Extraction of Malware Deception Parameters,” 2018, doi: 10.1145/3289239.3289244.
[29] S. Jamalpur, Y. S. Navya, P. Raja, G. Tagore, and G. R. K. Rao, “Dynamic Malware Analysis Using Cuckoo Sandbox,” Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018, pp. 1056–1060, Sep. 2018, doi: 10.1109/ICICCT.2018.8473346.

[30] M. Akiyama, T. Yagi, T. Yada, T. Mori, and Y. Kadobayashi, "Analyzing the ecosystem of malicious URL redirection through longitudinal observation from honeypots," Comput Secur, vol. 69, pp. 155–173, Aug. 2017, doi: 10.1016/J.COSE.2017.01.003.
[31] X. Han, N. Kheir, and D. Balzarotti, "Evaluation of deception-based web attacks detection," MTD 2017 - Proceedings of the 2017 Workshop on Moving Target Defense, co-located with CCS 2017, vol. 2017-January, pp. 65–73, Oct. 2017, doi: 10.1145/3140549.3140555.
[32] X. Han, N. Kheir, and D. Balzarotti, "Evaluation of deception-based web attacks detection," in MTD 2017 - Proceedings of the 2017 Workshop on Moving Target Defense, co-located with CCS 2017, Association for Computing Machinery, Inc, Oct. 2017, pp. 65–73. doi: 10.1145/3140549.3140555.
[33] M. Akiyama, T. Yagi, K. Aoki, T. Hariu, and Y. Kadobayashi, "Active credential leakage for observing web-based attack cycle," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8145 LNCS, pp. 223–243, 2013, doi: 10.1007/978-3-642-41284-4_12/COVER.
[34] "GitHub - greim/hoxy: Web-hacking proxy API for node." Accessed: Jun. 21, 2023. [Online]. Available: https://github.com/greim/hoxy
[35] K. Ferguson-Walter, M. Major, D. Van Bruggen, S. Fugate, and R. Gutzwiller, "The world (of CTF) is not enough data: Lessons learned from a cyber deception experiment," Proceedings - 2019 IEEE 5th International Conference on Collaboration and Internet Computing, CIC 2019, pp. 346–353, Dec. 2019, doi: 10.1109/CIC48465.2019.00048.
[36] B. Ksiezopolski, K. Mazur, M. Miskiewicz, and D. Rusinek, "Teaching a Hands-On CTF-Based Web Application Security Course," Electronics 2022, Vol. 11, Page 3517, vol. 11, no. 21, p. 3517, Oct. 2022, doi: 10.3390/ELECTRONICS11213517.