Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Three decades of deception techniques in active cyber defense - Retrospect and outlook

*Li Zhang**, *Vrizlynn.L.L. Thing*

Cybersecurity Strategic Technology Center, ST Engineering, 609602, Singapore

ARTICLE INFO

Article history:
Received 3 June 2020
Revised 28 January 2021
Accepted 4 April 2021
Available online 18 April 2021

Keywords:
Cyber defense
Deception techniques
Honeypots
Honeytokens
Moving target defense
Computer network defense

ABSTRACT

Deception techniques have been widely seen as a game changer in cyber defense. In this paper, we review representative techniques in honeypots, honeytokens, and moving target defense, spanning from the late 1980s to the year 2021. Techniques from these three domains complement with each other and may be leveraged to build a holistic deception based defense. However, to the best of our knowledge, there has not been a work that provides a systematic retrospect of these three domains all together and investigates their integrated usage for orchestrated deceptions. Our paper aims to fill this gap. By utilizing a tailored cyber kill chain model which can reflect the current threat landscape and a four-layer deception stack, a two-dimensional taxonomy is developed, based on which the deception techniques are classified. The taxonomy literally answers which phases of a cyber attack campaign the techniques can disrupt and which layers of the deception stack they belong to. Cyber defenders may use the taxonomy as a reference to design an organized and comprehensive deception plan, or to prioritize deception efforts for a budget conscious solution. We also discuss two important points for achieving active and resilient cyber defense, namely deception in depth and deception lifecycle, where several notable proposals are illustrated. Finally, some outlooks on future research directions are presented, including dynamic integration of different deception techniques, quantified deception effects and deception operation cost, hardware-supported deception techniques, as well as techniques developed based on better understanding of the human element.

1. Introduction

In his book *The Art of Deception* (Mitnick et al., 2007), Kevin Mitnick, the world's most infamous hacker, asserted that the human element is security's weakest link. By attacking this link through various deception based social engineering techniques such as pretexting and phishing, cyber criminals have achieved wide success. For instance, according to 2019 Verizon data breach investigations report 201 (0000), phishing attacks accounted for more than 80% of reported security incidents. In the COVID-19 pandemic, we have also witnessed the enormous quantity of cases where hackers exploited coronavirus fears to deliver their phishing and malware attacks Cimpanu (2020a,b); Palmer (2020).

Deception aims to manipulate humans' perception by exploiting their psychological vulnerabilities Cybenko et al. (2002), which has direct impact on their beliefs, decisions, and actions. It can be a powerful tool for both hackers and cyber defenders. In as early as the late 1980s, Clifford Stoll managed to set up an imaginary computer environment (now known as *honeypot*), in which a fictitious account was created along with a number of fake documents with enticing names, to lure a hacker to reveal

* Corresponding author. .
E-mail addresses: zhang.li@stengg.com (L. Zhang), vrizlynn.thing@stengg.com (Vrizlynn.L.L. Thing).
https://doi.org/10.1016/j.cose.2021.102288

himself and his objectives Stoll (1989). In the battle between the hacker and the cyber defender, a conventional wisdom is that the offense has the upper hand: cyber defenders have to make sure everything is properly maintained and prevent intrusions at every single point, whereas hackers may just need to take advantage of one vulnerability to breach the defense Lynn III (2010). At the same time, attackers can always gain knowledge about a target system or network through a variety of reconnaissance and discovery tactics, while defenders are usually short of intelligence about their adversaries. Such asymmetric disadvantage for cyber defenders is well promised to be re-balanced through the use of defensive deception, which is expected to deliver a game-changing impact on how threats are facedFerguson-Walter et al. (2019); Lawrence Pingree (2015); Shade et al. (2020).

The perimeter-based defense strategy utilizing conventional security measures such as firewalls, authentication controls, and intrusion prevention systems (IPS) has been proven feeble against infiltration. Even with the defense-in-depth strategy U.S. Department of Homeland Security (2016), where multiple layers of the conventional security controls are placed throughout the target network, cyber defenders still find it hard to prevent and detect sophisticated attacks like Advanced Persistent Threat (APT) based intrusions. Such targeted attacks typically exploit zero-day vulnerabilities to establish footholds on the target network and leave very few traces of their malicious activities behind for detection. Besides, conventional anomaly detection solutions such as intrusion detection systems (IDS) and behavior based malware scanners tend to raise an overwhelming number of false positive alerts, which plagues cyber defenders and hurts their efficacy in identifying and responding to the true attacks. Defensive deception, featured by its capability of detecting zero-day vulnerabilities and its low false alarm rates due to a clear line between legitimate user activities and malicious interactions, can act as an additional layer of defense to mitigate the issues.

Instead of focusing on attackers' actions, defensive deception works on their perception by obfuscating the attack surface. The objective is to hide critical assets from attackers and confuse or mislead them, thereby increasing their risk of being detected, causing them to misdirect or waste resources, delaying the effect of attacks, and exposing the adversary tradecraft prematurely Ross et al. (2019). In other words, defensive deception helps establish an *active cyber defense* posture, wherein the key elements are to anticipate attacks before they happen, to increase the costs of the adversary, and to gather new threat intelligence for preventing similar attacks.

Since Stoll's honeypot, there have been numerous honeypots of different flavors proposed. These honeypots can be classified from different perspectives, such as whether they are server-based or client-based, of low interaction or high interaction, and based on real systems or virtual machines (VMs). Despite of the various flavors, all the honeypots share the same definition of being a security resource whose value lies in being probed, attacked, or compromisedSpitzner (2002). The term *honeypot* typically refers to decoy computer systems. Multiple interconnected honeypots form a honeynet. For bait resources that are of other forms (e.g., accounts, user files, database entries, and passwords), they can be collectively termed as *honeytokens* Augusto Paes de Barros (2003); Lance Spitzner (2003). Take the honeyfiles proposed in Yuill et al. (2004) as an example. These spurious files reside on a file server; once they are accessed, the server will send an alarm to alert a possible intrusion. Honeypots and honeytokens, when used in tandem, can introduce multi-tier fake attack surfaces for intruders. Unless the intruder can correctly select his target at every turn, his maneuver will be detected.

Nevertheless, if the honeypots and honeytokens are left with static deployment and configurations, the adversary will have enough time to infer their existence, map out them, and in turn evade them. Even worse, honeypots, especially the high-interaction ones which offer the intruder a real Operating System (OS) environment to interact with, may be exploited by the intruder to gain privileged control and used as a pivot point to compromise other systems Lance Spitzner (2004). This is where the moving target defense (MTD) comes into the picture, which was identified as a key cybersecurity R&D theme by U.S. NITRD Program NIT (2010). Specifically, MTD techniques accomplish defensive deception through randomization and reconfiguration of networks, assets, and defense toolsPawlick et al. (2019). By dynamically shifting both the real and fake attack surfaces, the attack surfaces of critical assets can be maximally obfuscated, with the attacker continuously confused and misled. For instance, Cohen reported in Fred Cohen (2010) that a combination of MTD techniques and honeytokens (e.g., automated responses on all unused ports), can help achieve long-term effectiveness of deceptions.

In this paper, we present a systematic review on the three aspects of defensive deception techniques (i.e., honeypots, honeytokens, and MTD) that have been proposed in the past three decades. The aim is to facilitate a better understanding of the advancement in each aspect and provide clues on how to better integrate them to build a holistic and resilient deception based defense. We limit our scope to techniques that can be directly applied to counter network intrusions. For example, the client-side honeypots Nazario (2009); Seifert et al. (2007), which manifest as vulnerable user agents and actively troll malicious servers to study the client-side attacks, will be excluded in our survey. Sophisticated cyber attacks usually involve phased progressions, and an effective defense should be designed to disrupt each phase of the attack lifecycle. In view of this, the surveyed methods are classified based on the attack phases where they can be applied as countermeasures. In particular, we will use our proposed cyber kill chain model, which is specifically developed to model the network intrusion end to end and can reflect the current threat landscape. In each unique phase of the kill chain model, we further categorize the defensive deception methods according to a four-layer deception stack Lawrence Pingree (2015) composed of the network, system, software, and data layers. Such a two-dimensional taxonomy can be employed as a reference for deciding what techniques can be used to disrupt which attack stages and what techniques can complement with each other.

There have been some excellent surveys on cyber defensive deception. However, most of them just focus on either honeypots and honeytokens Christian Seifert et al. (2006); Efendi et al. (2019); Han et al. (2018); Nawrocki et al. (2016);

Scott Smith (2016) or MTD techniques B.C. Ward et al. (2018); Cai et al. (2016); Lei et al. (2018); Okhravi et al. (2013); Sengupta et al. (2020). Although the survey of deception technology in Fraunholz et al. (2018) includes MTD techniques, its main focus is on honeypots and honeytokens; MTD techniques are just briefly mentioned and not systematically reviewed. In Pawlick et al. (2019), twenty-four articles that use game theory to model defensive deception, comprising honeypots, honeytokens, and MTD techniques, are surveyed. Despite of representing an important direction, these game-theoretic models are just a small part of the literature. To the best of our knowledge, there has not been a work that provides a systematic retrospect of honeypots, honeytokens, as well as MTD techniques and investigates their integrated usage for orchestrated deceptions. Our survey aims to fill this gap.

The remainder of this paper is organized as follows. The proposed cyber kill chain model is illustrated in Section II, while the survey method is presented in Section III. Representative honeypots, honeytokens, and MTD techniques that can disrupt the adversary kill chain are reviewed in Section IV to VI, respectively. Section VII discusses how to use the deception techniques to achieve active and resilient cyber defense from two aspects, i.e., deception in depth and deception lifecycle. Finally, the paper is concluded in Section VIII with reflection and outlook of defensive deception research.

2. Cyber kill chain model

A sophisticated cyber attack typically has to go through multiple consecutive phases before accomplishing its objective, be it stealthy collection and exfiltration of sensitive data or violation of critical assets' integrity or availability. The Lockheed Martin's intrusion kill chain Hutchins et al. (2011) has been widely applied to assist structured analyses of the phased progressions. Nonetheless, this kill chain model, which consists of seven phases (i.e., *reconnaissance, weaponization, delivery, exploitation, installation, command & control (C2)*, and *actions on objectives*), is often criticized for reinforcing the perimeter-focused thinking and failing to cover the attack paths inside the network perimeter Giora Engel (2014); Reidy (2013). There have been several efforts aimed at expanding it for improved coverage. For instance, to explicitly model the intruder's movement from an initially compromised system to the target system, Laliberte Marc Laliberte (2016) proposes to add a *lateral movement* phase between the *C2* and *action on objectives* phase. Besides, Laliberte removes the *weaponization* phase as it happens outside of the victim network and no security measure can directly defend against it. The removal of the *weaponization* phase is in line with the kill chain model proposed in Blake D. Bryant and Hossein Saiedian (2017), which further introduces the *privilege escalation* and *exfiltration* phase. By contrast, the kill chain models proposed in Malone (2016); Pols (2017) significantly expands the Lockheed Martin's kill chain model. Both of them divide the adversary kill chain into three sub-kill chains, i.e., the external kill chain aiming to establish an initial foothold, the internal kill chain aiming to propagate inside the victim network, and the target manipulation kill chain aiming to manipulate the target system to achieve attack objectives. In particular, the kill chain model in Pols (2017) includes four tactics from the MITRE ATT&CK model[1], namely *defense evasion, credential access, execution*, and *collection*, as additional attack phases.

Our proposed cyber kill chain model, shown in Fig. 1, is developed based on Malone (2016); Pols (2017). The purpose is to address some of their limitations. For example, there is only one attack objective in Malone (2016), represented by the final *execution* phase (i.e., activating malware to subvert operations of the target system). This omits many other possible impacts that a threat actor may incur. Besides, defense evasion dominated the attack tactics in 2019 Kelly Sheridan (2020), which we believe should be explicitly modeled in the kill chain. However, it is missing in Malone (2016). Regarding the kill chain model in Pols (2017), we think its *weaponization* and *pivoting* phases are superfluous. For the former, we are in consonance with Blake D. Bryant and Hossein Saiedian (2017); Marc Laliberte (2016) that it is not actionable to cyber defenders; for the latter, its actions and benefits actually have already been encompassed by the *lateral movement* phase. In addition, the MITRE ATT&CK model is recently supplemented with a new class of adversary tactics, i.e., *impact*. The inclusion of an *impact* phase in a kill chain model will enable it to model attack objectives more comprehensively.

In the proposed kill chain model, there are also three sub-kill chains, whose intents have been described above. Each modeled sub-kill chain starts with a *reconnaissance* phase as it provides the attacker with crucial information (such as network topology, vulnerabilities, and deployed security tools) to move further along the kill chain. All the three sub-kill chains also typically contain the *defense evasion* phase due to the widely adopted defense-in-depth strategy. No matter where the intruder propagates, he has to ensure that his maneuver is not detected by defensive measures. The *installation* phase in the external kill chain embodies both the *persistence* and *execution* tactics in the MITRE ATT&CK model. The internal kill chain for network propagation may be repeated for several times before the attacker finally reaches the target system. In the target manipulation kill chain, a combination of the *collection* and *exfiltration* phases is used to model the attacker's action of covertly stealing sensitive data, while the *impact* phase is used to represent the action of manipulating, interrupting, or destroying the critical assets.

With the intrusion activities covered end to end, the proposed kill chain model can be used to guide attack analyses, threat intelligence extraction, as well as defensive measures selection and prioritization. Such an intelligence-driven, threat-focused approach is essential to establish the active cyber defense posture against threats from both external actors and malicious insiders. For example, synthesis of the remaining kill chain of a detected attack may reveal a zero-day exploit Hutchins et al. (2011). This yields insights into possible future attacks and thereby drive the defender to implement countermeasures beforehand.

To facilitate the coordinated selection and deployment of deception techniques, the honeypot, honeytoken, and MTD techniques to be reviewed will be mapped to the proposed kill chain model's unique attack phases as listed in Table 1. Based

[1] https://attack.mitre.org/

Table 1 – Unique attack phases in the proposed cyber kill chain model.

		Deception Layer			
Attack Phase	Description	Network	System	Software	Data
Reconnaissance	Gather information from open-source intelligence, internet-facing/internal system probing, or network traffic sniffing	✓	✓	✓	✓
Delivery	Deliver the tailored malicious payload through entry vectors like email, websites, and removable media	✓	✓	✓	
Defense Evasion	Evade defense by disabling security tools, abusing trusted process to hide malware, obfuscation/encryption, etc.	✓	✓	✓	
Exploitation	Exploit identified vulnerabilities or simply user negligence to facilitate the following kill chain phases	✓	✓	✓	
Installation	Drop the payload (e.g., install a remote access Trojan or backdoor to maintain the foothold)		✓		
C2	Beacon outbound to establish a C2 channel with attacker for additional commands or payloads	✓	✓		
Privilege Escalation	Horizontally or vertically elevate privileges to gain access to sensitive assets		✓	✓	
Credential Access	Steal credentials like account names and passwords		✓	✓	✓
Lateral Movement	Move through the network by using installed remote access tools or stolen credentials	✓	✓	✓	✓
Collection	Collect data of interest from local databases/file system, screen/user input capture, shared network drives, etc.	✓	✓	✓	✓
Exfiltration	Process collected data and send it out through C2 channel or other channels	✓	✓		
Impact	Disrupt availability or compromise integrity of critical data or system/network services	✓	✓	✓	✓

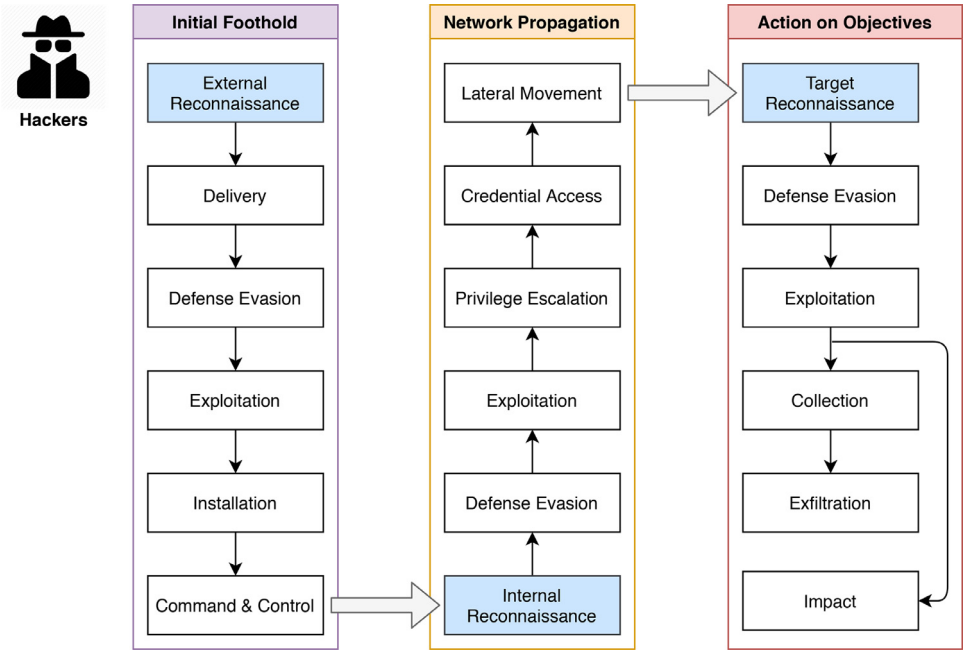

Fig. 1 – The proposed cyber kill chain model.

on the characteristics of each unique attack phase, the possible deception layers, in which the specific attack phase may be disrupted, are ticked accordingly.

3. Survey method

To build the paper repository for this survey, we firstly searched in two leading research databases (i.e., IEEE Xplore and ACM Digital Library) with the keyword *cyber deception*, which returned a list of 253 research articles. Then we utilized the title and abstract of each paper to determine its relevance to our survey. The selection criteria is whether the paper is on defensive cyber deception (including evasion techniques of defensive deception). For example, papers on cyber deception attacks Amin et al. (2013); Hou et al. (2020); Meira-Goes et al. (2021); Zhang et al. (2020) were removed. This reduced the list to 87 research works. By including relevant papers cited in these works, we managed to collect additional articles. Together with some writings that we saw in other venues or mediums and think important to our survey (e.g., Ph.D. dissertations available online), the final repository contains 192 research works, which are referred to as primary studies in a systematic survey Kitchenham (2004). Although our paper repository does not cover all the related papers that were published in the past three decades, we are confident that most representative deception techniques have been included, through which the overall development trends in this field are accurately pictured.

4. Honeypots

Honeypots can be broadly classified into two categories: research and productionLance Spitzner (2001). Although research honeypots play an important role in gathering intelligence on the threat landscape, they do not directly benefit a specific organization. In contrast, production honeypots are placed in an organization's environment for attack detection and risk mitigation. They may be deployed as sacrificial lamb, hacker zoo, minefield, proximity decoys, redirection shield, and deception ports (on production systems) Scottberg et al. (2002), as described in Table 2.

Stoll's honeypot Stoll (1989) is a good example of the sacrificial lamb, which is the oldest and maybe also the most intuitive strategy. Being usually isolated from production systems, the sacrificial lamb honeypot may be easily identified and bypassed by attackers. The same limitation is shared by the hacker zoo. The minefield honeypots are commonly placed near the network perimeter, which will sound alarms upon attacker probing. This strategy helps enhance the perimeter based defense, but cannot handle attackers already inside the network. Both the proximity decoys and the redirection shield aim to lead the attacker astray and away from production systems. Their difference lies in that the redirection shield strategy, through the use of traffic rerouting or port redirection, does not require honeypots to be in the production network and hence has more flexibility. Among the five honeypot deployment strategies, the deception ports on production systems can be seen as the final defense. The various simulated vulnerable services on well-known ports can be used to detect and delay the attack even if the adversary reaches the production system. For example, the Deception Toolkit Fred Cohen (1998), which is the first open source honeypot, can set up the deception services.

Featured by deceiving to detect, derail and/or delay attacks, honeypots may be used to disrupt a number of attack phases in the cyber kill chain model, namely the reconnaissance, delivery, exploitation, installation, C2, lateral movement, and im-

Table 2 – Common Deployment Strategies for Honeypots Scottberg et al. (2002).

Strategy	Description
Sacrificial Lamb	An isolated system that has no entry point to production systems
Hacker Zoo	An entire subnet of honeypots with varied platforms, services, vulnerabilities, and configurations, which are isolated from production systems
Minefield	A number of honeypots placed in forefront to serve as first attack targets
Proximity Decoys	Honeypots deployed in close proximity to production systems
Redirection Shield	External honeypots that appear on production systems through port redirection
Deception Ports	Simulated services (e.g., SMTP, DNS, FTP) on production systems

pact phase. On the other hand, in the defense evasion phase, the attacker may be able to identify honeypots and evade them. The remaining part of this section will be on these two aspects.

4.1. *Disrupting the cyber kill chain*

The intruder relies on successful reconnaissance to achieve tactical advantage in the campaign. Sticky honeypots can be used to mitigate the threat from network scans. For example, LaBrea Tom Liston (2001) can take over unused IP addresses in the network and create virtual hosts to attract worms and hackers; connection attempts to the impersonated hosts will then be tarpitted. Greasy Leslie Shing (2016) further improves the sticky connection parameters to generate more realistic traffic. Besides slowing down the scanning activities, both LaBrea and Greasy are able to produce false network topologies and hence get adversaries confused. To dissimulate the network topology, many other honeypot techniques can also be used. For instance, HoneyD Provos (2004) can simulate a large number of virtual systems with configurable fingerprints and provide arbitrary services and routing topologies. These honeypot techniques are typically of low interaction and virtually adopt the minefield or proximity decoys deployment strategy.

To disrupt the attack phases such as delivery, C2, and lateral movement, the key is to direct the malicious traffic to high-interaction honeypots. By providing high-fidelity forged environment to interact with attackers, the exploitation, installation, and impact phases may also be broken. In addition, attackers' time and resources will be wasted and their tactics, techniques, and procedures (TTPs) may be revealed. As these high-interaction honeypots typically monitor one IP address each and have the problem of limited field of view, the redirection shield strategy is often adopted. In Anagnostakis et al. (2005), it is proposed to handle anomalous traffic identified by IDS by a shadow honeypot, as shown in Fig. 2. The shadow honeypot is an instrumented instance of the application (e.g., transactional applications) in protected system and share all internal states. Attacks mounted in the shadow honeypot will be caught and the induced state changes will be discarded, while legitimate traffic misclassified by IDS will be validated in the shadow honeypot and

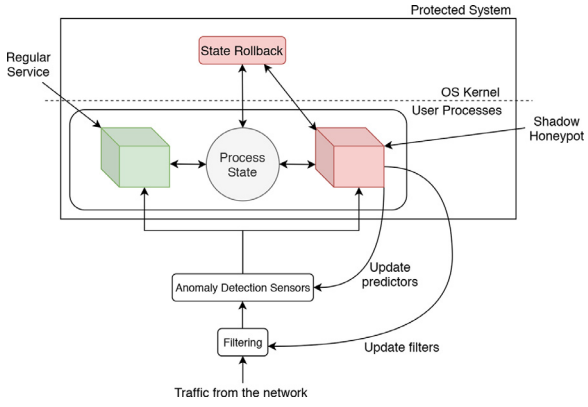

Fig. 2 – The Shadow Honeypot architecture in Anagnostakis et al. (2005).

transparently handled. OpenFire Borders et al. (2007) presents additional false targets by appearing to attackers that all IPs and ports of an organization network are open. Suspicious traffic will then be forwarded to a cluster of decoy machines. In the cloud environment, Biedermann *et al.* Biedermann et al. (2012) propose to redirect potential attacks against an operational VM to a honeypot VM created through a live cloning process. The honeypot VM has exactly the same configuration as the original VM, but without the sensitive data. This way, the impact of the attack can be analyzed without risking the integrity of the original target VM. Similarly, in Urias et al. (2016), the endpoint VM suspected of malicious activities will be cloned and forked in a deception environment with the same network and system configurations of the real network environment. If the suspicions for the VM are not found, the VM may be migrated back to the operational environment; otherwise, all artifacts related to the attack in the deception environment can be documented for further scrutiny.

Besides standard IT systems, the above honeypot concepts are also applicable to industrial control systems (ICS). In Disso et al. (2013), after analysing the threat landscape and unique security requirements of supervisory control and data acquisition (SCADA) systems, a plausible honeypot sys-

tem is built, which is composed of both a low-interaction HoneyD honeypot emulating the programmable logic controller (PLC) and a high-interaction honeypot using a genuine PLC. In Winn et al. (2015), HoneyD is extended to address the authenticity flaw of emulated PLCs; together with the proxy technology, multiple high-interaction honeypots can be distributed at the cost of a single actual PLC. A number of recent honeypot applications in ICS are based on Conpot Lukas Rist et al. (2015), which is a low-interaction virtual ICS honeypot designed for easy deployment, modification and extension and supports a range of common industrial control protocols such as Modbus TCP, SNMP, and BACnet. In Zhao and Qin (2017), a high-interaction honeypot is created by improving Conpot in the aspects of control protocol, human-machine interface (HMI) and equipment simulation. In Kuman et al. (2017), with the use of Conpot and the IMUNES network simulator, a complex high-interaction ICS is emulated.

4.2. *Evading the honeypot*

Despite of being a powerful tool to trap, delay, and even gather information about intruders, honeypots have their own weakness. At best, they are counterfeits of the real target. If intruders are able to identify honeypots, they will circumvent them or keep the malicious payload dormant, making honeypots useless. To some extent, attackers are highly motivated to push the detection of honeypots to early phases of their kill chain, so that their intrusion efforts are not rendered in vain and their TTPs are not disclosed.

Honeypots may be fingerprinted based on timing or behavior discrepancies in probing responses. After the introduction of the seminal HoneyD, it was soon found that it can be remotely fingerprinted based on its response to bad packets Joseph Corey (2003) or the latency of its emulated network links Fu et al. (2006). Degreaser in Alt et al. (2014) can efficiently fingerprint sticky honeypots like LaBrea by sending a series of specially crafted probe packets; real hosts can then be discerned from tarpits based on the response. In Vetterl and Clayton (2018); Vetterl (2019), by leveraging the flaw of many honeypots' reliance on off-the-shelf libraries to implement the transport layer, distinguishing probes constructed at this layer is able to systematically fingerprint honeypots.

Other unique features of honeypots may also be taken advantage of by attackers. Honeypot evader Rrushi (2019) exploits honeypots' innate characteristic of not initiating any network traffic and attacks only the hosts with obvious network activity. In Wang et al. (2010), by exploiting the liability constraint that cyber defenders cannot allow their honeypots to participate in real attacks that could cause damage to other entities, an attacker can detect honeypots by checking whether his compromised machines can successfully send out unmodified malicious traffic. A more specific example of this concept is given in Krawetz (2004), where spammers can simply check if an open proxy relay is a honeypot based on whether emails can be sent to themselves.

Besides exploiting a single factor to tell whether a target is honeypot, information collected from different factors may be combined to reach a more accurate decision. In Hayatle et al. (2012), such combination is performed with Dempster-Shafer theorySentz et al. (2002), while in Huang et al. (2019), machine learning (ML) techniques are used. For the latter, the design is depicted in Fig. 3.

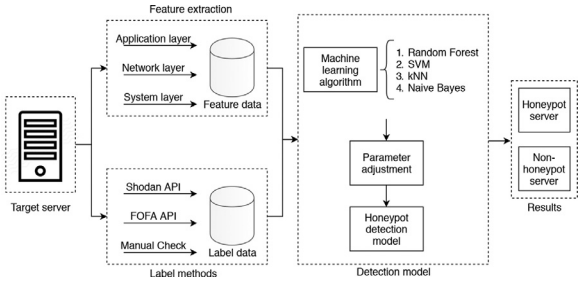

Fig. 3 – The machine learning based Honeypot server identification method in Huang et al. (2019).

5. Honeytokens

Honeytokens share the same concept of honeypots, whose value lies in being used illicitly. In fact, the history of honeytokens is as long as that of honeypots. Besides the fictitious files with tempting names and contents in Stoll's honeypot in the late 1980s Stoll (1989), Spafford built files with Unix sparse file structure in 1990s Eugene H. Spafford (2011), which are of small size on disk but will result in "endless" transfer for attacker's copy attempt.

Honeytokens have remarkable flexibility. They can be in the form of any digital entity and placed anywhere across an organization's environment. The polymorphism and omnipresence bring two benefits for the defense: even though attackers are able to evade some forms of honeytokens, they may still be trapped by others; the uncertainty of whether and where honeytokens are placed will slow down attackers and may even turn them away (i.e., the deterrent effect). Typically, the honeytoken is simple to deploy and cost effective, making it considered as an exciting new dimension for honeypot Lance Spitzner (2003).

Unlike honeypots which usually can only disrupt specific attack phases in the kill chain from the network and system layer, the various forms of honeytokens can be applied to thwart almost all the attack phases through all the four layers of the deception stack.

In the external reconnaissance phase, before engaging the target network, attackers will actively gather information from open-source intelligence. For example, Project Spacecrab creates credential honeytokens in the form of Amazon Web Services (AWS) keys, and found that the average time for a hacker to exploit the honeytoken is just thirty minutes after it is posted on GitHub Bourke and Grzelak (2018). There is a wealth of personal information on social network platforms, from which attackers might find valuable tips to drive targeted phishing campaigns. If bogus profiles are disseminated on these platforms Stringhini et al. (2010), attackers may be misdirected in the delivery attack phase. To increase the authenticity, Virvilis *et al.* Virvilis et al. (2014) suggest that the created fake personas should have positions of interest to attackers, connections with people from both inside and outside the organization, valid email addresses, as well as real,

but closely monitored, organization accounts. To facilitate the creation of the bogus profiles, a method for automatically generating realistic personally identifiable information (PII) based honeytokens is proposed in White (2010).

The internet-facing web servers of an organization is another important intelligence source in external reconnaissance. To confuse and misdirect malicious website visitors, decoy hyperlinks embedded in web pages are used in Gavrilis et al. (2007). These hyperlinks are invisible to legitimate human users, but can be detected by automated programs. An algorithm is also proposed for optimal placement of the decoys in website pages. Brewer *et al.* Douglas Brewer et al. (2010) propose to add the decoy links under two design principles: the multiple-link principle where multiple decoy links are positioned off the visible page and valid links remain in their original; the shadow-link principle where multiple, invisible decoy links are stacked at the same coordinates as the valid link. In Virvilis et al. (2014), three types of honey tokens are proposed for public web servers: fake entries in `robots.txt` files (used to tell crawlers which web pages to crawl and which ones not to), invisible decoy links as described above (e.g., white links with white font), and fake credentials in HTML comments.

When attackers probe the target network for more information, deceptive responses can be utilized to confuse them and delay their progress. To conceal the operating system (OS) related information that may be retrieved by attackers via OS fingerprinting, host-based OS obfuscation is suggested in Murphy et al. (2010) as a deception technique. With the attacker being unsure of the OS or even assuming the wrong OS, his penetration will be impeded. In Trassare (2013), the defender controls the fake routes to be presented to attackers who use traceroute to map the target network's topology. Instead of directly rejecting the connection after an attack is suspected, which either is a false positive or will inform the adversary of being detected, deceptive delays are suggested in Julian (2002); Neil C. Rowe (2007). The defender can use excuses (e.g., a computation requires a long time) to keep the suspect waiting, and use the time to collect more evidence or reorganize the defense. Katsinis and Kumar Katsinis and Kumar (2012, 2013) propose to deploy honeytokens such as fake form fields, fake parameters, and fake files in the web server. Alarms from these honeytokens will be sent to a deception module, which is responsible for redirecting the attacker traffic to a honeypot and supplying the attacker with misinformation that his attack is successful. By leveraging the modular design of Apache web server, the deception module can be conveniently inserted between the metadata processor and the content generator, as shown in Fig. 4. A similar framework for achieving deceptive response is proposed in Han et al. (2017), where the deception module is deployed as a transparent reverse proxy.

A vital part of the attacker kill chain is to bypass the defense in the target network. Taking advantage of attackers' fear of having their TTPs exposed and resources wasted, Rowe *et al.* Rowe et al. (2007) propose to plant clues in systems such that they appear as honeypots (i.e., fake honeypots) and thereby turn attackers away. The planted clues can be names of known honeypot tools, non-standard system calls in security critical subroutines, reduced number of common files, and

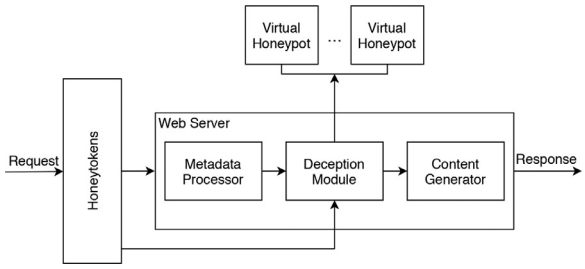

Fig. 4 – The deceptive web server in Apache architecture in Katsinis and Kumar (2013).

appearance of the system being little used. Besides fake honeypots, the deception effects on attackers of "fake fake honeypots", which refer to real honeypots that pretend to be noticeable fake honeypots, are also investigated.

Exploitation is another imperative phase in the attacker kill chain. Only after successfully exploiting some vulnerabilities can the adversary gain escalated privilege and be able to move further in the kill chain. In Crane et al. (2013), "booby trap" codes are inserted into the protected software or system during compilation or program loading. These booby traps remains dormant under normal operation but may be triggered by attackers' exploitation attempts. Once triggered, the booby trap can perform advanced forensics to identify the attack in real time and send attackers deceptive responses. Frederico *et al.* Araujo et al. (2014) propose to use decoy vulnerabilities that have been patched as honeytokens (aka honeypatches). In particular, the vulnerabilities are patched in such a way that attackers' exploitation attempts appear successful but their connections are actually redirected to an ephemeral honeypot with the un-patched version of the system or software. Besides, the honeypot may host a deceptive file system laced with disinformation to further deceive, delay, and misdirect attackers.

Attackers already inside the network will eavesdrop on the traffic to collect sensitive information and/or use the information to guide their following activities. For example, an attacker may map out systems that do not initiate any network traffic, which are likely to be honeypots, and circumvent them during the lateral movement. Such activity-guided target selection can be disabled by introducing decoy network and user space activities Rrushi (2019). Bowen *et al.* Bowen et al. (2010, 2012) propose to inject decoy traffic with enticing information that will induce the eavesdropper to take observable actions (e.g., using sniffed credentials to access a decoy account). In particular, to maximize the realism of the decoy traffic, a "record, modify, and replay" method (see Fig. 5) is used to automatically generate a large amount of decoy traffic; the decoy traffic is also continuously updated to prevent an adversary from recognizing the bait over time. On the other hand, encryption may be used to restrict access to sensitive information in the network traffic. However, the eavesdropper may still reveal the secret through offline brute-force attacks. As decryption with a wrong key will result in random gibberish, the adversary will know that he is successful if the output complies with some expected structure. To mitigate this risk, honey encryption (HE) Juels and Risten-

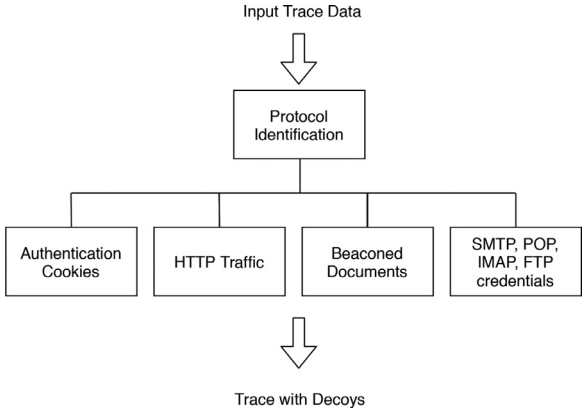

Fig. 5 – The "record, modify, and replay" process for decoy traffic generation in Bowen et al. (2012).

part (2014) can be used. When the ciphertext generated by HE is decrypted by an incorrect key, a plausible-looking but bogus plaintext will be yielded. The adversary will be confused and may be misdirected to reveal himself if the bogus plaintext is a credential honeytoken. To make the bogus plaintext in HE more deceptive, i.e., contextually correct and domain specific, natural language processing (NLP) based techniques Abiodun et al. (2020) and deep learning (DL) based onesOmolara et al. (2019) have been used.

In Juels (2014); Kaghazgaran and Takabi (2015), decoy permissions are used to extend role-based access control (RBAC) model for detecting the insider threat. These decoy permissions are not required for the specific roles to handle their tasks, and they are designed to give access to fake versions of sensitive assets. By monitoring attempts to access the fake assets, malicious users can be traced. We think that the decoy permissions are also useful for trapping outside attackers who have managed to infiltrate and reach the credential access attack phase. Legitimate users may know that they are not supposed to use the decoy permissions, but attackers who steal their credentials are not aware of that, leading to their activities being detected.

The following three categories of honeytoken techniques can be used to disrupt the last three attack phases in the kill chain model, namely the collection, exfiltration, and impact phase. As a result, threat actors may be hampered from achieving their objectives and their malicious activities may be detected.

Decoy passwords: Juels and Rivest Juels and Rivest (2013) propose to assign multiple false passwords (aka honeywords) along with the real password to each account. This way, even though the adversary manages to crack the passwords from the stolen password hash files, he is still not sure which passwords are real. If the honeywords are used for login, an alarm will be set off. Instead of using multiple fake passwords to protect an account, Almeshekah *et al.* Almeshekah et al. (2015) propose to use a machine-dependent function (e.g., a physically unclonable function (PUF) Chang et al. (2017) or a hardware security module (HSM) PCI (2009)) at the password server to generate "ersatzpasswords" from the stored password hashes; the hash of the ersatzpasswords are then stored in place of the original password hashes. This way, without physical access to the target's machine, any offline password cracking attempt will fail. If the attacker is unaware of the scheme and use the recovered ersatzpassword to login, the system administrator will be alerted.

Decoy database entries: Decoy database objects like `TABLE CREDIT_CARDS` or `VIEW EMPLOYEES_SALARY` can be inserted into databases to lure attackers. ys *et al.*Antanas Čenys et al. (2005) propose to implement modules for Oracle database management system (DBMS), which are responsible for monitoring access to the honeytokens, alerting the DBMS administrator, and logging malicious activities. To address the challenge of creating realistic decoy entries, HoneyGen Bercovitch et al. (2011) extrapolates rules describing the data structure, attributes, constraints and logic of real data items, and then automatically generates artificial items that comply with these rules. Padayachee Padayachee (2014) proposes to leverage aspect-oriented programming (AOP) to seamlessly augment a target DBMS with the basic honeytoken deployment processes, namely honeytoken generation, distribution, management, and detection.

Decoy user/system files: Yuill *et al.* Yuill et al. (2004) propose to use a honeyfile system to generate and monitor baits files; once these files are accessed, alerts will be sent to the system user. To ensure the detectability, Bowen *et al.* Bowen et al. (2009) propose to embed multiple signals in the decoy files, including a unique watermark that can be detected when the file is loaded in memory or appears in network traffic, a beacon that will signal a remote web site once the file is opened, and bait information such as credential honeytokens that will trigger alerts once used. To maximize the likelihood of an attacker taking the bait (i.e., conspicuousness), Voris *et al.* Voris et al. (2015, 2013) propose some automated deployment methods which can strategically place the decoy files. With the aim of increasing the enticingness of the decoys, NLP techniques are used in Ben Whitham (2017), where the fake file content is generated based on substitution and transposition of words collected from the target directory and file system. Existing file-based deception techniques mainly focus on decoy user data files, while PhantomFS Lee et al. (2020) proposes to use decoy system files. To prevent false alarms triggered by legitimate activities accessing the decoy system files, a hidden interface is introduced, through which the decoy files are excluded. As an attack has to invoke some system files, this approach can further improve the detection of the adversary, especially for disrupting the impact attack phase.

6. Moving target defense

Sun Tzu once wrote "just as water remains no constant shape, in warfare there are no constant conditions" Tzu (2007). Similarly, in cyber defense, a dynamic, constantly evolving attack surface for the protected network is extremely valuable to retain a resilient security posture. MTD techniques seek to randomize network components to reduce the likelihood of a successful attack, increase network dynamics to reduce the lifetime of an attack, and diversify otherwise homogeneous systems to limit the damage of a large-scale attack

Okhravi et al. (2014). In other words, MTD intensifies uncertainty and workload for attackers by making the protected network less static, less deterministic, and less homogeneous.

Similar to honeytokens, the various MTD techniques are able to disrupt the adversary kill chain through all the four layers of the deception stack.

In the external reconnaissance phase, attackers have to gain necessary knowledge about the target network before they can move on along the kill chain. This attack phase can be guarded against by obfuscating the following two aspects of network properties:

IP obfuscation: To prevent attackers from tracing hosts in the target network based on IP addresses, a number of techniques have been proposed. Two early examples are *dynamic network address translation* (DyNAT) Kewley et al. (2001) which is a protocol-obfuscation technique that can scramble source and destination IP addresses in packet headers and *network address space randomization* (NASR) Antonatos et al. (2005) which modifies a DHCP server to have short IP address leases so that host machines' IP addresses are changed frequently. Many recent techniques follow the line of randomly changing IP addresses. OpenFlow Random Host Mutation (OF-RHM) Jafarian et al. (2012) is able to mutate IP addresses with high unpredictability and rate. In particular, OF-RHM frequently assigns each host a random virtual IP (vIP) address, which will be automatically translated to/from the real IP (rIP) address of the host at the network edge. As a result, IP mutation is transparent to host machines and will not disrupt any active connection. To manage the random host mutation efficiently and minimizes the operational overhead, software-defined networking (SDN) Kreutz et al. (2015) is utilized, where a centralized approach is realized based on OpenFlow McKeown et al. (2008). A variant of the method, called Random Host Mutation (RHM), is proposed in Al-Shaer et al. (2013), which changes vIP addresses in a distributed fashion and can be deployed on traditional networks. Due to the IPv4 network's limited unoccupied address space, which reduces the unpredictability of IP address hopping, Dunlop *et al.* propose MT6D Dunlop et al. (2011, 2012). By leveraging the immense address space of IPv6, MT6D makes it harder for attackers to locate and subsequently target host machines. Besides, by encapsulating the original packet in a tunnel, MT6D also allows to change the IP address at any time without disrupting ongoing sessions.

OS obfuscation: To defend against OS fingerprinting attacks, Kampanakis *et al.* Kampanakis et al. (2014) propose an SDN based method, which hides the OS information in the response to detected illicit traffic by randomizing TCP sequence numbers and payload patterns in TCP, UDP, and ICMP protocols. Zhao *et al.* Zhao et al. (2017) propose to further model the interaction between the fingerprinting attack and defense as a signaling game Banks and Sobel (1987) and develop optimal fingerprint hopping strategies by analyzing the equilibriums of the game. A strategy selection algorithm is also proposed to maximize the defense utility.

The defense evasion phase may be disrupted by dynamically and continuously changing the placement of IDS over time. By creating uncertainty about the location of IDS, the likelihood of attackers' actions being detected will be increased. Venkatesan *et al.* Venkatesan et al. (2016) analyze the problem of deploying IDS across the network in a resource-constrained environment using a graph-theoretic approach and propose several deployment strategies based on centrality measures Friedkin (1991) that capture important properties of the network. Sengupta *et al.* Sengupta et al. (2018) model the same problem as a two-player general-sum Stackelberg game Korzhyk et al. (2010). Two scalable algorithms are designed to find the equilibrium of the game, which corresponds to optimal strategies for switching IDS placement that balance the overall security and usability. On the other hand, as many IDS have been based on artificial intelligence (AI) techniques Sinclair et al. (1999); Wang et al. (2018); Yin et al. (2017), there have been several adversarial attacks against the underlying AI models to induce misclassification Huang et al. (2018); Lin et al. (2019); Usama et al. (2019). The AI models may also adopt the moving target concept to improve the resilience against adversarial attacks, e.g., by randomizing the classification schemes Sengupta et al. (2019); Vorobeychik and Li (2014) as depicted in Fig. 6.

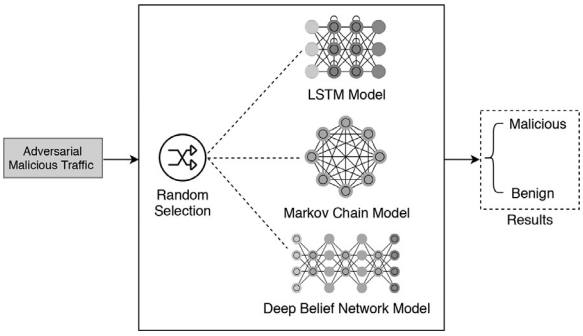

Fig. 6 – Randomized classifiers to mitigate adversarial attacks, where malicious traffic perturbed to evade the long short term memory (LSTM) model will be detected by other models.

The exploitation attack phase may be guarded against by various dynamic system and software techniques, which are also helpful to disrupt the impact attack phase:

Dynamic System: Among others, the most commonly used technique for increasing system dynamics is address space layout randomization (ASLR) PaX (2003); Li et al. (2006), which hinders the exploitation of memory corruption vulnerabilities by randomizing memory addresses of a loaded software. To address code-injection attacks, an instruction set randomization (ISR) technique is proposed in Kc et al. (2003), where an encoded version of software instructions is loaded into the memory and will be decoded by a key before being executed. Attackers' exploitation usually depend on vulnerabilities or characteristics of specific OS or CPU architectures. Thompson *et al.* Thompson et al. (2014) propose to enhance the security through a rotation of multiple OSs. Specifically, the method consists of several VMs equipped with different OSs. These VM hosts store shared data in a database and at one time only one of them will be mapped to an external IP address. The periodic rotation of VM hosts is controlled from an administrator machine running a daemon process, and the VM host that was previously in use is analyzed for evidence of intrusion and will be removed from rotation if compromised. Okhravi *et al.* Okhravi et al. (2011) propose a TALENT framework to im-

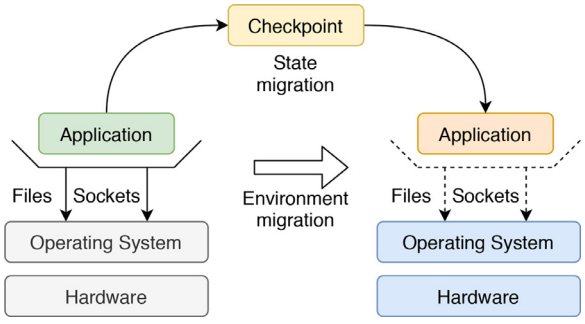

Fig. 7 – The TALENT migration process in Okhravi et al. (2011).

prove cyber survivability through platform diversity (i.e., different OSs and architectures). In TALENT, as depicted in Fig. 7, a running application can be migrated between VMs with different platforms while preserving the state (e.g., the execution state, open files and network connections). A portable checkpoint compiler is used to facilitate the application live migration process. Note that the migration among different platforms must take less time than the time needed for attacking a specific platform. Or else, the migration actually diminishes security because threat actors now have a choice of multiple platforms to attack Okhravi et al. (2014).

Dynamic Software: There is also a wide range of attacks exploiting software vulnerabilities, which requires precise understanding of the target software. By randomizing the implementation, software diversity introduces uncertainty in the target, increases the cost to attackers, and may provide an effective counter to side-channel attacks Larsen et al. (2014). ChameleonSoft Azab et al. (2011) proposes to divide a complex software program into smaller tasks, each of which has a set of executable variants that are functionally equivalent but with different quality attributes (e.g., performance, robustness, and mobility). The executable variants can then be shuffled to change the attack surface in accordance with different security situations. To defend against code reuse attacks, such as return-oriented programming (ROP), Gupta *et al.* Gupta et al. (2013) propose a fine-grained software diversity approach called Marlin. Marlin breaks a software binary into function blocks and randomly shuffles the order. Such a process can be performed transparently at load time, which ensures every execution instance of the software to be unique. On the other hand, to prevent a software program from being exploited by identified vulnerabilities, Le Goues *et al.* Le Goues et al. (2012) propose an automatic software repair method called GenProg. By utilizing an extended form of genetic programming, GenProg is able to evolve a software program with identified vulnerabilities to a functionally equivalent variant that are no longer susceptible to the previous risks. The dynamically patched software can be legacy programs without formal specifications and annotations.

To prevent attackers who are already inside the network from eavesdropping on communication flows, Germano da Silva *et al.* Germano da Silva et al. (2015) propose a multipath routing strategy, which relies on SDN features to frequently modify communication routes between SCADA devices. As each route transmits only a portion of the packets exchanged during the communication, even though the eavesdropper is well positioned in a strategic point of the network, he will not be able to intercept an entire communication between two devices. As the multipath routing strategy always relies on the shortest path to transmit the acknowledgment (ACK) packets from the receiver, Aseeri *et al.* Aseeri et al. (2017) found that an attacker can still capture all the packets by eavesdropping on the shortest path and blocking the ACK packet corresponding to the packet sent through other routes until it is retransmitted via the path he is listening to. To address this defect, the SDN controller can be utilized to instruct the receiver to send the ACK packet via the path used by the sender. In the self-shielding dynamic network architecture (SDNA) Yackoski et al. (2013, 2011), packets go through one or more intermediate devices before reaching the receiver. The intermediate devices are not simply routers; they also rewrite traffic to conceal the sender and receiver's identities. As a result, the eavesdropping attack can also be thwarted.

The collection phase in the kill chain may be disrupted by dynamic data approaches. To prevent the cryptographic keys stored in the cloud from being extracted by attackers using cross-VM side-channel attacks Zhang et al. (2012), Pattuk *et al.* Pattuk et al. (2014) propose to partition the keys into random shares based on the secret sharing and threshold cryptography Beimel (2011). The random shares are then stored in different VMs and will be regenerated periodically. As a result, the adversary has to attack multiple VMs to steal the key and the impact of a successful attack will be limited to a certain time period. On the other hand, dynamic data approaches may also impede the impact phase. Smutz and Stavrou Smutz and Stavrou (2015) propose to randomize the data block order of Microsoft office documents while keeping the visual interpretation intact. As malicious payloads embedded in the documents usually rely on a specific order of internal components, the randomization prevents them from being executed.

7. Deception for active defense

7.1. *Deception in depth*

Although each of the deception techniques surveyed in Section 4 to Section 6 is able to disrupt one or several kill chain phases, when used alone, attackers can always find a way to circumvent it. One example is the various honeypot evasion techniques described in Section 4.2. By contrast, when multiple deception techniques that complement with each other are used together, forming an overall deception fabric covering several or even all layers in the deception stack, a more resilient cyber defense posture can be established. It is believed that such a deception in depth strategy should be leveraged by organizations to achieve comprehensive defense against the onslaught of advanced adversaries and attack techniques Lawrence Pingree (2015). In fact, there have already been commercial products implementing this strategy to create a complete illusion for the adversary Dec (2017).

A number of works have investigated the hybrid use of deception techniques. Through the combination, the deception

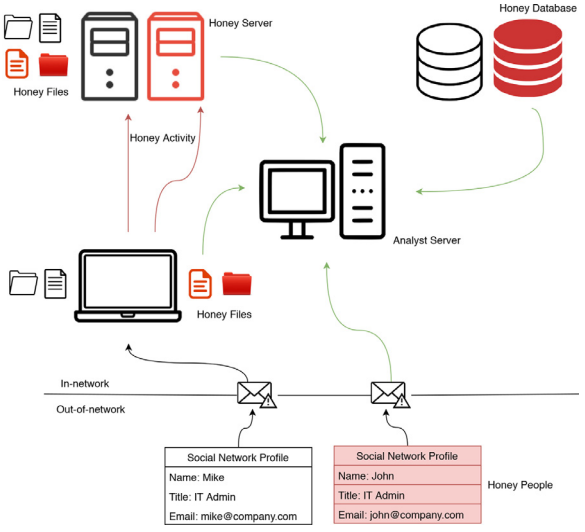

Fig. 8 – The multi-layer deception system in Wang et al. (2013).

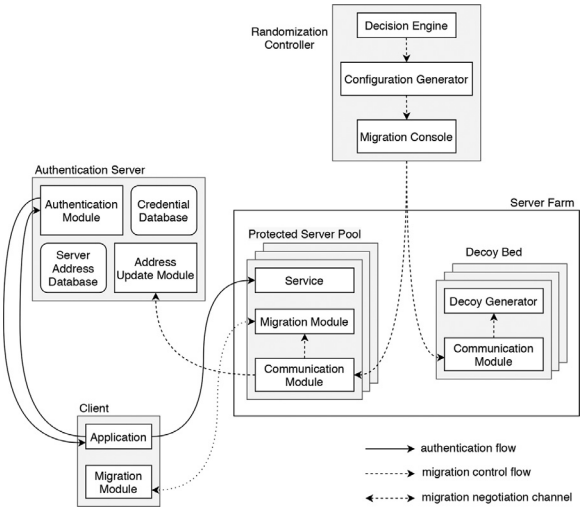

Fig. 9 – The DESIR system in Sun and Sun (2016), which is based on decoy-enhanced network address randomization.

effect on the adversary can be magnified, leading to the threat actor being deterred, delayed, distracted or detected.

Wang *et al.* Wang et al. (2013) propose a multi-layer deception system (see Fig. 8), which is composed of honeypot servers and various honeytokens such as honey people, honey files, honey database, and honey activities. The honey people is fake personas created on social network platforms. The honey activities are coupled with honey files and honeypot servers to prevent sophisticated attackers from discerning the bogus resources by observing user behaviors or network traffic. The alerts from all the deception entities are sent to the analyst server, where analysis is performed to confirm or remove the alerts. The analyst server may also correlate different alerts to extract more information of a penetration attempt. For example, if an alert is triggered on a honey file and later on a honey database entry, some correlation analyses may reveal that the two separate alerts correspond to the same espionage campaign.

A decoy-enhanced network address randomization method called DESIR (see Fig. 9) is proposed in Sun and Sun (2016), which dynamically mutates the network topology with a number of decoy servers to invalidate attacker's knowledge about the network. DESIR consists of four main components, i.e., an authentication server, a randomization controller, a protected server pool, and a decoy bed. The authentication server is responsible for verifying the client's credential, providing requested server's current IP address upon successful authentication, and updating the IP addresses of servers in the server pool. The randomization controller coordinates the mutation of the network. Its decision module determines the frequency to randomize the network addresses, configuration generator module controls the overall topology of the network, and migration console module distributes the new configurations to the real and decoy servers. Upon receiving new configurations, the decoy generator in the decoy bed will update the decoy network, including the decoy server's IP addresses and MAC addresses as well as the installed or emulated OS and applications. The decoy bed may include both high-interaction and low-interaction honeypots as decoys, which depend on the received configurations. Although the honeypots can be used to attract attackers and learn their TTPs, their main function in DESIR is to further confuse attackers, prolong their network scanning time, and invalidate the knowledge that can be gained.

In the DESIR system, as shown in Fig. 9, the authenticated client and the moving server is seamlessly connected via the migration module. However, this means that if the client is compromised, it will be easy for the adversary to trace the moving server by analyzing the network traffic. To cope with this threat, Park *et al.* Park et al. (2018) propose to inject decoy connection and traffic with the honeypot servers, as illustrated in Fig. 10. To generate decoy traffic that is even convincing for sophisticated attackers, a context-aware traffic generation mechanism is used (see Fig. 10b). On the client, the connection generator module of the decoy operation daemon is responsible for creating decoy connections with honeypot servers, and the traffic generator module creates decoy traffic of a similar pattern to the legitimate traffic. The traffic deception module on the moving server shares the characteristics of the outbound traffic, which are imitated by the traffic generator module on the decoy server to generate similar traffic with decoy processes on the client. Besides, similar to Jafarian et al. (2016), OS fingerprint mutation is also applied on all servers so that the attack surface is further obfuscated.

The SDN based CHAOS system (see Fig. 11) in Shi et al. (2017) obfuscates the network attack surface by using honeypot (i.e., decoy servers), honeytoken (i.e, fake response to port scanning), and MTD (i.e, random host mutation) techniques. In particular, host machines in the network is divided into several layers according to their security levels, which forms a CHAOS tower structure (CTS). Communication rules are defined in a CTS module. For example, connection requests from a host machine in lower layers to hosts in higher layers will be deemed as suspicious. The suspicious

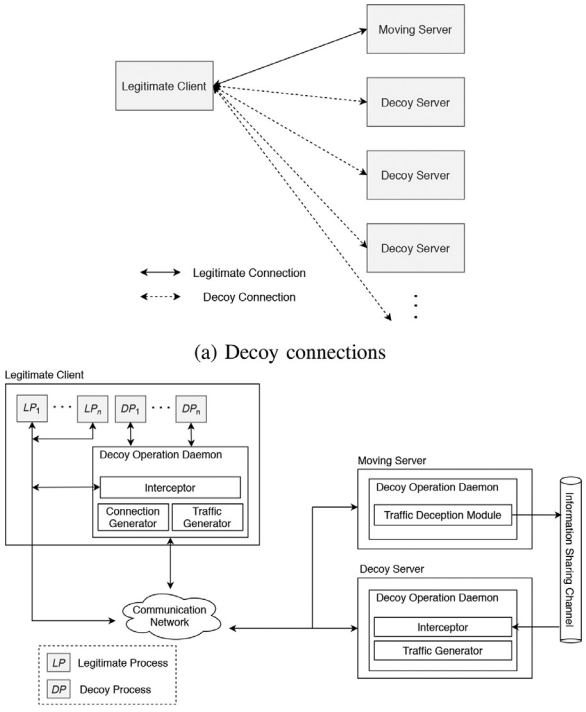

Fig. 10 – The decoy connection and traffic injection in Park et al. (2018).

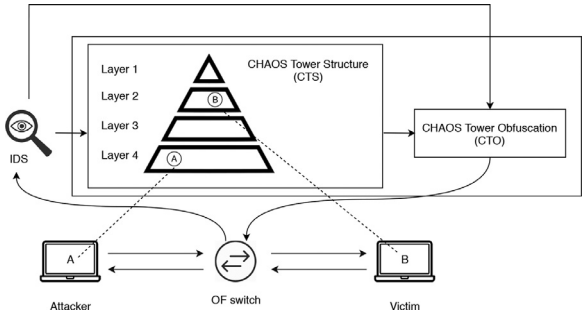

Fig. 11 – The CHAOS system in Shi et al. (2017), where suspicious connections identified by IDS or CTS are obfuscated by CTO.

communications determined by the CTS module and other traffic identified by IDS as malicious will be forwarded to a CHAOS tower obfuscation (CTO) module, where the three types of techniques listed above are implemented. The three obfuscation strategies are applied based on a threshold factor, which can be controlled by the administrator according to the required security level and the structure of the protected network.

A complete list of the reviewed deception techniques are shown in Table 3, where they are classified based on the two-dimensional taxonomy, i.e., which attack phases they can disrupt and which deception layer they belong to. Note that some methods, especially the hybrid ones, are able to disrupt multiple attack phases and use techniques from multiple layers. These methods are repeated in the table to fully indicate

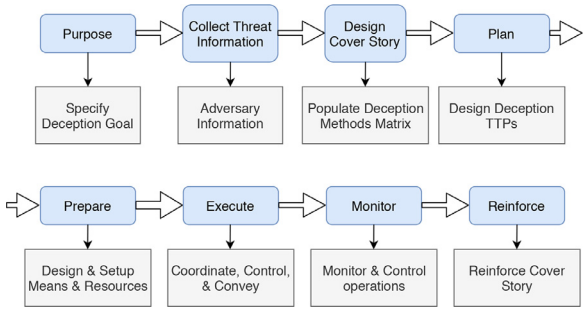

Fig. 12 – The eight-phase cyber deception chain in Heckman et al. (2015).

their characteristics and effects. The reconnaissance phase in Table 3 just refers to the external reconnaissance, while deception techniques that can guard against the internal reconnaissance and target reconnaissance attack phases are the same as those for the lateral movement phase. This separation is aimed to make it easier to understand the different applicabilities and effects of the numerous deception techniques for disrupting attacker reconnaissance.

7.2. *Deception lifecycle*

Deception techniques are employed to affect threat actors such that they take action or inaction to the advantage of cyber defenders. To achieve and maintain the desired perceptual and cognitive effects, deception mechanisms have to be properly designed and updated. Almeshekah Almeshekah (2015) proposes a deception framework comprising three main phases, i.e., planning, implementing and integrating, and monitoring and evaluating. In the planning phase, the goal of deception is specified, the attacker's bias that can be exploited to achieve desired reactions is analyzed, and the risk that may be introduced by deception techniques is also assessed. De Faveri *et al.* De Faveri et al. (2016) propose a goal-driven approach for designing the deception based defense. The approach integrates three phases, i.e., system modeling for specifying the goal, security modeling for specifying security concerns from the attacker perspective, and the deception modeling for specifying the defense (e.g., designing deception stories, monitoring channels, and deception metrics). The first two phases establish the context for modeling the deception. Heckman *et al.* Heckman et al. (2015) propose a deception chain for deception operation management from a lifecycle perspective, which is composed of eight phases as depicted in Fig. 12. Note that its second phase to the fourth phase, i.e., collecting threat information, designing cover story, and planning, can be implemented by leveraging our two-dimensional taxonomy.

After the coordinated deception tactics are built and executed, they should evolve in response to environment changes and attacker's behavior De Faveri and Moreira (2016). Take the honeypot for example. A static honeypot is very likely to be detected by the adversary. By contrast, dynamic honeypots Budiarto et al. (2004); Wang et al. (2020); Zanoramy Ansiry Zakaria and Laiha Mat Kiah (2013), besides their capability of learning about the network for automated deployment, can

Table 3 – Deception techniques classified based on the two-dimensional taxonomy, where HP, HT, and MTD denotes honeypot, honeytoken, and moving target defense, respectively.

Attack Phase	Deception Layer			
	Network	System	Software	Data
Reconnaissance	HP:Adebayo and Rawat (2020); Borders et al. (2007); Jafarian et al. (2016); Kuman et al. (2017); Leslie Shing (2016); Park et al. (2018); Provos (2004); Shi et al. (2017); Sun and Sun (2016); Tom Liston (2001)	HP:Disso et al. (2013); Jiang and Zheng (2020); Lukas Rist et al. (2015); Provos (2004); Sun et al. (2017); Vetterl (2019); Winn et al. (2015); Zhao and Qin (2017)	HP:Provos (2004)	HP:
	HT:Park et al. (2018); Shi et al. (2017); Sun et al. (2020); Trassare (2013)	HT:Albanese et al. (2015); Fred Cohen (2010); Julian (2002); Murphy et al. (2010); Park et al. (2018); Sun et al. (2020)	HT:Douglas Brewer et al. (2010); Gavrilis et al. (2007); Han et al. (2017); Katsinis and Kumar (2012, 2013); Virvilis et al. (2014)	HT:Bourke and Grzelak (2018); Karuna et al. (2020); Katsinis and Kumar (2012, 2013); Stringhini et al. (2010); Virvilis et al. (2014); Wang et al. (2013); White (2010)
	MTD: Al-Shaer et al. (2013); Antonatos et al. (2005); Duan et al. (2020); Dunlop et al. (2011, 2012); Fred Cohen (2010); Jafarian et al. (2012, 2016); Kewley et al. (2001); Park et al. (2018); Shi et al. (2017); Sun et al. (2019); Sun and Sun (2016); Sun et al. (2017); Trassare et al. (2013)	MTD: Jafarian et al. (2016); Kampanakis et al. (2014); Park et al. (2018); Zhao et al. (2017)	MTD:	MTD:
Delivery	HP:	HP:Biedermann et al. (2012); Borders et al. (2007); Disso et al. (2013); Jiang and Zheng (2020); Lukas Rist et al. (2015); Urias et al. (2016); Vetterl (2019); Winn et al. (2015); Zhao and Qin (2017)	HP:Anagnostakis et al. (2005)	HP:
	HT:	HT:	HT:	HT:
	MTD: Al-Shaer et al. (2013); Antonatos et al. (2005); Dunlop et al. (2011, 2012); Jafarian et al. (2012)	MTD:	MTD:	MTD:
Defense Evasion	HP: Alt et al. (2014); Fu et al. (2006)	HP:	HP:	HP:
	HT:	HT: Rowe et al. (2007); Rrushi (2019)	HT:	HT:
	MTD: Sengupta et al. (2018); Venkatesan et al. (2016)	MTD: Sengupta et al. (2019); Vorobeychik and Li (2014)	MTD:	MTD:
Exploitation	HP: Rrushi (2021); Virvilis et al. (2014)	HP: Biedermann et al. (2012); Borders et al. (2007); Disso et al. (2013); Jafarian et al. (2016); Jiang and Zheng (2020); Lukas Rist et al. (2015); Park et al. (2018); Sun et al. (2017); Urias et al. (2016); Vetterl (2019); Wang et al. (2013); Winn et al. (2015); Zhao and Qin (2017)	HP: Anagnostakis et al. (2005)	HP:
	HT:	HT: Araujo et al. (2014); Choi et al. (2020); Lee et al. (2020)	HT: Araujo et al. (2014); Crane et al. (2013); Virvilis et al. (2014)	HT:
	MTD:	MTD: PaX (2003); Kc et al. (2003); Li et al. (2006); Okhravi et al. (2011); Thompson et al. (2014)	MTD: Azab et al. (2011); Gupta et al. (2013); Le Goues et al. (2012)	MTD:

(continued on next page)

Table 3 (*continued*)

Installation	HP:	HP: Biedermann et al. (2012); Borders et al. (2007); Disso et al. (2013); Lukas Rist et al. (2015); Urias et al. (2016); Vetterl (2019); Winn et al. (2015); Zhao and Qin (2017)	HP:	HP:
	HT:	HT:	HT:	HT:
	MTD:	MTD:	MTD:	MTD:
C2	HP:	HP: Biedermann et al. (2012); Borders et al. (2007); Disso et al. (2013); Lukas Rist et al. (2015); Urias et al. (2016); Vetterl (2019); Winn et al. (2015); Zhao and Qin (2017)	HP:	HP:
	HT:	HT:	HT:	HT:
	MTD:	MTD:	MTD:	MTD:
Privilege Escalation	HP:	HP:	HP:	HP:
	HT:	HT: Juels (2014); Kaghazgaran and Takabi (2015)	HT:	HT:
	MTD:	MTD:	MTD:	MTD:
Credential Access	HP:	HP:	HP:	HP:
	HT:	HT: Juels (2014); Kaghazgaran and Takabi (2015)	HT:	HT: Abiodun et al. (2020); Almeshekah et al. (2015); Juels and Ristenpart (2014); Juels and Rivest (2013); Karuna et al. (2020); Omolara et al. (2019); Wang et al. (2013)
	MTD:	MTD:	MTD:	MTD:
Lateral Movement	HP: Borders et al. (2007); Jafarian et al. (2016); Park et al. (2018); Provos (2004); Shi et al. (2017); Sun and Sun (2016)	HP: Biedermann et al. (2012); Borders et al. (2007); Disso et al. (2013); Lukas Rist et al. (2015); Provos (2004); Sun et al. (2017); Urias et al. (2016); Vetterl (2019); Wang et al. (2013); Winn et al. (2015); Zhao and Qin (2017)	HP: Anagnostakis et al. (2005); Provos (2004)	HP:
	HT: Bowen et al. (2010, 2012); Park et al. (2018); Rrushi (2019); Shi et al. (2017); Sun et al. (2020); Wang et al. (2013)	HT: Araujo et al. (2014); Fred Cohen (2010); Park et al. (2018); Rrushi (2019); Sun et al. (2020); Wang et al. (2013)	HT: Araujo et al. (2014)	HT: Almeshekah et al. (2015); Juels and Rivest (2013); Karuna et al. (2020); Wang et al. (2013)
	MTD: Aseeri et al. (2017); Chiang et al. (2016); Fred Cohen (2010); Germano da Silva et al. (2015); Jafarian et al. (2016); Park et al. (2018); Shi et al. (2017); Sun et al. (2019); Sun and Sun (2016); Sun et al. (2017); Yackoski et al. (2013, 2011)	MTD: Al-Shaer et al. (2013); Antonatos et al. (2005); Dunlop et al. (2011, 2012); Jafarian et al. (2012, 2016); Kampanakis et al. (2014); Park et al. (2018); Zhao et al. (2017)	MTD:	MTD:
Collection	HP:	HP: Biedermann et al. (2012); Borders et al. (2007); Disso et al. (2013); Lukas Rist et al. (2015); Urias et al. (2016); Vetterl (2019); Winn et al. (2015); Zhao and Qin (2017)	HP: Anagnostakis et al. (2005)	HP:

(*continued on next page*)

Table 3 (*continued*)

	HT:	HT: Juels (2014); Kaghazgaran and Takabi (2015); Virvilis et al. (2014); Wang et al. (2013); Yuill et al. (2004)	HT: Antanas Čenys et al. (2005); Padayachee (2014)	HT: Abiodun et al. (2020); Almeshekah et al. (2015); Antanas Čenys et al. (2005); Ben Whitham (2017); Bercovitch et al. (2011); Bowen et al. (2009); Juels and Ristenpart (2014); Juels and Rivest (2013); Karuna et al. (2020); Omolara et al. (2019); Padayachee (2014); Voris et al. (2015, 2013); Wang et al. (2013); Yuill et al. (2004)
	MTD: Aseeri et al. (2017); Germano da Silva et al. (2015); Yackoski et al. (2013, 2011)	MTD:	MTD:	MTD: Pattuk et al. (2014)
Exfiltration	HP: Virvilis et al. (2014)	HP: Biedermann et al. (2012); Borders et al. (2007); Disso et al. (2013); Lukas Rist et al. (2015); Urias et al. (2016); Vetterl (2019); Winn et al. (2015); Zhao and Qin (2017)	HP:	HP:
	HT:	HT:	HT:	HT:
	MTD:	MTD:	MTD:	MTD:
Impact	HP: Borders et al. (2007)	HP: Biedermann et al. (2012); Borders et al. (2007); Disso et al. (2013); Jafarian et al. (2016); Lukas Rist et al. (2015); Park et al. (2018); Urias et al. (2016); Vetterl (2019); Wang et al. (2013); Winn et al. (2015); Zhao and Qin (2017)	HP: Anagnostakis et al. (2005)	HP:
	HT:	HT: Araujo et al. (2014); Choi et al. (2020); Lee et al. (2020)	HT: Araujo et al. (2014); Crane et al. (2013)	HT:
	MTD:	MTD: PaX (2003); Kc et al. (2003); Li et al. (2006); Okhravi et al. (2011); Thompson et al. (2014)	MTD: Azab et al. (2011); Gupta et al. (2013); Le Goues et al. (2012)	MTD: Smutz and Stavrou (2015)

continuously monitor the network environment for changes and reconfigure themselves accordingly. Moreover, some dynamic honeypots are able to adapt based on their interactions with attackers. By taking advantage of reinforcement learning, Wagener *et al.* Wagener et al. (2011a,b) build honeypots that can learn to adopt the best behavior such as blocking or executing commands, returning erroneous messages, and insulting the adversary. The insults act as reverse Turing tests Baird et al. (2003) and aim to identify whether the opponent is human or an automated tool. Based on the same concept, Pauna and Bica Pauna and Bica (2014) build a self-adaptive honeypot that also emulates a secure shell (SSH) server, where an extra interaction strategy, i.e., delaying the command execution, is added.

Besides using reinforcement learning to improve the interaction of deception techniques with the adversary, game theory may also be utilized. Carroll and Grosu Carroll and Grosu (2009) model the interaction between the defender and the attacker as a signaling game, which is a non-cooperative two player dynamic game (i.e., the two players take turns to choose actions) of incomplete information. The incomplete information is due to the attacker's uncertainty of the target (e.g., whether the target system is a honeypot). Deceptive equilibrium strategies are then derived to achieve better defense of the network. Rahman *et al.* Rahman et al. (2013) model the interaction between OS fingerprinter and the defender as a signaling game and the equilibrium analysis results in a counter-fingerprinting mechanism called DeceiveGame. Unlike many other tools which alter all connections' outgoing packets to deceive fingerprinting and incur significant performance degradation, DeceiveGame can distinguish fingerprinters from benign clients and selectively mystify packets to confuse the fingerprinters, hence minimizing the side effects. Carter *et al.* Carter et al. (2014) model the interaction as a two-palyer Stackelberg game to discover optimal moving target strategies (instead of simple randomization) for dynamic

platforms based defense, while Lei *et al*. Lei et al. (2017) model the confrontation in MTD as a Markov game to identify the optimal hopping strategy. In general, game theory makes it possible for cyber defenders to investigate how the adversary's belief evolves and influences his actions, and provides a quantitative framework for optimizing the manipulation of this belief to the benefit of defense Horák et al. (2017).

In deception defense, it is critical to continuously monitor the feedback channels to decide whether the desired effects on attackers are achieved. Honeypots may be easily identified, evaded, and even compromised by the adversary, honeytokens may not be enticing, and the hopping frequency in MTD may not be high enough. If the feedback indicates the deception defense is lack of effectiveness, the deception strategies, tactics, and techniques must be immediately adjusted. For this part, game theory may also be helpful. For instance, by building a multi-layer game model, a feedback learning framework is developed in Zhu and Başar (2013), which enables the system to monitor its current state and update the defense strategy based on the risk it estimates on the fly.

8. Reflection and outlook

In cyber defense, deception techniques exploit attackers' psychological biases and vulnerabilities and have direct impact on their beliefs, decisions, and actions. Even just some clues that the target system's response may be fake will delay or even turn away the adversary Rowe (2004). Compared to conventional attack prevention or detection tools which can only impede the adversary's current actions, deception techniques may have long-term impact on the adversary. Nonetheless, as deception techniques typically involve active adversary engagement, they have to be carefully maintained to stay effective. Especially when addressing APT and insider threats, high-fidelity deception over a long period is necessary. This poses stringent requirements to the deception operator. Although a vast number of deception techniques in various domains have been proposed since their inception in late 1980s, very few of them achieve real-life applications. According to Lance Spitzner, deception techniques were held back not by the concept, but by the technology Anton Chuvakin (2019). For example, early honeypots require manual customization and management, which is extremely time-consuming and error-prone. Only after recent advancement in virtualization and SDN technology, which simplifies and automates the tedious process, honeypot techniques become scalable in real-life networks. To further enhance the usability of deception techniques, some recent works Inc (2020); Islam and Al-Shaer (2020) propose to provide *deception as a service* through automatically orchestrated deception deployment with minimal human involvement. These efforts will definitely facilitate wide adoption of deception techniques. On the other hand, most of the early deception techniques have the drawback of assuming static network configurations, while recent dynamic techniques leveraging game theory models usually oversimplify the adversary's strategies Ye et al. (2021). These limitations make the actual deployment less effective and easy to be evaded by the adversary. We think recent efforts in testbeds and experimentation platforms Acosta et al. (2020); Ferguson-Walter et al. (2018) is promising to solve this problem. With deception techniques tested and validated on realistic systems and in realistic settings, not only the possible design flaws can be identified much more easily, but also the effectiveness of different techniques can be compared for easier tradeoff or complementary usage. It has been shown both analytically and experimentally that a single deception technique is not enough to attain highly resilient cyber deception Duan et al. (2018). The testbed and experimentation platforms will be an ideal environment for finding the optimal composition of different deception building blocks.

The two-dimensional taxonomy, built based on our proposed cyber kill chain model and the four-layer deception stack, facilitates the systematic review of representative approaches from the domains of honeypots, honeytokens, and MTD techniques in a threat-focused manner. To create a holistic deception fabric covering the protected network and form a complete illusion for the adversary, an integrated use of these techniques is believed to be a prerequisite. Our taxonomy may serve as a guide or reference to consolidate and coordinate the different techniques. By adopting the deception in depth strategy and properly managing deception mechanisms throughout their lifecycle, a resilient deception defense will be built, which helps organizations establish the active cyber defense posture.

For future research directions, we think that there will be more works on effective integration of the deception techniques from different domains. A well-designed deception defense should fully exploit the characteristics of different deception techniques. As these characteristics often complement with each other, the overall deception effect may be magnified and the defense cost may be optimized. For instance, defenders may distribute low-cost honeytokens all over the network to monitor the security status. Based on the indicated threat level, the instances of high-interaction honeypots, the hopping frequencies of MTD techniques, and the density of decoy activities can be dynamically adjusted. Such context awareness will be further enhanced when deception defense is combined with conventional threat detection and response (TDR) solutions. On the other hand, to smooth the integration, the deception effects on the adversary and the cost of deception operations should be quantified. In fact, there have been some works in this direction. For the former, Maleki *et al*. Maleki et al. (2016) propose a Markov model based framework for analyzing MTD techniques, where security capacity is defined to measure their strength or effectiveness. For the latter, Wang *et al*. Wang et al. (2013) model the design of the multi-layer deception system as an optimization problem to minimize the total expected loss due to system deployment and asset compromise. To better address these two problems, we feel that the quantitative framework offered by game theory will play an important role.

We may also witness hardware become a more important participant in cyber defense. A lesson from the Spectre and Meltdown attacks Abu-Ghazaleh et al. (2019) is that no security is possible if the underlying hardware is vulnerable. Conversely, a more secure hardware may better obfuscate the attack surface and boost the uncertainty. For instance, the Morpheus secure architecture in Gallagher et al. (2019) implements a hardware based churning mechanism to trans-

parently randomize key program values, which are needed by attackers for crafting successful attacks, at runtime. To enhance the value of the churning mechanism, Morpheus incorporates an attack detector. Once sensing a potential attack, the detector can immediately trigger an increased churn rate to strengthen the defense and repel the attack. Besides, the ensembles of MTD techniques developed on Morpheus, such as relocating pointers and encrypting code and pointers, can use the hardware support to achieve more randomness at a lower cost.

The ultimate target of deception defense is the adversary's perception and belief. We think that there will also be more works developed based on better understanding of the human element. Ferguson-Walter Ferguson-Walter (2020) suggests that advances in behavioral science should be leveraged to better influence attacker's target selection and operations. By manipulating threat actors' cognitive biases and cognitive load, it will be made more difficult for them to achieve their objectives.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

Abiodun EO, Jantan A, Abiodun OI, Arshad H. Reinforcing the security of instant messaging systems using an enhanced honey encryption scheme: the case of whatsapp. Wireless Personal Communications 2020. doi:10.1007/s11277-020-07163-y.

Abu-Ghazaleh N, Ponomarev D, Evtyushkin D. How the spectre and meltdown hacks really worked. IEEE Spectr. 2019;56(3):42–9. doi:10.1109/MSPEC.2019.8651934.

Acosta JC, Basak A, Kiekintveld C, Leslie N, Kamhoua C. Cybersecurity Deception Experimentation System. In: 2020 IEEE Secure Development (SecDev); 2020. p. 34–40. doi:10.1109/SecDev45635.2020.00022.

Adebayo A, Rawat DB. Deceptor-in-the-Middle (DitM): Cyber Deception for Security in Wireless Network Virtualization. In: 2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC); 2020. p. 1–6. doi:10.1109/CCNC46108.2020.9045164.

Al-Shaer E, Duan Q, Jafarian JH. Random Host Mutation for Moving Target Defense, Vol 106. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 310–27. doi:101007/978-3-642-36883-7_19.

Albanese M, Battista E, Jajodia S. A deception based approach for defeating OS and service fingerprinting. In: 2015 IEEE Conference on Communications and Network Security (CNS); 2015. p. 317–25. doi:10.1109/CNS.2015.7346842.

Almeshekah MH. Using Deception to Enhance Security: A Taxonomy, Model, and Novel Uses; 2015.

Almeshekah MH, Gutierrez CN, Atallah MJ, Spafford EH. ErsatzPasswords: Ending Password Cracking and Detecting Password Leakage. In: Proceedings of the 31st Annual Computer Security Applications Conference on - ACSAC 2015. Los Angeles, CA, USA: ACM Press; 2015. p. 311–20. doi:10.1145/2818000.2818015.

Alt L, Beverly R, Dainotti A. Uncovering network tarpits with degreaser. In: Proceedings of the 30th Annual Computer Security Applications Conference on - ACSAC '14. New Orleans, Louisiana: ACM Press; 2014. p. 156–65. doi:10.1145/2664243.2664285.

Amin S, Litrico X, Sastry S, Bayen AM. Cyber security of water SCADA systems—part i: analysis and experimentation of stealthy deception attacks. IEEE Trans. Control Syst. Technol. 2013;21(5):1963–70. doi:10.1109/TCST.2012.2211873.

Anagnostakis KG, Sidiroglou S, Akritidis P, Xinidis K, Markatos E, Keromytis AD. Detecting Targeted Attacks Using Shadow Honeypots. In: USENIX Security; 2005. p. 16.

Antanas Čenys, Darius Rainys, Lukas Radvilavičius, Nikolaj Goranin. In: IEEE Computer Society's TC on Security and Privacy. Implementation of Honeytoken Module in DBMS Oracle 9ir2 Enterprise Edition for Internal Malicious Activity Detection; 2005.

Anton Chuvakin, 2019. Will Deception Fizzle ... Again?

Antonatos S, Akritidis P, Markatos EP, Anagnostakis KG. Defending against hitlist worms using network address space randomization. In: Proceedings of the 2005 ACM Workshop on Rapid Malcode. Fairfax, VA, USA: Association for Computing Machinery; 2005. p. 30–40. doi:10.1145/1103626.1103633.

Araujo F, Hamlen KW, Biedermann S, Katzenbeisser S. From Patches to Honey-Patches: Lightweight Attacker Misdirection, Deception, and Disinformation. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. Scottsdale, Arizona, USA: Association for Computing Machinery; 2014. p. 942–53. doi:10.1145/2660267.2660329.

Aseeri A, Netjinda N, Hewett R. Alleviating eavesdropping attacks in software-defined networking data plane. In: Proceedings of the 12th Annual Conference on Cyber and Information Security Research. Oak Ridge, Tennessee, USA: Association for Computing Machinery; 2017. p. 1–8. doi:10.1145/3064814.3064832.

Augusto Paes de Barros, 2003. IDS: RES: Protocol Anomaly Detection IDS - Honeypots. https://seclists.org/focus-ids/2003/Feb/95.

Azab M, Hassan R, Eltoweissy M. ChameleonSoft: A moving target defense system. In: 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom); 2011. p. 241–50. doi:10.4108/icst.collaboratecom.2011.247115.

Baird HS, Coates AL, Fateman RJ. Pessimalprint: a reverse turing test. Int. J. Doc. Anal. Recogn. 2003;5(2):158–63. doi:10.1007/s10032-002-0089-1.

Banks JS, Sobel J. Equilibrium selection in signaling games. Econometrica 1987;55(3):647. doi:10.2307/1913604.

B.C. Ward, S.R. Gomez, R.W. Skowyra, D. Bigelow, J.N. Martin, J.W. Landry, H. Okhravi. In: Technical Report. Survey of Cyber Moving Targets: 2nd Edition. Lincoln Lab, MIT; 2018.

Beimel A. Secret-Sharing Schemes: A Survey. In: Coding and Cryptology. Berlin, Heidelberg: Springer; 2011. p. 11–46. doi:10.1007/978-3-642-20901-7_2.

Ben Whitham. Automating the Generation of Enticing Text Content for High-Interaction Honeyfiles. IEEE computer society; 2017.

Bercovitch M, Renford M, Hasson L, Shabtai A, Rokach L, Yuval Elovici. HoneyGen: An automated honeytokens generator. In: Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics; 2011. p. 131–6. doi:10.1109/ISI.2011.5984063.

Biedermann S, Mink M, Katzenbeisser S. Fast dynamic extracted honeypots in cloud computing. In: Proceedings of the 2012 ACM Workshop on Cloud Computing Security Workshop. Raleigh, North Carolina, USA: Association for Computing Machinery; 2012. p. 13–18. doi:10.1145/2381913.2381916.

Blake D. Bryant, Hossein Saiedian. A novel kill-chain framework

for remote security log analysis with SIEM software. Computers & Security 2017;67:198–210. doi:10.1016/j.cose.2017.03.003.
Borders K, Falk L, Prakash A. OpenFire: Using deception to reduce network attacks. In: 2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007; 2007. p. 224–33. doi:10.1109/SECCOM.2007.4550337.
Bourke D, Grzelak D. In: Black Hat Asia. Breach Detection at Scale with AWS Honey Tokens; 2018.
Bowen BM, Hershkop S, Keromytis AD, Stolfo SJ. Baiting Inside Attackers Using Decoy Documents. International Conference on Security and Privacy in Communication Systems, 2009.
Bowen BM, Kemerlis VP, Prabhu P, Keromytis AD, Stolfo SJ. Automating the injection of believable decoys to detect snooping. In: Proceedings of the Third ACM Conference on Wireless Network Security - WiSec '10. Hoboken, New Jersey, USA: ACM Press; 2010. p. 81. doi:10.1145/1741866.1741880.
Bowen BM, Kemerlis VP, Prabhu P, Keromytis AD, Stolfo SJ. A system for generating and injecting indistinguishable network decoys. J. Comput. Secur. 2012;20(2–3):199–221. doi:10.3233/JCS-2011-0439.
Budiarto R, Samsudin A, Heong C, Noori S. Honeypots: Why we need a dynamics honeypots?. In: Proceedings of International Conference on Information and Communication Technologies: From Theory to Applications; 2004. p. 565–6. doi:10.1109/ICTTA.2004.1307887.
Cai G-l, Wang B-s, Hu W, Wang T-z. Moving target defense: state of the art and characteristics. Frontiers of Information Technology & Electronic Engineering 2016;17(11):1122–53. doi:10.1631/FITEE.1601321.
Carroll TE, Grosu D. A Game Theoretic Investigation of Deception in Network Security. In: Proceedings of 18th International Conference on Computer Communications and Networks; 2009. p. 6.
Carter KM, Riordan JF, Okhravi H. A Game Theoretic Approach to Strategy Determination for Dynamic Platform Defenses. In: Proceedings of the First ACM Workshop on Moving Target Defense. Scottsdale, Arizona, USA: Association for Computing Machinery; 2014. p. 21–30. doi:10.1145/2663474.2663478.
Chang C-H, Zheng Y, Zhang L. A retrospective and a look forward: fifteen years of physical unclonable function advancement. IEEE Circuits Syst. Mag. 2017;17(3):32–62. doi:10.1109/MCAS.2017.2713305.
Chiang CJ, Gottlieb YM, Shridatt James Sugrim, Chadha R, Serban C, Poylisher A, Marvel LM, Santos J. ACyDS: An adaptive cyber deception system. In: MILCOM 2016 - 2016 IEEE Military Communications Conference; 2016. p. 800–5. doi:10.1109/MILCOM.2016.7795427.
Choi J, Lee H, Park Y, Kim HK, Lee J, Kim Y, Lee G, Shim S-W, Kim T. Phantomfs-v2: dare you to avoid this trap. IEEE Access 2020;8:198285–300. doi:10.1109/ACCESS.2020.3034443.
Christian Seifert, Ian Welch, Peter Komisarczuk. In: Technical Report. Taxonomy of Honeypots; 2006.
Cimpanu, C., 2020a. Czech hospital hit by cyberattack while in the midst of a COVID-19 outbreak. https://www.zdnet.com/article/czech-hospital-hit-by-cyber-attack-while-in-the-midst-of-a-covid-19-outbreak/.
Cimpanu, C., 2020b. There's now COVID-19 malware that will wipe your PC and rewrite your MBR. https://www.zdnet.com/article/theres-now-covid-19-malware-that-will-wipe-your-pc-and-rewrite-your-mbr/.
Crane S, Larsen P, Brunthaler S, Franz M. Booby trapping software. In: Proceedings of the 2013 Workshop on New Security Paradigms Workshop - NSPW '13. Banff, Alberta, Canada: ACM Press; 2013. p. 95–106. doi:10.1145/2535813.2535824.
Cybenko G, Giani A, Thompson P. Cognitive hacking: a battle for the mind. Computer (Long Beach Calif.) 2002;35(8):50–6. doi:10.1109/MC.2002.1023788.
2019 Data Breach Investigations Report. https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf.
2003. PaX Address Space Layout Randomization. https://pax.grsecurity.net/docs/aslr.txt.
2017. Deception in Depth – The Case for a Full-Stack Architecture. https://trapx.com/deception-in-depth-the-case-for-a-full-stack-architecture/.
De Faveri C, Moreira A. Designing Adaptive Deception Strategies. In: 2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C); 2016. p. 77–84. doi:10.1109/QRS-C.2016.15.
De Faveri C, Moreira A, Amaral V. Goal-Driven Deception Tactics Design. In: 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE). Ottawa, ON, Canada: IEEE; 2016. p. 264–75. doi:10.1109/ISSRE.2016.44.
Disso JP, Jones K, Bailey S. A Plausible Solution to SCADA Security Honeypot Systems. In: 2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications; 2013. p. 443–8. doi:10.1109/BWCCA.2013.77.
Douglas Brewer, Kang Li, Laksmish Ramaswamy, Calton Pu. A Link Obfuscation Service to Detect Webbots; 2010.
Duan Q, Al-Shaer E, Islam M, Jafarian H. CONCEAL: A Strategy Composition for Resilient Cyber Deception-Framework, Metrics and Deployment. In: 2018 IEEE Conference on Communications and Network Security (CNS); 2018. p. 1–9. doi:10.1109/CNS.2018.8433196.
Duan Q, Al-Shaer E, Xie J. Range and Topology Mutation Based Wireless Agility. In: Proceedings of the 7th ACM Workshop on Moving Target Defense. New York, NY, USA: Association for Computing Machinery; 2020. p. 59–67. doi:10.1145/3411496.3421228.
Dunlop M, Groat S, Urbanski W, Marchany R, Tront J. MT6D: A Moving Target IPv6 Defense. In: 2011 - MILCOM 2011 Military Communications Conference; 2011. p. 1321–6. doi:10.1109/MILCOM.2011.6127486.
Dunlop M, Groat S, Urbanski W, Marchany R, Tront J. The blind man's bluff approach to security using IPv6. IEEE Security & Privacy 2012;10(4):35–43. doi:10.1109/MSP.2012.28.
Efendi AM, Ibrahim Z, Zawawi MA, Abdul Rahim F, Pahri NM, Ismail A. A Survey on Deception Techniques for Securing Web Application. In: 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS); 2019. p. 328–31. doi:10.1109/BigDataSecurity-HPSC-IDS.2019.00066.
Eugene H. Spafford, 2011. More than passive defense. https://www.cerias.purdue.edu/.
Ferguson-Walter K, Fugate S, Mauger J, Major M. Game theory for adaptive defensive cyber deception. In: Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security - HotSoS '19. Nashville, Tennessee: ACM Press; 2019. p. 1–8. doi:10.1145/3314058.3314063.
Ferguson-Walter K, Shade T, Rogers A, Trumbo MCS, Nauer KS, Divis KM, Jones A, Combs A, Abbott RG. In: Technical Report. The Tularosa Study: An Experimental Design and Implementation to Quantify the Effectiveness of Cyber Deception.. Sandia National Lab. (SNL-NM), Albuquerque, NM (United States); 2018. doi:10.24251/HICSS.2019.874.
Ferguson-Walter KJ. An Empirical Assessment of the Effectiveness of Deception for Cyber Defense. University of Massachusetts Amherst; 2020.

Fraunholz, D., Anton, S.D., Lipps, C., Reti, D., Krohmer, D., Pohl, F., Tammen, M., Schotten, H.D., 2018. Demystifying Deception Technology:A Survey. arXiv:1804.06196 [cs], 1804.06196.
Fred Cohen, 1998. Deception ToolKit. http://www.all.net/dtk/.
Fred Cohen, 2010. Moving target defenses with and without cover deception. http://all.net/Analyst/2010-10.pdf.
Friedkin NE. Theoretical foundations for centrality measures. American Journal of Sociology 1991;96(6):1478–504.
Fu X, Yu W, Cheng D, Tan X, Streff K, Graham S. On Recognizing Virtual Honeypots and Countermeasures. In: 2006 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing; 2006. p. 211–18. doi:10.1109/DASC.2006.36.
Gallagher M, Biernacki L, Chen S, Aweke ZB, Yitbarek SF, Aga MT, Harris A, Xu Z, Kasikci B, Bertacco V, Malik S, Tiwari M, Austin T. Morpheus: A Vulnerability-Tolerant Secure Architecture Based on Ensembles of Moving Target Defenses with Churn. In: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. Providence RI USA: ACM; 2019. p. 469–84. doi:10.1145/3297858.3304037.
Gavrilis D, Chatzis I, Dermatas E. Flash Crowd Detection Using Decoy Hyperlinks. In: 2007 IEEE International Conference on Networking, Sensing and Control; 2007. p. 466–70. doi:10.1109/ICNSC.2007.372823.
Germano da Silva E, Dias Knob LA, Wickboldt JA, Gaspary LP, Granville LZ, Schaeffer-Filho A. Capitalizing on SDN-based SCADA systems: An anti-eavesdropping case-study. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM); 2015. p. 165–73. doi:10.1109/INM.2015.7140289.
Giora Engel, 2014. Deconstructing The Cyber Kill Chain. https://www.darkreading.com/attacks-breaches/deconstructing-the-cyber-kill-chain/a/d-id/1317542.
Gupta A, Kerr S, Kirkpatrick MS, Bertino E. Marlin: A Fine Grained Randomization Approach to Defend against ROP Attacks. In: Network and System Security. Berlin, Heidelberg: Springer; 2013. p. 293–306. doi:10.1007/978-3-642-38631-2_22.
Han X, Kheir N, Balzarotti D. Evaluation of Deception-Based Web Attacks Detection. In: Proceedings of the 2017 Workshop on Moving Target Defense - MTD '17. Dallas, Texas, USA: ACM Press; 2017. p. 65–73. doi:10.1145/3140549.3140555.
Han X, Kheir N, Balzarotti D. Deception techniques in computer security: a research perspective. ACM Comput Surv 2018;51(4). doi:10.1145/3214305.
Hayatle O, Youssef A, Otrok H. Dempster-shafer evidence combining for (anti)-honeypot technologies. Information Security Journal: A Global Perspective 2012;21(6):306–16. doi:10.1080/19393555.2012.738375.
Heckman KE, Stech FJ, Schmoker BS, Thomas RK. Denial and deception in cyber defense. IEEE Computer 2015;48(4):36–44. doi:10.1109/MC.2015.104.
Horák K, Zhu Q, Bošanský B. Manipulating Adversary's Belief: A Dynamic Game Approach to Deception by Design for Proactive Network Security, Vol. 10575; 2017. p. 273–94. doi:10.1007/978-3-319-68711-7_15.
Hou N, Wang Z, Ho DWC, Dong H. Robust partial-nodes-based state estimation for complex networks under deception attacks. IEEE Trans. Cybern. 2020;50(6):2793–802. doi:10.1109/TCYB.2019.2918760.
Huang C, Han J, Zhang X, Liu J. Automatic identification of honeypot server using machine learning techniques. Security and Communication Networks 2019;2019:1–8. doi:10.1155/2019/2627608.
Huang C-H, Lee T-H, Chang L-h, Lin J-R, Horng G. Adversarial Attacks on SDN-Based Deep Learning IDS System. In: Kim KJ, Kim H, editors. In: Mobile and Wireless Technology 2018. Singapore: Springer; 2018. p. 181–91. doi:10.1007/978-981-13-1059-1_17.
Hutchins EM, Cloppert MJ, Amin RM. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues in Information Warfare & Security Research 2011;1(1).
Inc, T.S., 2020. TrapX Introduces Industry-First Deception-As-A-Service Solution, TrapX Flex™. https://www.prnewswire.com/news-releases/trapx-introduces-industry-first-deception-as-a-service-solution-trapx-flex-301162363.html.
Islam MM, Al-Shaer E. Active Deception Framework: An Extensible Development Environment for Adaptive Cyber Deception. In: 2020 IEEE Secure Development (SecDev); 2020. p. 41–8. doi:10.1109/SecDev45635.2020.00023.
Jafarian JH, Al-Shaer E, Duan Q. Openflow random host mutation: Transparent moving target defense using software defined networking. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks - HotSDN '12. Helsinki, Finland: ACM Press; 2012. p. 127. doi:10.1145/2342441.2342467.
Jafarian JH, Niakanlahiji A, Al-Shaer E, Duan Q. Multi-dimensional Host Identity Anonymization for Defeating Skilled Attackers. In: Proceedings of the 2016 ACM Workshop on Moving Target Defense - MTD'16. Vienna, Austria: ACM Press; 2016. p. 47–58. doi:10.1145/2995272.2995278.
Jiang K, Zheng H. Design and Implementation of A Machine Learning Enhanced Web Honeypot System. In: 2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI); 2020. p. 957–61. doi:10.1109/CISP-BMEI51763.2020.9263640.
Joseph Corey. In: Technical Report. Advanced Honey Pot Identification and Exploitation. Phrack Inc.; 2003.
Juels A. A bodyguard of lies: The use of honey objects in information security. In: Proceedings of the 19th ACM Symposium on Access Control Models and Technologies - SACMAT '14. London, Ontario, Canada: ACM Press; 2014. p. 1–4. doi:10.1145/2613087.2613088.
Juels A, Ristenpart T. In: Technical Report. Honey Encryption: Security Beyond the Brute-Force Bound; 2014.
Juels A, Rivest RL. Honeywords: Making password-cracking detectable. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security - CCS '13. Berlin, Germany: ACM Press; 2013. p. 145–60. doi:10.1145/2508859.2516671.
Julian DP. Delaying-Type Responses for Use by Software Decoys; 2002.
Kaghazgaran P, Takabi H. Toward an insider threat detection framework using honey permissions. Journal of Internet Services and Information Securuirty (JISIS) 2015. doi:10.22667/JISIS.2015.08.31.019.
Kampanakis P, Perros HG, Beyene T. SDN-Based solutions for moving target defense network protection. Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014. doi:10.1109/WoWMoM.2014.6918979.
Karuna P, Purohit H, Jajodia S, Ganesan R, Uzuner O. Fake document generation for cyber deception by manipulating text comprehensibility. IEEE Syst. J. 2020:1–11. doi:10.1109/JSYST.2020.2980177.
Katsinis C, Kumar B. A Security Mechanism for Web Servers Based on Deception. In: Proceedings on the International Conference on Internet Computing (ICOMP); 2012. p. 6.
Katsinis C, Kumar B. A Framework for Intrusion Deception on Web Servers. In: International Conference on Internet Computing(ICOMP); 2013. p. 7.
Kc GS, Keromytis AD, Prevelakis V. Countering code-injection attacks with instruction-set randomization. In: Proceedings of the 10th ACM Conference on Computer and Communications Security. Washington D.C., USA: Association for Computing Machinery; 2003. p. 272–80. doi:10.1145/948109.948146.

Kelly Sheridan, 2020. Defense Evasion Dominated 2019 Attack Tactics. https://www.darkreading.com/vulnerabilities—threats/defense-evasion-dominated-2019-attack-tactics/d/d-id/1337457.
Kewley D, Fink R, Lowry J, Dean M. Dynamic approaches to thwart adversary intelligence gathering, Vol. 1; 2001. p. 176–85. doi:10.1109/DISCEX.2001.932214.
Kitchenham B. Procedures for performing systematic reviews. Keele, UK, Keele University 2004:33.
Korzhyk D, Conitzer V, Parr R. Complexity of computing optimal stackelberg strategies in security resource allocation games. Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10) 2010:6.
Krawetz N. Anti-honeypot technology. IEEE Security Privacy 2004;2(1):76–9. doi:10.1109/MSECP.2004.1264861.
Kreutz D, Ramos FMV, Veríssimo PE, Rothenberg CE, Azodolmolky S, Uhlig S. Software-defined networking: a comprehensive survey. Proc. IEEE 2015;103(1):14–76. doi:10.1109/JPROC.2014.2371999.
Kuman S, Groš S, Mikuc M. An experiment in using IMUNES and Conpot to emulate honeypot control networks. In: 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO); 2017. p. 1262–8. doi:10.23919/MIPRO.2017.7973617.
Lance Spitzner, 2001. The Value of Honeypots, Part One: Definitions and Values of Honeypots. Security Focus information.
Lance Spitzner, 2003. Honeytokens: The Other Honeypot. Security Focus information.
Lance Spitzner, 2004. Problems and Challenges with Honeypots. Security Focus information.
Larsen P, Homescu A, Brunthaler S, Franz M. SoK: Automated Software Diversity. In: 2014 IEEE Symposium on Security and Privacy; 2014. p. 276–91. doi:10.1109/SP.2014.25.
Lawrence Pingree. In: Technical Report. Emerging Technology Analysis: Deception Techniques and Technologies Create Security Technology Business Opportunities. Gartner, Inc.; 2015.
Le Goues C, Nguyen T, Forrest S, Weimer W. Genprog: a generic method for automatic software repair. IEEE Trans. Software Eng. 2012;38(1):54–72. doi:10.1109/TSE.2011.104.
Lee J, Choi J, Lee G, Shim S-W, Kim T. Phantomfs: file-based deception technology for thwarting malicious users. IEEE Access 2020;8:32203–14. doi:10.1109/ACCESS.2020.2973700.
Lei C, Ma D-H, Zhang H-Q. Optimal strategy selection for moving target defense based on markov game. IEEE Access 2017;5:156–69. doi:10.1109/ACCESS.2016.2633983.
Lei C, Zhang H-Q, Tan J-L, Zhang Y-C, Liu X-H. Moving target defense techniques: a survey. Security and Communication Networks 2018;2018. doi:10.1155/2018/3759626.
Leslie Shing. An Improved Tarpit for Network Deception. Naval Postgraduate School; 2016.
Li L, Just J, Sekar R. Address-Space Randomization for Windows Systems. In: 2006 22nd Annual Computer Security Applications Conference (ACSAC'06). Miami Beach, FL, USA: IEEE; 2006. p. 329–38. doi:10.1109/ACSAC.2006.10.
Lin, Z., Shi, Y., Xue, Z., 2019. IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection. arXiv:1809.02077 [cs], 1809.02077.
Lukas Rist, Johnny Vestergaard, Daniel Haslinger, Andrea Pasquale, John Smith, 2015. Conpot. http://conpot.org/.
Lynn III WJ. Defending a new domain: the pentagon's cyberstrategy. Foreign Affairs 2010;89(5):97–108.
Maleki H, Valizadeh S, Koch W, Bestavros A, van Dijk M. Markov Modeling of Moving Target Defense Games. In: Proceedings of the 2016 ACM Workshop on Moving Target Defense - MTD'16. Vienna, Austria: ACM Press; 2016. p. 81–92. doi:10.1145/2995272.2995273.
Malone ST. In: Black Hat. USA. Using an Expanded Cyber Kill Chain Model to increase attack resiliency; 2016.
Marc Laliberte, 2016. A Twist On The Cyber Kill Chain: Defending Against A JavaScript Malware Attack. https://www.darkreading.com/attacks-breaches/a-twist-on-the-cyber-kill-chain-defending-against-a-javascript-malware-attack/a/d-id/1326952.
McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. Openflow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review 2008;38(2):69–74. doi:10.1145/1355734.1355746.
Meira-Goes R, Lafortune S, Marchand H. Synthesis of supervisors robust against sensor deception attacks. IEEE Trans Automat Contr 2021. doi:10.1109/TAC.2021.3051459.
Mitnick KD, Simon WL, Wozniak S. The art of deception: controlling the human element of security. 1 edition. Wiley; 2007.
Murphy SB, McDonald JT, Mills RF. An Application of Deception in Cyberspace: Operating System Obfuscation. International Conference on Information Warfare and Security, 2010.
Nawrocki, M., Wählisch, M., Schmidt, T.C., Keil, C., Schönfelder, J., 2016. A Survey on Honeypot Software and Data Analysis. arXiv:1608.06249 [cs], 1608.06249.
Nazario J. In: USENIX Workshop on Large-Scale Exploits and Emergent Threats. PhoneyC: A Virtual Client Honeypot; 2009.
Neil C. Rowe. Cyber War and Cyber Terrorism. In: A Colarik, L Janczewski, editors. Deception in Defense of Computer Systems from Cyber-attack; 2007.
2010. NITRD CSIA IWG Cybersecurity Game-Change Research & Development Recommendations.
Okhravi H, Comella A, Robinson E, Yannalfo S, Michaleas P, Haines J. Creating a Cyber Moving Target for Critical Infrastructure Applications. In: International Conference on Critical Infrastructure Protection. Hanover, NH, USA; 2011. p. 107–23. doi:10.1007/978-3-642-24864-1_8.
Okhravi H, Hobson T, Bigelow D, Streilein W. Finding focus in the blur of moving-target techniques. IEEE Security & Privacy 2014;12(2):16–26. doi:10.1109/MSP.2013.137.
Okhravi H, Rabe MA, Mayberry TJ, Leonard WG, Hobson TR, Bigelow D, Streilein WW. In: Technical Report. Survey of Cyber Moving Target Techniques. Lincoln Lab, MIT; 2013. doi:10.21236/ADA591804.
Omolara AE, Jantan A, Isaac Abiodun O, Victoria Dada K, Arshad H, Emmanuel E. A deception model robust to eavesdropping over communication for social network systems. IEEE Access 2019;7:100881–98. doi:10.1109/ACCESS.2019.2928359.
Padayachee K. Aspectising honeytokens to contain the insider threat. IET Inf. Secur. 2014;9(4):240–7. doi:10.1049/iet-ifs.2014.0063.
Palmer, D., 2020. Coronavirus-themed phishing attacks and hacking campaigns are on the rise. https://www.zdnet.com/article/coronavirus-themed-phishing-attacks-and-hacking-campaigns-are-on-the-rise/.
Park K, Woo S, Moon D, Choi H. Secure cyber deception architecture and decoy injection to mitigate the insider threat. Symmetry (Basel) 2018;10(1):14. doi:10.3390/sym10010014.
Pattuk E, Kantarcioglu M, Lin Z, Ulusoy H. Preventing cryptographic key leakage in cloud virtual machines. In: Proceedings of the 23rd USENIX Conference on Security Symposium. San Diego, CA: USENIX Association; 2014. p. 703–18.
Pauna A, Bica I. RASSH - Reinforced adaptive SSH honeypot. In: Proceedings of the 10th International Conference on Communications (COMM); 2014. p. 1–6. doi:10.1109/ICComm.2014.6866707.

Pawlick, J., Colbert, E., Zhu, Q., 2019. A Game-Theoretic Taxonomy and Survey of Defensive Deception for Cybersecurity and Privacy. arXiv:1712.05441 [cs], 1712.05441.
. In: Technical Report. PCI Hardware Security Module Security Requirements, Version 1.0. PCI Security Standards Council; 2009.
Pols P. The Unified Kill Chain. Cyber Security Academy; 2017.
Provos N. A virtual honeypot framework. In: Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13. San Diego, CA: USENIX Association; 2004. p. 1.
Rahman MA, Manshaei MH, Al-Shaer E. A game-theoretic approach for deceiving Remote Operating System Fingerprinting. In: Proceedings of IEEE Conference on Communications and Network Security (CNS); 2013. p. 73–81. doi:10.1109/CNS.2013.6682694.
Reidy P. In: Black Hat USA. Combating the Insider Threat at the FBI: Real World Lessons Learned; 2013.
Ross R, Pillitteri V, Graubart R, Bodeau D, McQuaid R. In: Technical Report. Developing Cyber Resilient Systems: A Systems Security Engineering Approach. Gaithersburg, MD: National Institute of Standards and Technology; 2019. doi:10.6028/NIST.SP.800-160v2.
Rowe NC. A model of deception during cyber-attacks on information systems. In: IEEE First Symposium onMulti-Agent Security and Survivability, 2004; 2004. p. 21–30. doi:10.1109/MASSUR.2004.1368414.
Rowe NC, Custy EJ, Duong BT. Defending cyberspace with fake honeypots. J. Comput. (Taipei) 2007;2(2):25–36. doi:10.4304/jcp.2.2.25-36.
Rrushi JL. Honeypot Evader: Activity-guided Propagation versus Counter-evasion via Decoy OS Activity. In: Proceedings of the 14th IEEE International Conference on Malicious and Unwanted Software. Nantucket, Massachusetts, USA; 2019. p. 11.
Rrushi JL. DNIC Architectural developments for 0-knowledge detection of OPC malware. IEEE Trans. Dependable Secure Comput. 2021;18(1):30–44. doi:10.1109/TDSC.2018.2872536.
Scott Smith. In: Technical Report. Catching Flies: A Guide to the Various Flavors of Honeypots. SANS; 2016.
Scottberg B, Yurcik W, Doss D. Internet honeypots: Protection or entrapment?. In: IEEE International Symposium on Technology and Society (ISTAS'02). Raleigh, NC, USA: IEEE; 2002. p. 387–91. doi:10.1109/ISTAS.2002.1013842.
Seifert C, Welch I, Komisarczuk P. HoneyC - The Low-Interaction Client Honeypot. Proceedings of the 2007 NZCSRCS, 2007.
Sengupta S, Chakraborti T, Kambhampati S. MTDeep: Boosting the Security of Deep Neural Nets Against Adversarial Attacks with Moving Target Defense. In: Decision and Game Theory for Security. Cham: Springer International Publishing; 2019. p. 479–91. doi:10.1007/978-3-030-32430-8_28.
Sengupta S, Chowdhary A, Huang D, Kambhampati S. Moving Target Defense for the Placement of Intrusion Detection Systems in the Cloud. In: Bushnell L, Poovendran R, Başar T, editors. In: Decision and Game Theory for Security. Cham: Springer International Publishing; 2018. p. 326–45. doi:10.1007/978-3-030-01554-1_19.
Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., Kambhampati, S., 2020. A Survey of Moving Target Defenses for Network Security. arXiv:1905.00964 [cs], 1905.00964.
Sentz K, Ferson S, Sentz K. In: Technical Report. Combination of Evidence in Dempster-Shafer Theory. Albuquerque, New Mexico.: Sandia National Laboratories; 2002.
Shade T, Rogers A, Ferguson-Walter K, Elsen SB, Fayette D, Heckman K. The Moonraker Study: An Experimental Evaluation of Host-Based Deception. Hawaii International Conference on System Sciences, 2020.
Shi Y, Zhang H, Wang J, Xiao F, Huang J, Zha D, Hu H, Yan F, Zhao B. CHAOS: An SDN-based moving target defense system. Security and Communication Networks 2017.
Sinclair C, Pierce L, Matzner S. An application of machine learning to network intrusion detection. In: Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99); 1999. p. 371–7. doi:10.1109/CSAC.1999.816048.
Smutz C, Stavrou A. In: RAID. Preventing Exploits in Microsoft Office Documents Through Content Randomization; 2015. doi:10.1007/978-3-319-26362-5_11.
Spitzner L. Honeypots: tracking hackers. Boston: Addison-Wesley Professional; 2002.
Stoll C. The Cuckoo's egg: Tracking a spy through the maze of computer espionage. USA: Doubleday; 1989.
Stringhini G, Kruegel C, Vigna G. Detecting Spammers on Social Networks. Proceedings of the 26th Annual Computer Security Applications Conference, 2010.
Sun J, Liu S, Sun K. A Scalable High Fidelity Decoy Framework against Sophisticated Cyber Attacks. In: Proceedings of the 6th ACM Workshop on Moving Target Defense. New York, NY, USA: Association for Computing Machinery; 2019. p. 37–46. doi:10.1145/3338468.3356826.
Sun J, Sun K. DESIR: Decoy-enhanced seamless IP randomization. In: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications; 2016. p. 1–9. doi:10.1109/INFOCOM.2016.7524602.
Sun J, Sun K, Li Q. CyberMoat: Camouflaging critical server infrastructures with large scale decoy farms. In: 2017 IEEE Conference on Communications and Network Security (CNS); 2017. p. 1–9. doi:10.1109/CNS.2017.8228642.
Sun J, Sun K, Li Q. Towards a Believable Decoy System: Replaying Network Activities from Real System. In: 2020 IEEE Conference on Communications and Network Security (CNS); 2020. p. 1–9. doi:10.1109/CNS48642.2020.9162163.
Thompson M, Evans N, Kisekka V. Multiple OS rotational environment an implemented Moving Target Defense. In: 2014 7th International Symposium on Resilient Control Systems (ISRCS); 2014. p. 1–6. doi:10.1109/ISRCS.2014.6900086.
Tom Liston, 2001. LaBrea: "Sticky" Honeypot and IDS. http://labrea.sourceforge.net/labrea-info.html.
Trassare ST. A Technique for Presenting a Deceptive Dynamic Network Topology. Naval Postgraduate School; 2013.
Trassare ST, Beverly R, Alderson D. A Technique for Network Topology Deception. In: MILCOM 2013 - 2013 IEEE Military Communications Conference; 2013. p. 1795–800. doi:10.1109/MILCOM.2013.303.
Tzu S. The art of war. first thus edition. Filiquarian; 2007.
Urias VE, Stout WMS, Lin HW. Gathering threat intelligence through computer network deception. In: 2016 IEEE Symposium on Technologies for Homeland Security (HST). Waltham, MA, USA: IEEE; 2016. p. 1–6. doi:10.1109/THS.2016.7568916.
U.S. Department of Homeland Security, 2016. Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies.
Usama M, Asim M, Latif S, Qadir J, Ala-Al-Fuqaha. Generative Adversarial Networks For Launching and Thwarting Adversarial Attacks on Network Intrusion Detection Systems. In: 2019 15th International Wireless Communications Mobile Computing Conference (IWCMC); 2019. p. 78–83. doi:10.1109/IWCMC.2019.8766353.
Venkatesan S, Albanese M, Cybenko G, Jajodia S. A Moving Target Defense Approach to Disrupting Stealthy Botnets. In: Proceedings of the 2016 ACM Workshop on Moving Target Defense. Vienna, Austria: Association for Computing Machinery; 2016. p. 37–46. doi:10.1145/2995272.2995280.

Vetterl A, Clayton R. Bitter harvest: Systematically fingerprinting low- and medium-interaction honeypots at internet scale. Proceedings of the 12th USENIX Conference on Offensive Technologies. Baltimore, MD, USA, 2018.
Vetterl AM. Honeypots in the Age of Universal Attacks and the Internet of Things. University of Cambridge; 2019.
Virvilis N, Vanautgaerden B, Serrano OS. Changing the game: The art of deceiving sophisticated attackers. In: 2014 6th International Conference On Cyber Conflict (CyCon 2014). Tallinn, Estonia: IEEE; 2014. p. 87–97. doi:10.1109/CYCON.2014.6916397.
Voris J, Jermyn J, Boggs N, Stolfo S. Fox in the trap: Thwarting masqueraders via automated decoy document deployment. In: Proceedings of the Eighth European Workshop on System Security - EuroSec '15. Bordeaux, France: ACM Press; 2015. p. 1–7. doi:10.1145/2751323.2751326.
Voris J, Jermyn J, Keromytis AD, Stolfo SJ. Bait and Snitch: Defending Computer Systems with Decoys. In: Proceedings of the Cyber Infrastructure Protection Conference, Strategic Studies Institute; 2013. p. 25.
Vorobeychik Y, Li B. Optimal randomized classification in adversarial settings. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems. Paris, France: International Foundation for Autonomous Agents and Multiagent Systems; 2014. p. 485–92.
Wagener G, State R, Dulaunoy A, Engel T. Heliza: talking dirty to the attackers. Journal in Computer Virology 2011;7(3):221–32. doi:10.1007/s11416-010-0150-4.
Wagener G, State R, Engel T, Dulaunoy A. Adaptive and self-configurable honeypots. In: 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops; 2011. p. 345–52. doi:10.1109/INM.2011.5990710.
Wang P, Wu L, Cunningham R, Zou CC. Honeypot detection in advanced botnet attacks. Int. J. Inf. Comput. Secur. 2010;4(1):30. doi:10.1504/IJICS.2010.031858.
Wang S, Pei Q, Wang J, Tang G, Zhang Y, Liu X. An intelligent deployment policy for deception resources based on reinforcement learning. IEEE Access 2020;8:35792–804. doi:10.1109/ACCESS.2020.2974786.
Wang W, Jeffrey Bickford, Ilona Murynets, Ramesh Subbaraman, Gokul Singaraju. Detecting targeted attacks by multilayer deception. Journal of Cyber Security and Mobility 2013;2(2):175–99. doi:10.13052/jcsm2245-1439.224.
Wang W, Sheng Y, Wang J, Zeng X, Ye X, Huang Y, Zhu M. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access 2018;6:1792–806. doi:10.1109/ACCESS.2017.2780250.
White J. Creating Personally Identifiable Honeytokens. In: Sobh T, editor. In: Innovations and Advances in Computer Sciences and Engineering. Dordrecht: Springer Netherlands; 2010. p. 227–32. doi:10.1007/978-90-481-3658-2_39.
Winn M, Rice M, Dunlap S, Lopez J, Mullins B. Constructing cost-effective and targetable industrial control system honeypots for production networks. Int. J. Crit. Infrastruct. Prot. 2015;10:47–58. doi:10.1016/j.ijcip.2015.04.002.
Yackoski J, Bullen H, Yu X, Li J. Applying Self-Shielding Dynamics to the Network Architecture. In: Moving Target Defense II. New York, NY: Springer; 2013. p. 97–115. doi:10.1007/978-1-4614-5416-8_6.
Yackoski J, Xie P, Bullen H, Li J, Sun K. A Self-shielding Dynamic Network Architecture. In: Military Communications Conference (MILCOM 2011); 2011. p. 1381–6. doi:10.1109/MILCOM.2011.6127498.
Ye D, Zhu T, Shen S, Zhou W. A differentially private game theoretic approach for deceiving cyber adversaries. IEEE Trans. Inf. Forensics Secur. 2021;16:569–84. doi:10.1109/TIFS.2020.3016842.
Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 2017;5:21954–61. doi:10.1109/ACCESS.2017.2762418.
Yuill J, Zappe M, Denning D, Feer F. Honeyfiles: deceptive files for intrusion detection. Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop 2004. doi:10.1109/IAW.2004.1437806.
Zanoramy Ansiry Zakaria W, Laiha Mat Kiah M. A review of dynamic and intelligent honeypots. ScienceAsia 2013;39S. doi:10.2306/scienceasia1513-1874.2013.39S.001.
Zhang Q, Liu K, Xia Y, Ma A. Optimal stealthy deception attack against cyber-physical systems. IEEE Trans. Cybern. 2020;50(9):3963–72. doi:10.1109/TCYB.2019.2912622.
Zhang Y, Juels A, Reiter MK, Ristenpart T. Cross-VM side channels and their use to extract private keys. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. Raleigh, North Carolina, USA: Association for Computing Machinery; 2012. p. 305–16. doi:10.1145/2382196.2382230.
Zhao C, Qin S. A research for high interactive honepot based on industrial service. In: 2017 3rd IEEE International Conference on Computer and Communications (ICCC). Chengdu: IEEE; 2017. p. 2935–9. doi:10.1109/CompComm.2017.8323069.
Zhao Z, Liu F, Gong D. An SDN-based fingerprint hopping method to prevent fingerprinting attacks. Security and Communication Networks 2017.
Zhu Q, Başar T. Game-Theoretic Approach to Feedback-Driven Multi-stage Moving Target Defense. In: Decision and Game Theory for Security; 2013. p. 246–63. doi:10.1007/978-3-319-02786-9_15.

Li Zhang received the B.Eng. (Hons.) and Ph.D. degrees from Nanyang Technological University (NTU), Singapore, in 2010 and 2015, respectively. He is currently a principal engineer in Cybersecurity Strategic Technology Center, ST Engineering, Singapore. His research interests include vulnerability detection, malware analysis, deception based cyber defense, as well as hardware security and trust.

Vrizlynn L. L. Thing is the Senior VP, Head of Cybersecurity Strategic Technology Centre at ST Engineering, Singapore. She also holds the appointment of Honorary Assistant Superintendent of Police. Prior to joining ST Engineering, she was the Head of Cybersecurity & Intelligence at A*STAR. She has over 20 years of cybersecurity research, development and large-scale cyber programme management & implementation experience. Dr Thing received her Ph.D. degree from Imperial College London, U.K. During her Ph.D. studies, she won the "Best Student Paper Award" at the 20th IFIP International Information Security Conference and the Imperial College London "Hilfred Chau Postgraduate Award".