Computers & Security 140 (2024) 103792

Contents lists available at ScienceDirect

Computers & Security

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

A comprehensive survey on cyber deception techniques to improve honeypot performance

Amir Javadpour [a,*,1], Forough Ja'fari [b], Tarik Taleb [c], Mohammad Shojafar [d], Chafika Benzaïd [e]

[a] ICTFICIAL Oy, Espoo, Finland
[b] Department of Computer Engineering, Yazd University, Safaeih, Yazd, 100190, Iran
[c] Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, Bochum, Germany
[d] 5G/6GIC, Institute for Communication Systems (ICS), University of Surrey, Guildford, GU27XH, United Kingdom
[e] Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland

ARTICLE INFO

Keywords:
Cyber deception
Honeynet efficiency
Honeypot performance
Cybersecurity
Professional adversaries

ABSTRACT

Honeypot technologies are becoming increasingly popular in cybersecurity as they offer valuable insights into adversary behavior with a low rate of false detections. By diverting the attention of potential attackers and siphoning off their resources, honeypots are a powerful tool for protecting critical assets within a network. However, the cybersecurity landscape constantly evolves, and professional attackers are always working to uncover and bypass honeypots. Once an adversary successfully identifies a deception mechanism in place, they may change their tactics, potentially causing significant harm to the network. Maintaining a high level of deception is crucial for honeypots to remain undetectable. This paper explores various deception techniques designed specifically for honeypots to enhance their performance while making them impervious to detection. Previous research has not provided a detailed comparison of these techniques, particularly those tailored to honeynets. Therefore, we categorize the presented techniques into relevant classes, subject them to a comparative analysis, and evaluate their effectiveness in simulation scenarios. We also present a mathematical model that comprehensively represents and compares various honeynet research endeavors. In addition, we provide insightful suggestions that highlight the existing research gaps in this field and offer a roadmap for future expansion. This includes extending deception techniques to emulate vulnerabilities inherent in 5G and software-defined networks, which address the evolving challenges of the cybersecurity landscape. The findings and insights presented in this paper are valuable to honeypot developers and cybersecurity researchers alike, providing a vital resource for advancing the field and fortifying network defenses against ever-evolving threats.

Contents

* Corresponding author.
E-mail addresses: a.javadpour87@gmail.com (A. Javadpour), azadeh.mth@gmail.com (F. Ja'fari), tarik.taleb@rub.de (T. Taleb), m.shojafar@surrey.ac.uk (M. Shojafar), chafika.benzaid@oulu.fi (C. Benzaïd).
[1] Previous address when this research work was initiated: Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland.

https://doi.org/10.1016/j.cose.2024.103792
Received 14 June 2023; Received in revised form 7 February 2024; Accepted 26 February 2024
Available online 1 March 2024

1. Introduction

Cyber threats include any events that can potentially harm an information system through unauthorized access. The number of new cyber threats targeting critical assets in industrial, governmental, and personal networks grows yearly. Moreover, these threats release different variants with improved malicious features, making them more complicated and hard to detect. For example, the existence of the Mirai botnet was first discovered in 2016. Mirai is an army of bots under the control of an adversary, and they can launch a Distributed Denial of Service (DDoS) against the devices in an Internet of Things (IoT) network. The adversaries did not stop with the current version of Mirai. They designed different variants of it, like Persirai (Kolias et al., 2017; Wang, 2022), to better perform their malicious activities without being detected. The security reports demonstrate that the number of botnets doubled after introducing Mirai (Javadpour et al., 2022a,b).

According to the seriousness of the effects of cyber threats on computer networks, network defenders attempt to design tools by which they can detect and analyze unknown threats and prevent dangerous ones. Even though different security mechanisms are designed for detecting and preventing the attacks, such as threat monitoring systems, firewalls, IPSec, and Intrusion Detection Systems (IDSs), they are not efficient enough to both (1) detect zero-day and unknown threats and (2) closely analyze the adversary's behavior. However, a honeypot is a deceptive tool that is capable of helping the network defender reach both of the two mentioned goals (Valero et al., 2020; Javadpour et al., 2023a; Sangaiah et al., 2023a,b; Javadpour et al., 2017; Hedayati and Mostafavi, 2021).

Honeypots overwhelm the adversaries and waste their resources, creating ambiguity for the adversaries and hampering them from achieving their notorious goals. They use deception techniques to be one step ahead of their adversaries. The suffix "Honeypots" defines various deception techniques to attract the adversary's attention. Honeypots are traps that attract the adversary with their attractive information and services and monitor their activities by deceiving him/her. From the adversary's point of view, honeypots have valuable information and provide real services. However, a honeypot's information and services are fake, aiming to extract the adversary's behavior pattern. The advantages of using deception strategies offered by the honeypots for a network are as follows. First, the adversary's certainty about the value of its stolen data is reduced. Since the adversary becomes more active when confusing, we can capture more information about its behavior. Moreover, the adversary wastes its time and other resources, which are kept from critical network parts. Besides, the adversary's sense of danger about being deceived prevents him/her from launching cyber attacks (Toor and Bhandari, 2017; Javadpour et al., 2023b).

The honeypots have *three* main functionalities, namely *Detection*, *Prevention*, and *Research*. For the detection feature, the superior advantage of honeypots over other security tools in detecting cyber attacks is their low rate of false detection. Since the legal users do not interact with the honeypots, their false detection is almost zero. This superiority helps the honeypots detect zero-day attacks better than the other tools. Regarding the prevention functionality, three aspects of the honeypots

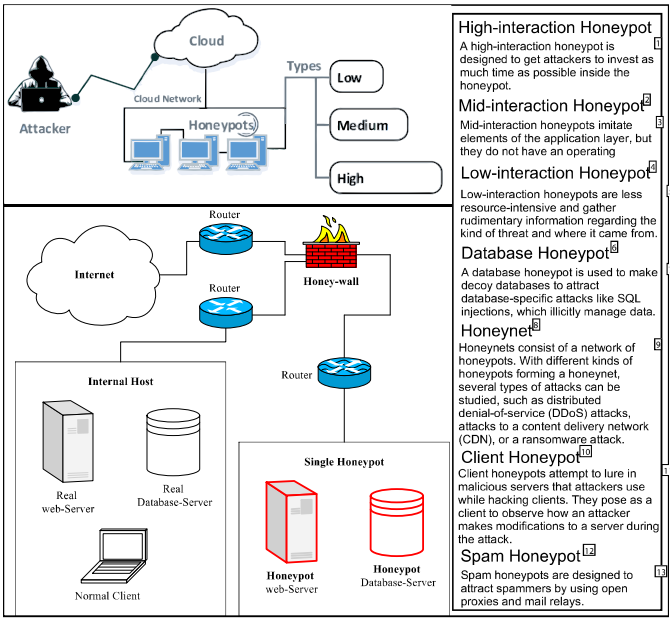


Fig. 1. A sample of a honeynet architecture.

are considered: (1) slowing down the adversary, (2) creating a sense of danger for the adversary even if there are no security mechanisms deployed on the network, and (3) wasting the adversary's resources. Honeypots play a crucial role in cybersecurity research by collecting comprehensive data on adversaries' activities and reactions. This wealth of information is essential for researchers to scrutinize and analyze patterns in adversarial behavior. These patterns offer valuable insights into adversaries' ever-evolving tactics and strategies, leading to a more informed and strategic approach to strengthening cybersecurity defenses. This deeper understanding not only informs the development of more powerful security tools but also aids in proactively identifying and mitigating emerging threats, contributing to a more resilient and proactive cybersecurity landscape (Almeshekah and Spafford, 2016).

There are different types of honeypot deployment methods in a network, such as a Minefield (Doubleday et al., 2016), Shield (Fan et al., 2015), and Honeyfarm (Fan et al., 2017a). Generally speaking, a deceptive network with one or more honeypots, despite the different deployment types, is called a honeynet (Han et al., 2016). A sample architecture for a honeynet is shown in Fig. 1. The Internet, the internal normal hosts, and the single honeypots can be connected to each other by the routers. The honeypots can provide fake services, such as web and database-related services. Traffic to the honeypots can first pass through a Honeywall, such as Roo (Ganesarathinam et al., 2020), which can be supported by an IDS, such as Snort (Koziol, 2003). IPtables can also restrict the communication between the honeypots and the real systems Gautam et al. (2015).

The paper cited as Ferguson-Walter et al. (2021) aims to tackle the rising global threat of cyber attacks and the pressing need for advanced cybersecurity measures. The study analyzes data from an experiment involving 130 professional red teamers who participated in a controlled network penetration test. The objective is to assess how defensive deception, encompassing both cyber and psychological aspects, influences attackers. By comparing attacker progress across various experimental conditions, the research investigates the effectiveness of decoy systems in cyber defense. The findings suggest that the most substantial impact on cyber attack behavior occurs when a combination of decoys and the explicit acknowledgment of deception is employed, in contrast to a control condition without deception. This paper presents the research work conducted in the past 15 years on honeypots and their deception techniques. Several surveys exist in this field; among them are the recent researches performed by Fraunholz et al. (2018), Razali et al. (2018), Zobal et al. (2019), Seungjin et al. (2020), and Lackner (2021). However, these researchers did not mention the honeynets' deception techniques. In addition, a comparative analysis of the deception methods is notably absent in the existing literature. This paper addresses these gaps by conducting a comprehensive review of honeypot research and their associated deception techniques. The investigation includes comparative analyses and simulation results to provide valuable insights. It is important to clarify that this paper focuses on honeypot-related challenges and their deceptive techniques, and does not delve into the description or examination of anti-honeypot techniques, which pertain to the methods used by adversaries to detect the presence of honeypots. Such considerations fall outside the scope of this paper.

Introduction

In this survey, we particularly intend to answer the following *five* questions:

- **Question 1:** What is the significance of understanding various honeypot types, and what metrics are used by developers, whether they are honeypot developers or security tool developers, in selecting the most suitable type? This question explores the importance of gaining insights into honeypot features and provides essential recommendations for their effective utilization. Whether honeypot developers are creating new honeypots or security tool developers are integrating honeypots into their solutions, this knowledge empowers them to make informed decisions that align with specific security needs and use cases. Honeypots come in various types: low-interaction, high-interaction, and hybrid honeypots. Understanding the nuances of each type is essential for making informed decisions. Metrics like detection rate, resource consumption, and ease of deployment help developers weigh the pros and cons of different honeypot types.
- **Question 2:** What deception techniques enhance honeypot performance, and which metrics can evaluate their effectiveness? The answer to this question enumerates deception techniques that can be employed to improve individual honeypots. This information empowers researchers and developers to apply these techniques individually or in combination to enhance their honeypots and potentially inspire the creation of new techniques.

Deception techniques in honeypots include emulating vulnerable services, altering response times, and honeytokens. Metrics like the interaction rate, attacker engagement, and false-positive rates help assess these techniques' effectiveness. Understanding which technique aligns best with specific goals is crucial for honeypot success.

- **Question 3:** How can we mathematically model a honeynet comprising multiple cooperating honeypots deployed in a network with varying parameters? Developers may face confusion about the parameters of a honeynet, and this model aids in comprehensively considering all parameters to manage their network of honeypots accurately.
  Modeling a honeynet involves capturing the relationships between honeypots, network topology, and attacker behavior. Parameters may include honeypot placement, data sharing, and communication protocols. A mathematical model offers a structured approach to honeynet design and management, reducing ambiguity.
- **Question 4:** What deception techniques are used to enhance honeynets' performance, and which ongoing research can yield better results? Similar to single honeypots, the answer to this question enumerates deception techniques that can be employed to improve honeynets. Researchers and developers can incorporate these techniques into their networks and compare them to select the most appropriate ones.
  Deception techniques in honeynets may involve coordinated responses, dynamic topology changes, and distributed data analysis. Research in this field continually evolves, with promising approaches like AI-driven deception and machine learning-based anomaly detection. Evaluating the latest research findings can lead to more effective honeynets.
- **Question 5:** How can current techniques be improved, and what research gaps exist? Answering this question helps define future research directions in the field of honeypots.
  Current honeypot techniques may have limitations, such as high false positives or evasion by sophisticated attackers. Improvements could include refining deception strategies, enhancing evasion detection, and developing more robust data analysis methods. Research gaps could encompass areas like IoT honeypots, deception at scale, and real-time threat intelligence integration.

- Comprehensive Classification of Honeypots: This paper meticulously categorizes and presents an in-depth analysis of various honeypot classifications, including low-interaction, high-interaction, and hybrid honeypots, among others. By comparing the strengths and weaknesses of each type, it equips developers and network administrators with a holistic view, allowing them to make well-informed decisions when choosing the most efficient honeypot for their specific network environment. This classification is a valuable reference point for practitioners aiming to enhance their cybersecurity infrastructure.
- Deception Techniques for Enhanced Single Honeypots: The paper delves into the realm of deception techniques tailored for single honeypots. It categorizes these techniques and provides practical sample scenarios for each. This approach goes beyond theoretical discussions, offering concrete and actionable insights. Developers and researchers can draw inspiration from these scenarios to implement deception strategies effectively. This contribution bridges the gap between theory and practical application in the field of honeypots.
- Enhancement of Honeynets through Deception Techniques: In addressing the optimization of honeynets, the paper categorizes various deception techniques and goes the extra mile by comparing them through simulation scenarios. Illustrating how these techniques perform in real-world scenarios provides network security professionals with valuable guidance on deploying honeynets effectively. This practical approach empowers practitioners to harness the collective power of multiple honeypots to detect and deter attackers more efficiently.
- Innovative Mathematical Model for Honeynets: The paper's proposal of a novel mathematical model for honeynets is a pioneering contribution. This model covers a broad spectrum of honeynet configurations, including previously unexplored modes. Offering a structured framework aids network administrators in accurately modeling and managing complex honeynet architectures. This innovative model allows for more precise honeynet design and deployment, ultimately strengthening network defense mechanisms.
- Practical Guidance and Research Directions: Beyond categorizing honeypots and honeynet techniques, the paper provides practical and detailed suggestions. It addresses research gaps and outlines future directions in the evolving landscape of honeypots and honeynets. It fosters continuous advancements in the field by offering a roadmap for future research endeavors. Additionally, it provides actionable insights for researchers and practitioners, enabling them to navigate evolving threats and challenges in the realm of cybersecurity effectively.

The structure of this paper is as follows. In section 2, we provide an overview of the various classifications for honeypot systems. Subsequently, in section 3 and section 5, we delve into the deception techniques employed in both single honeypots and honeynets, also offering a comparative analysis of these approaches. Moving forward, section 7 draws attention to the unresolved issues within the realm of honeypots and subsequently identifies potential avenues for future research. Finally, in section 8, we summarize the key findings and conclusions drawn from this survey.

2. Honeypot classification

As dynamic cybersecurity tools, Honeypots manifest in many types, each distinct in its own right and serving unique purposes. These distinctions often arise from varying criteria, whether it be their intended purpose, the methodology of their implementation, or the specific threat landscape they are designed to confront. Consequently, selecting the most appropriate honeypot type is a pivotal decision that must be made after careful consideration of several critical factors. The first factor to weigh is the current state of the network itself. Assessing the network's topology, scale, and the critical assets it houses is paramount. Different honeypot types are more suitable for specific network configurations than others. For instance, a low-interaction honeypot might be a pragmatic choice for a small, resource-constrained network. In contrast, a high-interaction honeypot could be deployed in a more complex environment with ample resources. Another vital consideration is the availability of resources in terms of hardware and personnel. High-interaction honeypots, which fully emulate systems and engage with potential attackers, demand more resources than their low-interaction counterparts. Resource constraints can often steer the choice towards one type over the other. Equally significant is the ever-evolving threat landscape within the network. The choice of honeypot should align with the predominant attack vectors and tactics observed in the network. Tailoring honeypot deployments to mirror the tactics of potential adversaries can yield invaluable insights and enhance network security. In this section, we aim to explore the diverse classifications of honeypots comprehensively. By delving into these classifications and offering insights into the strengths and weaknesses of each honeypot type, we aim to empower developers, network administrators, and cybersecurity practitioners with the knowledge needed to make informed decisions. The subsequent sections of this paper will further build upon this foundation, equipping readers with the tools to harness honey-

pots effectively as a strategic asset in safeguarding network infrastructures.

2.1. Classification based on purpose

Honeypots can be classified based on their primary purpose, leading to several meaningful distinctions:

1. Research Honeypots: These honeypots are primarily designed for the purpose of data collection and analysis. They are invaluable tools for gathering information about attackers' techniques, tactics, and motivations. Researchers and threat analysts often deploy honeypots to gain insights into emerging threats and vulnerabilities.

2. Production Honeypots: In contrast to research honeypots, production honeypots are integrated into live, operational networks. Their primary function is to actively divert and engage attackers away from critical systems, effectively acting as decoys that protect legitimate targets. Production honeypots are commonly employed in operational environments to enhance overall security.

3. High-Interaction Honeypots: High-interaction honeypots offer a realistic environment that closely emulates actual systems and services. They facilitate extensive interaction with potential attackers, making them invaluable for capturing in-depth information about attack techniques and strategies. However, their complexity necessitates careful management.

4. Low-Interaction Honeypots: Low-interaction honeypots, on the other hand, simulate services with limited functionality, reducing the risk of exposing vulnerabilities. While they may not provide as much data as high-interaction honeypots, they are notably easier to deploy and maintain, making them suitable for various scenarios.

2.2. Functionality metrics for honeypot selection

Selecting the most appropriate honeypot type necessitates using specific functionality metrics that align with the network's unique requirements. We present *six* key functionality metrics to assist honeypot developers and administrators in their selection process, each addressing critical aspects of honeypot deployment:

- **Implementation Cost (ImCo):**
  1. This metric quantifies the honeypots' cost, focusing on physical implementation expenses. Developers with limited physical resources must pay particular attention to this metric, as it directly impacts the feasibility of deployment.
- **Design Complexity (DeCo):**
  2. DeCo specifies the complexity of designing the algorithms and operations required for the honeypot. Honeypots with higher DeCo demand significant effort and time during their design phase, influencing the overall project timeline and resource allocation.
- **Compromising Risk (CoRi):**
  3. CoRi assesses the level of risk posed when an attacker compromises a honeypot. Networks with critical resources must consider this metric carefully to mitigate risks and safeguard their valuable assets.
- **Collected Data (CoDa):**
  4. The CoDa quantifies the volume of data collected by the honeypot during its operation. Developers seeking comprehensive insights into attack patterns should opt for honeypots with a high data collection capacity, directly impacting the quality and richness of information gathered.
- **Deception Power (DePo):**
  5. The DePo indicates the effectiveness of a honeypot's deception mechanisms. While some honeypots may be relatively easy to implement, they may also be more susceptible to early detection by adversaries. Evaluating DePo is crucial for determining the honeypot's ability to deceive and divert attackers effectively.
- **Handled Connections (HaCo):**
  6. The HaCo measures the number of connections a honeypot must actively manage. This metric assumes significance in scenarios where network bandwidth and other resource-related constraints come into play. Careful consideration of HaCo ensures that the honeypot operates optimally within the network's limitations.

  These functionality metrics provide a structured framework for evaluating and selecting the most appropriate honeypot type based on specific network conditions, objectives, and resource constraints. In the ensuing sections, we delve into each of these metrics in greater depth, offering practical guidance and insights to aid in deploying and optimizing honeypots.

Fig. 2 compares the honeypot types in a class according to the mentioned functionality metrics. Hereunder, different types of classification, defined for honeypots, are characterized. Each class's features are also stated. We also present some practical researches on honeypots along with their type and class in Table 1, Fig. 3. The mentioned classes in each research are marked with a star symbol in this table.

2.3. Honeypots purposes

Honeypots can be classified based on their purpose and application. Accordingly, honeypots are classified into two types:

- **Research Honeypots (ReH):** These honeypots are designed to gather nearly complete information about the launched attacks and a list of network vulnerabilities. The researchers can analyze and use this information to design mitigation methods. Research honeypots do not have a direct productive value for the organization using them. However, they can bring indirect value to the organization's future security. This type of honeypot is more utilized in governmental organizations, big research companies, and universities. Since research honeypots record all the adversary's behaviors, their implementation and maintenance are so complex. For example, Ferretti et al. (2019) proposed a research honeypot to obtain more information about industrial cyber threats.
- **Production Honeypots (PrH):** This type of honeypot is designed for protecting the networks and reducing the threatening dangers. These honeypots simulate the favored vulnerabilities and services to protect the adversary from the main servers and protect them. For example, Guerra Manzanares (2017) designed a production honeypot that can secure IoT devices. Production honeypots are easier to implement and maintain because they do not need to collect complete information about the adversaries' behavior.

It is worth mentioning that research and production honeypots are not completely separated. An organization may use a honeypot as a research tool, but another one may use it to protect its network from attacks.

2.4. Honeypots interactions

Another method for classifying the honeypots is based on the level of their interaction with the adversary. Three types of honeypots exist in this classification:

- **High-interaction Honeypots (HiH):** These honeypots emulate all parts of a system and all of its services. Hence, the adversary can hardly distinguish these honeypots from a real system. However, if the adversary successfully compromises these honeypots, it can abuse them to launch powerful attacks against the network. As a result, designing and implementing high-interaction honeypots require more attention. For example, You et al. (2020) proposed a high interaction honeypot for industrial logic controllers.

(a) By purposes (b) By interactions (c) By implementations

(d) By activities (e) By running sides (f) By consistency

(g) By uniformity

Fig. 2. Comparing the honeypot types in each class regarding Implementation Cost (ImCo), Design Complexity (DeCo), Compromising Risk (CoRi), Collected Data (CoDa), Deception Power (DePo), and Handled Connections (HaCo).

- **Low-interaction Honeypots (LoH):** This type of honeypot just emulates a specific part of an operating system and a certain number of services. Therefore, its design and implementation are much simpler and bring less damage when compromised. However, its recognition is simpler than the honeypots with a higher level of interaction and is not able to analyze all aspects of the adversary's behavior. As an example, Fan et al. (2019) proposed a low-interaction honeypot that is powerful in capturing the adversary's data.

  High-interaction honeypots provide a realistic emulation of entire systems and services, attracting and engaging advanced adversaries and providing comprehensive insights into their tactics. However, designing and implementing HiH can be complex and resource-intensive, and there is a risk of attackers abusing compromised honeypots. Low-interaction honeypots offer a simpler and more lightweight alternative, suitable for resource-constrained environments. While LoH may not provide as detailed insights as HiH, they can still gather valuable threat intelligence data and serve as early warning systems. Both HiH and LoH have their advantages and challenges, and organizations should carefully consider their use-cases and deployment strategies based on their specific security needs and resource constraints. Significant use-case scenario

Table 1
The types of honeypots adopted in some practical researches. Research Honeypots (ReH), Medium-interaction Honeypots (MeH), Virtual Honeypots (ViH), Physical Honeypots (PhH), Server-side Honeypots (SeH), Static Honeypots (StH) and Homogeneous Honeypots (HoH).

Ref.	Purpose	Interaction	Implementation	Activity	Running Side	Consistency	Uniformity	Short Description
Zhuge et al. (2007)	ReH	MeH	ViH *	PaH	SeH	StH	HoH	A honeypot to learn about the botnets activities
Nazario (2009)	ReH *	LoH *	ViH *	AcH	ClH *	DyH *	HoH	A honeypot to detect malicious web pages
Jiang et al. (2010)	ReH	HiH *	ViH	AcH	ClH *	StH	HeH *	A honeypot to track the malware and capture their information
Alosefer and Rana (2010)	ReH	LoH *	PhH	AcH *	ClH *	StH	HoH	A web-based honeypot to learn about malicious contents
Kumar et al. (2012)	ReH	HiH *	ViH *	AcH *	SeH/ClH	StH	HoH	A honeypot to collect a wide range of attack vectors
Ayeni et al. (2013)	PrH	MeH *	ViH *	PaH	SeH *	DyH *	HoH	A honeypot to detect and limit denial of service attacks
Zarras (2014)	PrH	MeH	ViH *	AcH	ClH *	DyH	HoH	A web browser honeypot to protect users from being infected
Hirata et al. (2015)	PrH	HiH *	ViH *	AcH	SeH *	StH	HoH	A web-based honeypot with live migration
Rahmatullah et al. (2016)	PrH *	LoH *	PhH	PaH	SeH *	StH	HoH	A web-based honeypot for embedded systems
Pa et al. (2016)	PrH	MeH	PhH*	PaH	SeH *	StH	HoH	A honeypot to attract telnet-based attacks
Perevozchikov et al. (2017)	ReH	HiH	PhH	PaH *	SeH *	StH	HoH	An FTP honeypot server for malware detection
Fraunholz et al. (2017)	ReH	MeH *	PhH	PaH	SeH	StH	HoH	A honeypot to learn about the attack sessions
Guerra Manzanares (2017)	PrH *	LoH *	ViH *	PaH	SeH	StH	HoH	A honeypot to secure IoT devices
Fan et al. (2017b)	PrH *	MeH	ViH *	PaH	SeH	DyH *	HeH *	A honeynet that can manage versatile deceptive tools
Luo et al. (2017)	PrH	MeH	ViH	PaH *	SeH	DyH	HoH	A honeypot that intelligently changes its interaction
Wang et al. (2018)	ReH	MeH *	ViH	AcH	ClH	StH	HoH	A honeypot for IoT networks
Ferretti et al. (2019)	ReH *	LoH *	PhH	PaH	SeH	StH	HeH	A honeynet to learn about industrial threats
Fan et al. (2019)	ReH	LoH *	ViH *	AcH	SeH	StH	HeH *	A honeypot which is powerful in capturing the attackers data
Park et al. (2019)	PrH	MeH *	ViH *	PaH	SeH	DyH *	HoH	A honeypot to be the destination of malicious traffic redirection
Naik et al. (2020)	PrH	LoH *	PhH	PaH	SeH	DyH *	HoH	A honeypot which is immune against fingerprinting attacks
You et al. (2020)	PrH	HiH *	PhH *	PaH	SeH	StH	HoH	A flexible and scalable honeypot for industrial logic controllers
Khan and Abbasi (2020)	ReH	LoH	PhH	PaH	SeH	StH	HoH *	A honeynet to help the performance of IDSs
Ja'fari et al. (2021)	PrH	MeH	ViH *	PaH	SeH/ClH	DyH	HoH	A honeynet to detect and prevent Mirai botnet propagation

for HiH is incident response training. These honeypots simulate real-world attack scenarios in a controlled environment, making them useful for training security personnel in incident response procedures. This helps organizations prepare for and mitigate cyber incidents effectively. By using HiH honeypots for training, organizations can learn from simulated attacks, making their incident response more effective and efficient. On the other hand, LoH is simpler to design, implement, and maintain compared to HiH. Their narrower focus and reduced scope of emulation make them more accessible for organizations with limited resources or expertise. LoH honeypots typically require fewer resources in terms of computational power, storage, and network bandwidth, making them more lightweight and easier to deploy at scale.

- **Medium-interaction Honeypots (MeH):** The interaction level of these honeypots is between the two previous ones. An operating system is not completely emulated, but an entire application layer service is implemented in medium-interaction honeypots. If a honeynet contains both low-interaction and high-interaction honeypots, we call it a medium-interaction honeynet. As an example of medium interaction honeypots, we can mention the honeypot proposed by Fraunholz et al. (2017), which serves Telnet and SSH to learn about attack sessions. Ja'fari et al. (2021) also proposed a honeynet that includes a front-end low-interaction and a back-end high-interaction honeypot server, which is assumed to be a medium-interaction honeynet.

2.5. Honeypots implementations

Honeypots are categorized into two types based on their implementation method:

- **Physical Honeypots (PhH):** This type of honeypot is implemented on a separate machine and has a unique IP address. Implementing this class is rather hard and time-consuming, and its security requires special supervision and attention. When the network has to support a widespread address space, using physical honeypots is not affordable. IoTPOT (Pa et al., 2016) is an example of a physical honeypot.
- **Virtual Honeypots (ViH):** Virtual honeypots did not require dedicated physical machines for implementation. Multiple virtual honeypots could be hosted on a single physical server, making them a cost-effective and efficient choice. The implementation time was often reduced, and network development was simplified compared to physical honeypots. For instance, HoneyIo4 (Guerra Manzanares, 2017) was an example of a virtual honeypot.

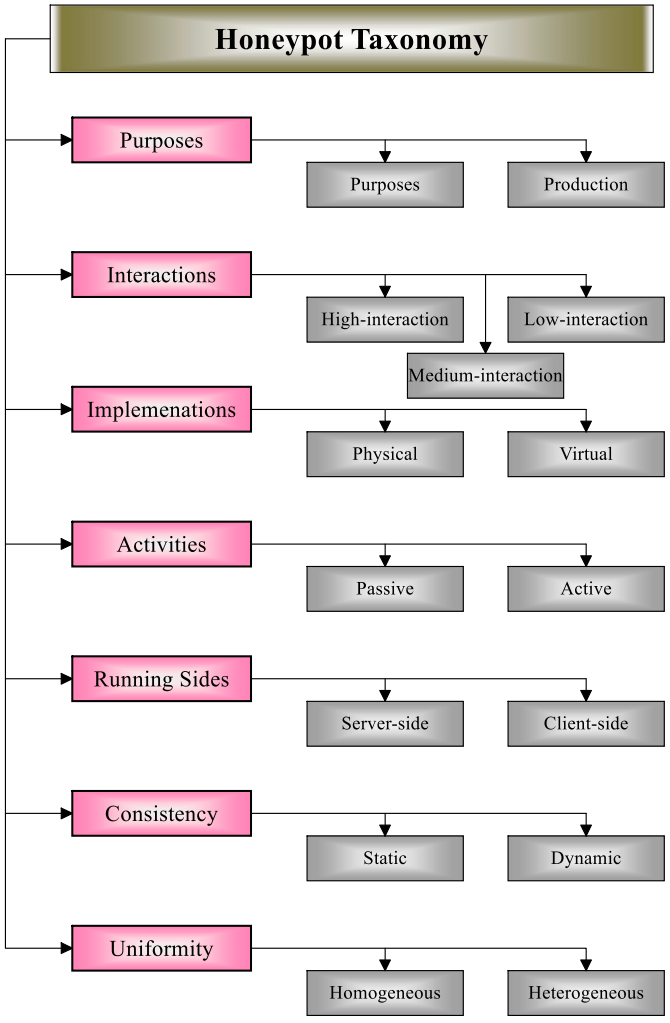


Fig. 3. Honeypot taxonomy.

2.6. Honeypots activities

The activity level of different honeypots is not the same. In this regard, honeypots are classified into two types:

- **Passive Honeypots (PaH):** The objective of these honeypots is to collect information from the adversary without guaranteeing to protect the other systems. A passive honeypot waits for the adversaries' connections to record their information and does not take much effort to attract them. For example, Perevozchikov et al. (2017) proposed a passive honeypot that serves as an FTP service to detect the malware.
- **Active Honeypots (AcH):** Active honeypots are designed to lure potential adversaries away from critical systems by proactively seeking and engaging with them. In contrast to passive honeypots, active honeypots employ various methods to attract and deceive potential attackers (McCarthy et al., 2022). An example of an active honeypot can be found in the work proposed by Kumar et al. (2012).

2.7. Honeypots running sides

According to the running side of honeypots, they are classified into two types:

- **Server-side Honeypots (SeH):** These honeypots try to identify server-side vulnerabilities and security leaks that a server may have. They attempt to protect the critical servers in a network. The adversary that launches an attack against server-side honeypots acts as a client. Hence, server-side honeypots are also useful tools to detect malicious clients in a network. IoTPOT (Pa et al., 2016) is a server honeypot that aims to attract telnet-based attacks.
- **Client-side Honeypots (ClH):** These honeypots are designed to find malicious servers and also try to identify client-side vulnerabilities. A client-side honeypot searches for suspicious servers, sends them a request, and then analyzes the response received. If the response is anomalous, the malicious servers can be detected, and the honeypot can identify the client-side vulnerabilities they exploit. A client honeypot is proposed by Zarras (2014), which acts like a web browser to find malicious web pages and protect legal users.

2.8. Honeypots operation

Honeypots are classified into two main types based on their consistency:

- **Static Honeypots (StH):** Static honeypots have a certain configuration and always act the same for different adversaries and in different times. Their behavior is fixed despite different network conditions and under different types of attacks. Hence, the adversary may suspect them and find out that they are decoys. Most of the honeypots mentioned in autoreftab:types are static honeypots. For example, ThingPot (Wang et al., 2018) is a static honeypot designed for IoT platforms.
- **Dynamic Honeypot (DyH):** These honeypots offer greater flexibility compared to static honeypots. They can dynamically adapt to changes in network status and modify their behavior in response to polymorphic attacks. For instance, Naik et al. (2020) introduced a dynamic honeypot that can adjust its configuration in various situations, particularly in response to fingerprinting attacks.

2.9. Honeypots uniformity

We can classify the honeypots according to the uniformity of their decoys:

- **Homogeneous Honeypots (HoH):** These honeypots use similar decoys in the network. They only use a single type of trap to deceive the adversary. The performance of these honeypots is restricted, and they can detect and delay only specific types of attacks. Khan and Abbasi (2020) proposed a team of homogeneous honeypots forming a honeynet that collects useful information for IDSs.
- **Heterogeneous Honeypot (HeH):** These honeypots use heterogeneous decoys and different types of security tools. Hence, they are more powerful in detecting attacks than the previous type. For example, Fan et al. (2019) designed a network of heterogeneous honeypots to obtain critical data from the adversaries.

3. Deception in single honeypots

It is important to note that if an adversary detects the presence of honeypots, their effectiveness can be compromised. This is especially true when anti-honeypot techniques are employed, as documented in various studies (Wang et al., 2017). When a savvy adversary exposes or recognizes a honeypot, it loses its value as an undercover resource within the network. In some alarming scenarios, rather than merely detecting the honeypot, adversaries might seize control of it, subsequently employing it as a weapon against the very network it was meant to protect. In cases involving particularly insidious threats like polymorphic or metamorphic malware (Popli and Girdhar, 2019), the stakes are even

higher. If a honeypot gets detected in such scenarios, it gives the adversary insights into the network's deceptive mechanisms. This newfound awareness may prompt adversaries to escalate their tactics, employing more sophisticated and evasive attacks to disrupt the network's functionality. To mitigate these risks, it becomes imperative to ensure that the deceptive techniques employed by honeypots are highly accurate and effective in reducing the likelihood of their detection. This necessity underscores the importance of developing and employing precise and reliable metrics to assess the effectiveness of honeypots' deceptive capabilities. These metrics are pivotal in refining honeypot deployments, and empowering network defenders to adapt and enhance their strategies in response to evolving threats. In the subsequent sections of this paper, we delve deeper into the intricacies of honeypot deception techniques, examining the methods used to emulate real network assets while maintaining a low profile. Furthermore, we explore the crucial role of metrics in quantifying the success of these deceptive measures, providing practitioners with the necessary tools to continually enhance honeypot resilience and overall network security (Nelson et al., 2009; Naeem et al., 2007).

We suggest some evaluation metrics to measure the effectiveness of single honeypots (i.e., without considering their communication with other honeypots in a honeynet). These metrics are used to measure the deception power of the mentioned techniques in this section. The suggested metrics are as follows:

- **Difference Amount deception discrepancy (DA):** This metric measures the difference between a honeypot and a real system. If a deception technique is concerned with simulating fake services, DA is the number of service responses that are not similar to the real service's response. On the other hand, in the case of deceptive data or files, DA is the amount of content which are not similar to the real ones. We can calculate DA as the ratio of the results that are different from a real result to the total number of tested requests.
- **Launched Attacks (LA):** The number of attacks directed at a honeypot serves as a valuable metric for assessing its deception efficacy. A honeypot's ability to lure adversaries is reflected in the volume of attacks it captures. A low LA value suggests that a honeypot may not be enticing enough to deceive potential attackers effectively.
- **Returned Adversaries (RA):** The number of adversaries who have launched attacks against a honeypot more than once is another metric for evaluating its attractiveness and deception power. If a honeypot lacks appeal, adversaries are less likely to initiate a second attack against it.
- **Second Session (SS):** Some adversaries do not launch attacks against the honeypot. However, they communicate with them to use them as a tool for further attacks. Hence, RA cannot measure this aspect appropriately, and we suggest counting the number of sessions established between the honeypots and an adversary who formerly communicated with that honeypot.
- **Wasted Time (WT):** An adversary's time spent communicating with a honeypot can also be used to assess its deception power. The higher the value of WT, the greater the deception power.
- **Using Ration (UR):** Some honeypots utilize specific data decoys to trace the adversary. To measure the effectiveness of these decoys we can calculate the ratio of the number of adversaries using the decoys to the number of adversaries accessing them. If an adversary accesses a decoy but does not use it, that decoy is not supposed to be a good one.
- **Traffic Volume (TV):** The volume of traffic forwarded to a honeypot serves as a metric to gauge its efficiency. It is important to emphasize that isolated honeypots may have a limited impact on network security. Since one of the primary objectives of a honeypot is to attract potential adversaries, a honeypot that consistently receives substantial traffic is generally regarded as more effective. However, it is essential to clarify that while attracting a significant volume of traffic can be an indicator of honeypot performance, the mere quantity of traffic alone does not guarantee enhanced network security. The relationship between honeypot traffic and security is more intricate, involving factors such as the nature of the traffic, adversary interactions, and the ability to detect and respond to threats effectively.
- **Confusion Matrix (CM):** This matrix is another common metric to evaluate classifying security methods such as honeypots. CM presents four possible cases for classifying the adversaries and legal users, which are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). In the case of honeypots, TP and FN are the number of adversaries detected as malicious and benign nodes, respectively, and TN and FP are the number of legal users detected as benign and malicious nodes, respectively. Since honeypots do not have a direct productive value, the legal users do not communicate with them. Hence, only the adversaries connect to honeypots, and the value of FP for a honeypot is almost zero. However, TP and FN can show the effectiveness of a honeypot. An efficient honeypot increases the value of TP and reduces the value of FN.

In the rest of this section, we present that The research about each technique is summarized as independent of the situation of a honeypot system in a network and how it cooperates with other honeypot systems. They are used to improve the deceptive power of a honeypot system without considering other honeypots. The research about each technique is summarized in Table 2. We also recommend which evaluation metrics can be used to measure the effectiveness of these techniques. The used metrics for each technique are also shown in Table 3, Fig. 4.

3.1. Advanced mimicking

An important point in designing a honeypot is to make it similar to real systems while keeping it attractive. The honeypots emulate some or all of the real systems operations and services to entice the adversary. The research about this mimicking technique mainly focuses on these two aspects (shown in Fig. 5):

- **Flawless imitation:** To avoid the adversary's suspicion, the honeypot must respond to the requests as everyone expects. To achieve this, the honeypot must emulate all the operating-system functionalities or generate error messages for non-implemented parts like a real error. Moreover, when a honeypot simulates a specific service, all the details of its protocols, such as the content of messages and the service port, must be the same as the real service. One of the methods that both adversaries and honeypot developers can use to check whether the responses of a system are similar to real production systems is network fingerprinting. Fingerprinting is the process of comparing the behavior of a fake system with a real one to analyze the differences. Dahbul et al. (2017) generated several fingerprinting requests and sent them to a real honeypot system. A comparative analysis offered suggestions to enhance the effectiveness of four commonly used honeypots: HoneyD, Dionaea, Kippo, and Glastopf. The recommendations encompassed various aspects, including carefully monitoring open ports, rectifying timestamps, and modifying certain scripts. Additionally, Naik et al. (2020) explored the use of fingerprinting attacks to optimize the honeypots, focusing on ten fields within TCP or IP packet headers. For instance, the study emphasized that developers should pay close attention to factors like the TCP window size and IP TTL value when simulating a real system. Equally vital is the concept of making the honeypot discoverable, ensuring that its appearance closely mirrors that of a production system. In this context, Chen and Buford (2009) introduced a honeypot database system that search engines can crawl, a strategy that helps the honeypot closely resemble real

Table 2
The researches focusing on improving single honeypots deception power.

Deception Technique	Main Method	Suggestion
advanced mimicking	Flawless imitation	Pay attention to the open ports, timestamps, and scripts (Dahbul et al., 2017). Pay attention to TCP and IP header fields (Naik et al., 2020). Make the honeypot discoverable by the search engines (Chen and Buford, 2009). Learning to behave real using neural networks (Siniosoglou et al., 2020)
	Attractive vulnerabilities	Use PHP and MySQL database services (Shumakov et al., 2017). Use FTP and MySQL database services (Perevozchikov et al., 2017). Create intelligent exploitable database services (Huang et al., 2020).
Fake Cooperation	Showing the attack success	Pretend to be compromised and leak some fake data (Chen and Buford, 2009) Use game models to decide when to pretend to be compromised (Wagener et al., 2009).
	Pretending to help the adversary	Mimic the activities of a compromised bot (Zhuge et al., 2007). Simulate the bot behavior and communicate with the other botnet members (Jiang et al., 2010). Use game models to cooperate with a botnet (Hayatle et al., 2012) effectively.
Deceptive Database	Looking real	Combine different parts of real filenames and fill the files with websites contents (Rowe, 2006). Use deep learning approaches to check the reality (Abay et al., 2019). Follow ontology concepts to fill the files based on meta-centrality metric (Chakraborty et al., 2019). Create the fake non-textual files based on probabilistic logic graph modeling (Han et al., 2021).
	Looking protected	Generate files with attractive extensions and fill them with random numbers (Rowe, 2006). Show a fake authentication process (Fraunholz and Schotten, 2018a). Prevent weak attacks against the data (Chen and Buford, 2009).
	Looking consistent	Create a copy of database to apply the changes (Chen and Buford, 2009). Store all the changes of each adversary and restore them when needed (Akingbola et al., 2015).
Subtle Interruptions	Connection restriction	Limit the number of new connections an infected host can create (Dantu et al., 2007). Limit the established connections queue length (Sun et al., 2017).
	Causing extra probes	Keep all the possible ports open (Gjermundrød and Dionysiou, 2015). Add connected decoys to the network to make it bigger (Shakarian et al., 2014). Use virtual topologies to make the network bigger (Achleitner et al., 2017). Utilizing machine learning techniques to waste the adversary's time (Pauna et al., 2018; Suratkar et al., 2021; Dowling et al., 2018)
Honeytoken Bait	Generating the honeytokens	Change the first character of a real password and add extra characters to its end (Juels and Rivest, 2013). Assign similarity score to honeytokens to evaluate them (Bercovitch et al., 2011). Change some of the characters from uppercase to lowercase and vice versa (Suryawanshi et al., 2017). Pay attention to the flatness of the generator algorithm (Erguler, 2016).
	Using the honeytokens	Use passive and active honeytokens to trace internal and external adversaries (Wegerer and Tjoa, 2016). Use honeytokens with beacons to trace the location and time (Bowen et al., 2009). Use honeytokens to alert when a Java script is compiled or executed (Park and Stolfo, 2012). Use honeytokens to detect different attack phases (Akiyama et al., 2018). Use honeytokens to detect the relation between botnet's members (Ja'fari et al., 2021).
Traffic Redirection	Interfering after IDSs detection	Use game models to decide which traffic flow to redirect (La et al., 2016). Redirect malicious traffic to a fake database (Selvaraj et al., 2016). Redirect malicious traffic to a dynamic honeypot in a software defined network (Park et al., 2019).
	Interfering after honeypots detection	Redirect the bots detected by honeypots to another honeypot (Ja'fari et al., 2021). Clone a virtual honeypot when redirection is required (Biedermann et al., 2012).
	Interfering after other methods	Detect flooding traffic with entropy checking and then redirect it (Sardana and Joshi, 2009). Detect malicious USB devices with user's feedback and then redirect their traffic (Tian et al., 2015).
	Improving the performance	Redirect powerful and weak attacks to different honeypots (Wang and Wu, 2019). Redirect interesting attack scenarios to another honeypot (Fan and Fernández, 2017).

Table 3
Deception techniques and their evaluation metrics.

Technique	Evaluation Metric							
Name	DA	LA	RA	SS	WT	UR	TV	CM
advanced mimicking	✓	✓	✓	✓	✓	✗	✓	✓
Fake Cooperation	✗	✗	✗	✓	✓	✗	✗	✓
Deceptive Database	✓	✓	✗	✗	✓	✗	✗	✓
Subtle Interruptions	✗	✗	✓	✓	✓	✗	✓	✓
Honeytoken Bait	✗	✗	✗	✗	✗	✓	✗	✓
Traffic Redirection	✗	✗	✓	✓	✓	✗	✓	✗

production systems. Siniosoglou et al. (2020) proposed a honeypot for industrial networks, NeuralPot, that uses neural networks to learn how to behave.

- **Attractive vulnerabilities:** Some security gaps and vulnerabilities are more attractive to adversaries than others. Hence, a honeypot can pretend that it has these vulnerabilities to attract more adversaries. When the number of adversaries connecting to the honeypot increases, the collected data will also increase and contain more significant information about the attack patterns and adversaries' behavior. Shumakov et al. (2017) aimed to find the most vulnerable web services from four websites. The results concluded that PHP and MySQL are attractive web services. One can replace other unattractive services on a honeypot with these services and make the honeypot attract more adversaries. Perevozchikov et al. (2017) tried to provide attractive services, such as FTP and MySQL database, by the honeypots to lure more adversaries. Huang et al. (2020) proposed a method to automatically and intelligently use different exploitable vulnerabilities in a database to deceive the adversaries.

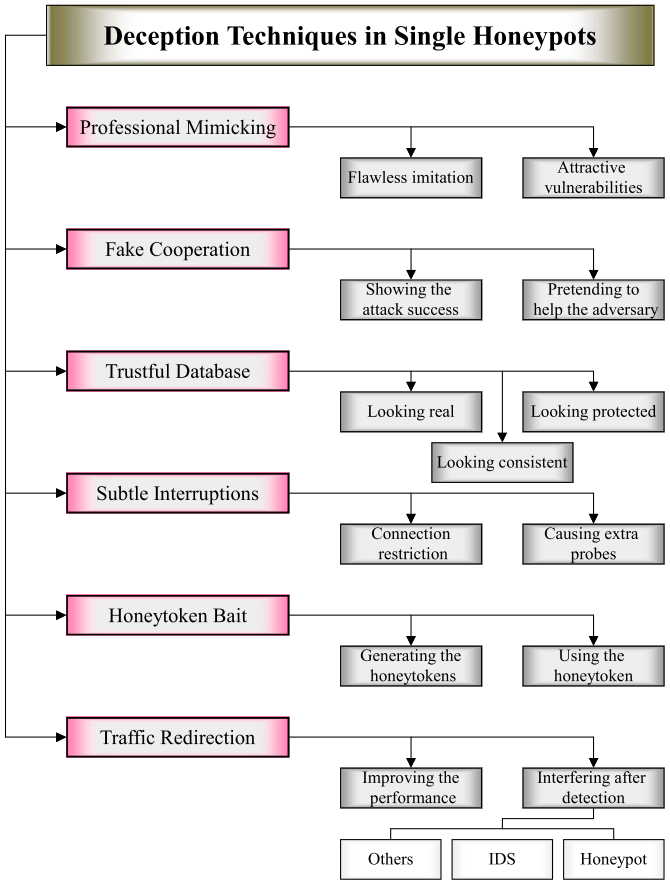


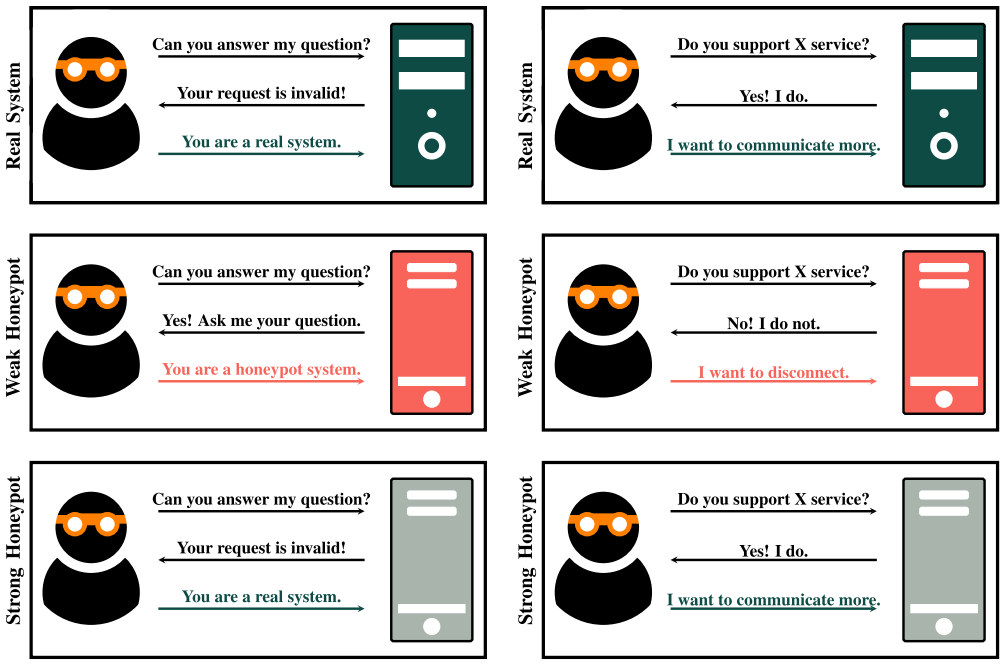
Fig. 4. The taxonomy of single honeypots deception techniques.

We can analyze the effectiveness of this deception technique by DA, LA, RA, and WT. Attractive vulnerabilities can result in a high number of attacks against the honeypots and bring the adversaries back to launch even more attacks. The adversary also consumes more time communicating with attractive vulnerabilities. On the other hand, flawless mimicking can cause less DA and show a honeypot's strength.

3.2. Fake cooperation

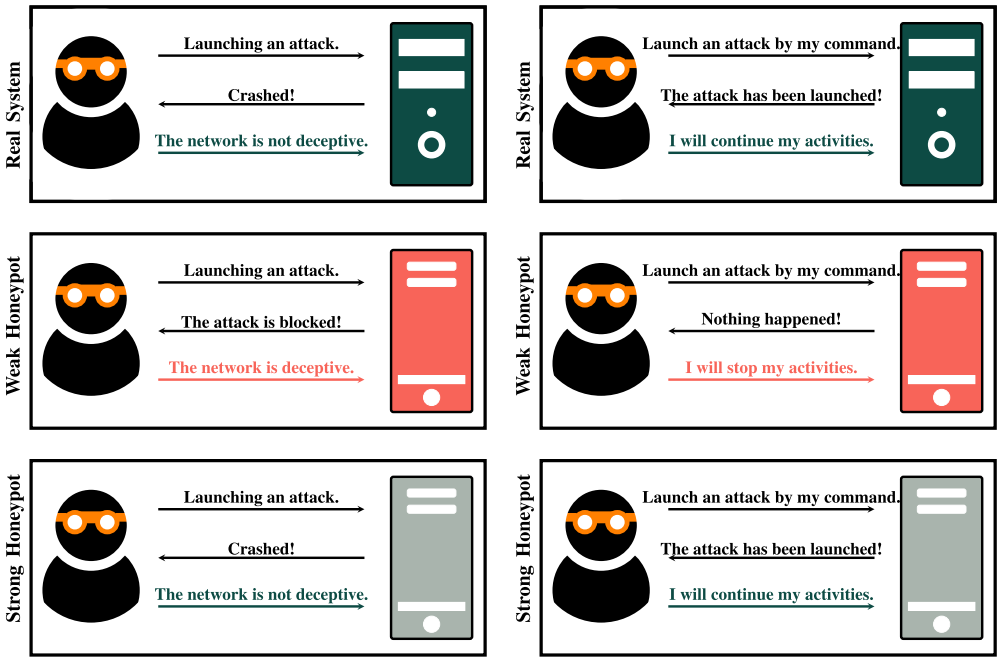
Cooperating with the adversaries is one of the ways to fool them. This cooperation can be performed in two different ways (shown in Fig. 6):

- **Showing the attack success:** In this type, the honeypot pretends that the adversary's attack will succeed. A good deception idea is to show the adversary that the attack steps are progressing, and its target, which is actually a honeypot, is crashed under its final attack step. Chen and Buford (2009) proposed a database honeypot that can be the target of SQL injection attacks. This honeypot pretends to be compromised by SQL injection attacks and leaks some fake data to show the adversary's spurious success. Wagener et al. (2009) modeled the communication between the adversary and a honeypot as a two-player game, in which the adversary attempts to compromise the hosts with the minimum possible cost, and the honeypot aims to learn as much as possible from him/her. This game aims to find the situations in which the honeypot can pretend to be compromised by the adversary without facing dangerous threats.
- **Pretending to help the adversary:** In this type, the honeypot goes along with the adversary to pretend it is helping him/her launch the attack. This type is used when the network is under a botnet attack or similar cyber threats, in which the adversary compromises several network hosts to gather an army. If a member in this army does not follow the adversary's command, it tries to gather another army. Hence, the honeypot pretends to be compromised and to obey the adversary's commands. This deception technique is hard to design. Many complicated situations must be considered to make the honeypot undetectable. On the other hand, pretending to follow the adversary while causing no real damage to the network is challenging. Zhuge et al. (2007) proposed HoneyBot, a honeypot that mimics the activities of a compromised



(a) Flawless imitation (b) Attractive vulnerabilities

Fig. 5. The scenarios of the advanced mimicking technique.



(a) Showing the attack success (b) Pretending to help the adversary

Fig. 6. The scenarios of the Fake Cooperation technique.

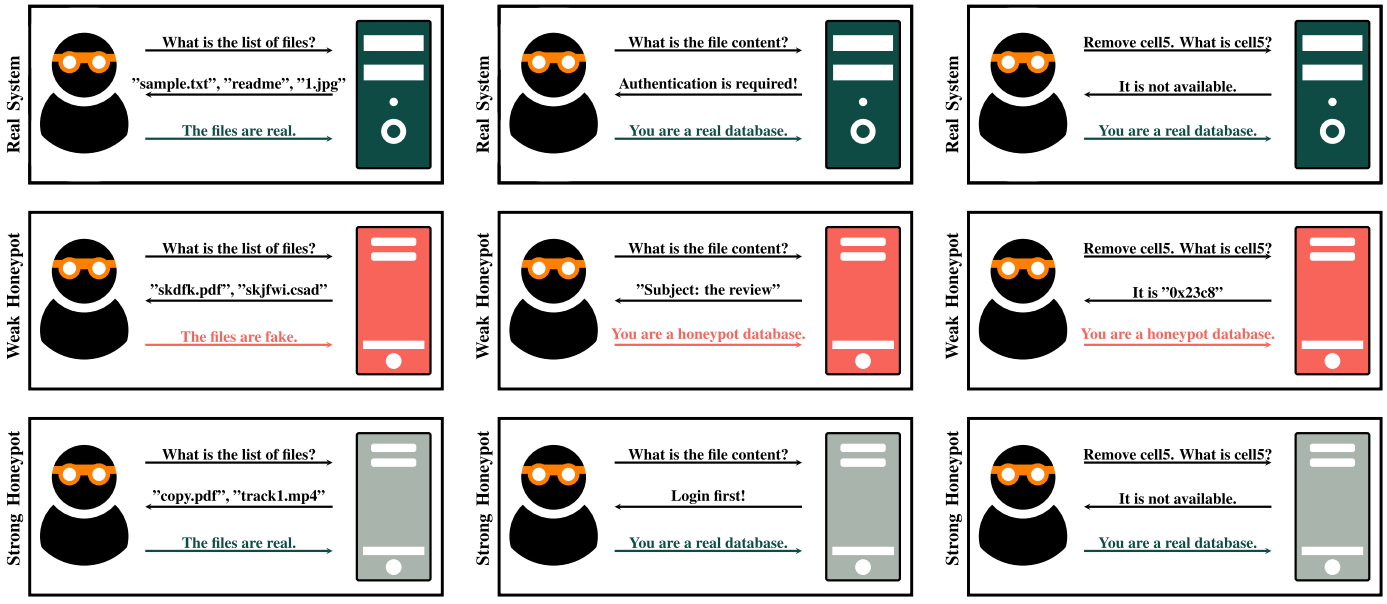
host by pretending to be a bot. Jiang et al. (2010) also proposed a tracking tool for botnets in which the system simulates the bot behavior and communicates with the other botnet members. Moreover, Hayatle et al. (2012) modeled the botmaster and honeypot interaction as a Bayesian game. In each step, the honeypot decides to follow the botmaster's command or to ignore it, and the botmaster chooses between these actions: testing the bot, avoiding further communications, or sending an attack command. Using this model, the developer can find the best strategy that can increase the adversary's trust in the honeypots cooperating with the adversary.

The efficiency of this deception technique can be measured by the time that the adversary wastes on the network (i.e., WT). If the adversary feels that the honeypot is cooperating with him/her to succeed in the attack goal, he/she will spend more time communicating with it. We can also use SS to measure the power of deceptive cooperation.

3.3. Deceptive database

Many adversaries are interested in gaining access to confidential information. Therefore, a honeypot must be able to produce invalid attractive data that deceive the adversary while keeping itself unrecognizable. But this process is challenging according to the following aspects (shown in Fig. 7):

- **Looking real:** The fake data content must be as similar to real data as possible. Meaningless names, unusual data structures, and trifling file contents are samples of useless fake data that reveal the identity of a honeypot. Rowe (2006) proposed a method to generate meaningful but invalid data to use in honeypots. In this method, the filenames are created by combining different parts of real filenames, and the file contents are generated by extracting data from different websites. Abay et al. (2019) checked the authenticity of fake data by deep learning approaches. Chakraborty et al. (2019) proposed FORGE; a fake data generator. FORGE creates k different fake but believable files for each real file to reduce the probability of real data leakage. The content of a fake file is constructed based on a meta-centrality metric regarding ontology concepts to look real. Since FORGE can only generate textual data, Han et al. (2021) proposed another method to generate trustful data, which can also create non-textual content, such as diagrams, equations, and tables. This method first models a document with a probabilistic logic graph that can fully express its different parts. Then a greedy algorithm is executed to generate fake graphs regarding the real ones, and finally, the fake graphs are converted into fake documents.
- **Looking protected:** The fake data must not be easy to access for the adversary. As critical data are hard to access, if the adversary collects it without effort, it becomes suspicious and discovers that the collected data are worthless. For example, encryption makes the stored fake data in honeypot database more valuable and real. Because when the adversary faces the plain data, it will have no motivation to continue the attack on the current system. Rowe (2006) suggested that we can create files with attractive extensions such as ".enc" and ".cyc" and fill them with random numbers for more attractiveness. Another example is the authentication process for granting access. The data that need authentication to be accessed can encourage the adversary to launch an attack against them. However, the authentication process must not be hard to penetrate. Fraunholz and Schotten (2018a) used a fake authentication page for the proposed deceptive web server to lure and attract more adversaries. Chen and Buford (2009) used another method to lure the adversary about the protected data. This research uses a database honeypot to mitigate some weak SQL injection attacks. Hence, the adversary becomes unsuspected about the existence of a honeypot database.
- **Looking consistent:** The changes made by the adversary must apply to the fake data so that it will observe the updated data not only in the current session but also in its next sessions. If the adversary finds any conflicts in its communication with the honeypots, it will be suspicious of a deceptive mechanism in the network. Chen and Buford (2009) designed a honeypot to detect and mitigate SQL injection attacks. When the adversary modifies the fake database in this database honeypot, the changes are applied to a copied version of that. Hence, the adversary will be sure about the consistency



(a) Looking real (b) Looking protected (c) Looking consistent

Fig. 7. The scenarios of the Deceptive Database technique.

of the database. Akingbola et al. (2015) also proposed a stronger method, in which a table is considered for each adversary to store their changes in the database. Hence, when that adversary returns, he/she will see the previous changes. Their IP and MAC addresses identify the adversaries in this method.

The effectiveness measurement metrics of this technique are the same as the *attractive vulnerabilities* technique. LA, RA, and WT can measure the trustworthiness of the honeypot fake data from the adversary's point of view.

3.4. Subtle interruptions

Cyber-attacks continue to evolve in sophistication, presenting an ever-growing challenge for detection within the dynamic threat landscape. Honeypots emerge as a formidable tool in this strategic defense approach. Functioning as specialized decoy systems meticulously designed to emulate real environments, honeypots lure and deceive potential attackers. These decoy systems employ various techniques to introduce deliberate delays and obstacles, significantly hindering adversaries' progress. Within the deception technique of "Subtle interruptions," a notable strategy is using tarpits. Tarpits are an ingenious mechanism to ensnare adversaries by intentionally slowing down their progress. These digital quagmires are designed to waste an adversary's time and resources, compelling them to navigate virtual mazes that impede their advancements. These delay tactics encompass a range of deceptive maneuvers, such as simulating intricate system responses, injecting faux complexities, or dynamically altering network configurations. For instance, a honeypot may convincingly mimic a sluggish network connection or deliberately introduce delays in the response time of specific applications. These nuanced strategies, including tarpits, effectively consume the adversary's time and resources, diminishing the impending attack's overall impact. However, the successful execution of these delay methods hinges on meticulous precision. Any misstep in their implementation risks inadvertently revealing the honeypot's true nature, thus diminishing its efficacy as a deceptive tool. Striking the delicate balance between hindering the adversary's progress and preserving the honeypot's covert status remains a nuanced challenge in cybersecurity defense (Rowe et al., 2007; Dalamagkas et al., 2019; Bringer et al., 2012). Several cyber-attacks diminish over time. Hence, wasting the adversary's time or slowing him/her down can significantly reduce the attack effects. Honeypots can use different methods to delay the adversary. However, these methods must be performed carefully to avoid revealing the existence of the honeypot. The subtle interruptions technique can be performed using two main methods (shown in Fig. 8):

- **Connection restriction:** Restricting the adversary's connections is a way to slow him/her down. Dantu et al. (2007) proposed a method to interrupt malware propagation by limiting the number of new connections an infected host can create. Sun et al. (2017) proposed a framework for deploying honeypots, in which the queue length that stores the established connections is limited. This can interfere the adversary and cause interruptions.
- **Causing extra probes:** Another way of interrupting the adversary is to make its target space larger, leading to extra probing attempts. Gjermundrød and Dionysiou (2015) proposed a honeypot, called CloudHoneyCY, in which all possible ports are open, and if the adversary communicates through them, the honeypot will respond with garbled messages. This technique may waste time since the adversary probes all the active ports. Shakarian et al. (2014) added distraction clusters, which are connected decoys, at specific network points to amaze the adversary and make the network seem bigger. Achleitner et al. (2017) also used a similar technique and proposed a honeypot-based system that delays the adversary by constructing virtual topologies for the network that takes a long time to be scanned and probed. Pauna et al. (2018), Suratkar et al. (2021), and Dowling et al. (2018) proposed Q-learning and reinforcement learning mechanisms, which are a type of machine learning techniques by which the honeypot learns how to interact with the adversary to waste its time.

The efficiency of this deception technique can be measured by the number of adversaries who communicate with the honeypot for more than one session (SS). If the interruptions are not usual, the adversary will not return to that honeypot.

3.5. Honeytoken bait

Honeytoken is a fake piece of information or resource that helps the honeypots trace the adversary. Since honeytokens do not contain

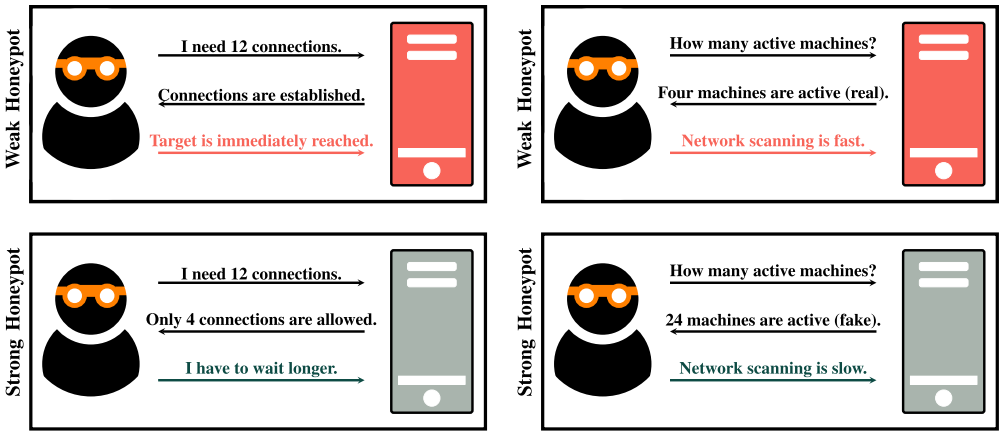


Fig. 8. The scenarios of the Subtle Interruptions technique.

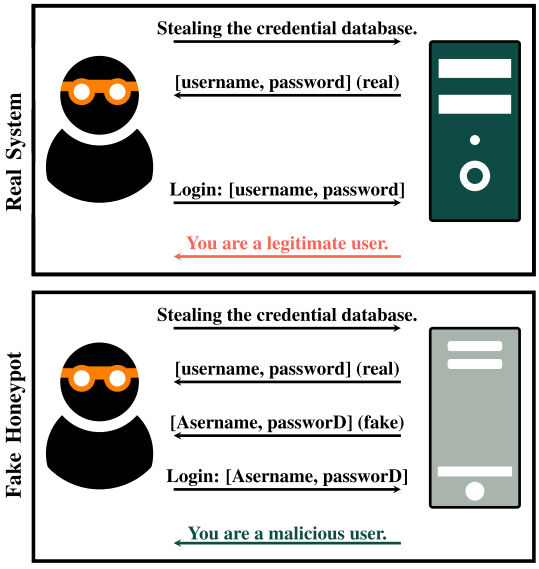


Fig. 9. The scenario of the Honeytoken Bait technique.

valid information, anyone using them is considered to be an adversary or an illegal user. For example, a honeypot authentication system can store several fake credential pairs in its database. If an adversary penetrates the database and gains access to these credentials, it will soon use them to login to the system. Using these specific credentials claims that their owner is a malicious user. These fake credentials are samples of honeytokens (Shabtai et al., 2016; Msaad et al., 2022). A sample scenario in using honeytokens is shown in Fig. 9. A honeytoken, a deceptive digital artifact, is a potent instrument within cybersecurity. Crafted deliberately to masquerade as legitimate data or resources, its primary function is to empower honeypots in identifying and tracing potential adversaries. Distinctively, honeytokens diverge from conventional data in their essence; devoid of genuine value, they lack the authenticity associated with valid information. This intrinsic characteristic endows honeytokens with a distinct purpose: anyone who engages with them is promptly marked as an adversary or an illicit user within the system. Consider the scenario of a honeypot authentication system: ingeniously, it embeds a repository of counterfeit credential pairs within its database. Intruders who infiltrate the system and gain access to these deceptive credentials subsequently deploy them to gain entry. The system's response to such actions is a telltale sign that the user's intent is malicious. These intentionally falsified credentials epitomize the essence of honeytokens, embodying the fusion of deception and strategic insight. Illustrating their application, envision a practical use case depicted in Fig. 9. Here, the trajectory of an adversary's actions unfolds, interwoven with the deployment of honeytokens. This visual representation offers a tangible glimpse into the intricate dance between the digital decoy and the malicious actor. With each interaction, honeytokens reveal their prowess not only as a tool of deception but as a mechanism that empowers defenders to pinpoint and respond to threats with heightened accuracy. Incorporating honeytokens into the cybersecurity landscape exemplifies a dynamic strategy transcending mere diversion. It stands as a testament to the evolution of defense mechanisms, showcasing how innovation and insight combine to outmaneuver adversaries in an ever-evolving digital battlefield (Mokube and Adams, 2007; Srinivasa et al., 2020).

- **Generating the honeytokens:** The landscape of honeytoken generation has witnessed a proliferation of methodologies, each contributing its unique approach to bolstering the efficacy of this deceptive cybersecurity technique. As highlighted by Juels and Rivest (2013), one of these strategies involves the manipulation of user passwords to form honeytokens. A honeytoken is ingeniously crafted by altering the first character and appending additional characters at the end of a legitimate password. This variant of honeytokens, commonly known as "honeywords," capitalizes on the premise of diverting unauthorized users toward these fabricated entry points. Extending the narrative, Bercovitch et al. (2011) introduced an innovative leap called "HoneyGen." This automatic honeytoken generator revolutionized the way honeytokens are conceived and utilized. What sets HoneyGen apart is its ability to construct honeytokens that seamlessly mirror authentic data. Operating on the principle that deceptive elements are most effective when closely resembling real ones, HoneyGen assigns a unique score to each generated honeytoken. This scoring mechanism quantifies the similarity between a honeytoken and actual data, ensuring the decoy is compelling enough to ensnare potential threats. The significance of these methodologies extends beyond their technical intricacies. They underscore the dynamism within cybersecurity, wherein innovation and ingenuity are harnessed to deceive and outsmart adversaries. These honeytoken generation techniques embody the continuous evolution of defense mechanisms, adapting to the ever-shifting landscape of cyber threats. By meticulously tailoring honeytokens through approaches like those proposed by Juels and Rivest (2013) and Bercovitch et al. (2011), the realm of cybersecurity augments its capabilities to deter, detect, and respond to potential breaches with heightened precision and agility. The landscape of honeytoken generation is a dynamic arena where innovation thrives, evident in diverse methodologies that seek to

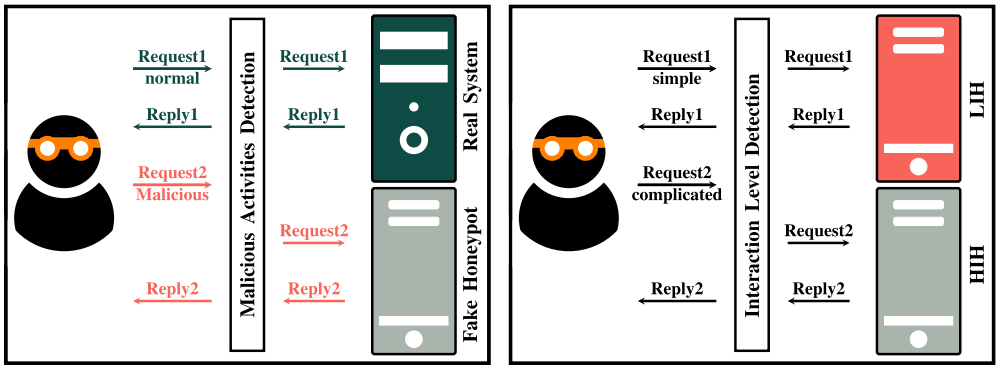
elevate the precision and effectiveness of these sophisticated cybersecurity instruments. Delving deeper into this realm, Suryawanshi et al. (2017) introduced a novel approach that revolves around strategic alterations within a user's authentic password. This intricate technique involves transforming characters at specific indices, and toggling them between uppercase and lowercase. Notably, this method offers heightened efficiency compared to its predecessors and retains a crucial attribute—meaningfulness. If the original password holds significance, the resulting honeywords echo this significance. This ingenious touch significantly diminishes the likelihood of arousing suspicion among potential adversaries, rendering Honeytokens even more adept at deception. Running parallel to this innovation, Erguler (2016) embarked on a similar trajectory, advocating for a method that only stores the genuine password and the index of the altered character within the database. Beneath the technical surface, the distinctive contribution of this work lies in the concept of "flatness." This principle asserts that the generated honeyword should closely mirror passwords crafted by human users, effectively erasing the boundary between authentic and counterfeit. The pursuit of flatness emerges as a strategic imperative, enhancing the illusion and rendering the distinction between genuine and fabricated honeytokens almost imperceptible. Transitioning from the theoretical to the practical realm, researchers are committed to maximizing honeytokens' efficacy in real-world contexts.

- **Using the honeytokens:** Wegerer and Tjoa (2016) significantly advances this objective by meticulously outlining the implementation steps for a MySQL honeypot database server. This cutting-edge server integrates passive and active honeytokens, catering to the distinct needs of tracing internal and external adversaries. This multifaceted approach underscores the versatility of honeytokens as a cybersecurity strategy, encompassing various scenarios and adversaries. These methodologies embody a relentless pursuit of innovation within the deception landscape. As the cybersecurity terrain continues to evolve, these techniques represent the symbiotic relationship between ingenuity and the proactive defense against digital adversaries. By further refining honeytoken generation techniques, as proposed by Suryawanshi et al. (2017) and Erguler (2016), and seamlessly integrating them into practical applications like Wegerer and Tjoa (2016), researchers provide a beacon for the continued advancement of cybersecurity strategies. Within the domain of honeytoken implementation, researchers have propelled the evolution of this cybersecurity technique by introducing innovative systems and methodologies that augment its effectiveness and versatility. A pioneering contribution by Bowen et al. (2009) materialized in the form of the D^3 system. Central to this innovation is the integration of a beacon within each honeytoken. This concealed signal transmitter serves as a revolutionary element, enabling honeytokens to establish a covert line of communication with the D^3 system. By effectively transmitting information regarding when and where honeytokens are activated, the D^3 system significantly refines the precision of threat identification and extends the purview of insights derived from honeytoken interactions. This level of real-time reporting fundamentally transforms honeytokens from mere decoys into instruments of actionable intelligence. Expanding the horizon of beacon-based applications, Park and Stolfo (2012) delves into their deployment as alert triggers during the compilation or execution of counterfeit Java source codes, including honeytokens. This ingenious utilization taps into the dynamic execution environment of Java, utilizing it to generate real-time alerts whenever deceptive or counterfeit code, like honeytokens, is executed. This approach enhances the real-time responsiveness of cybersecurity measures and exemplifies honeytokens' multifaceted potential beyond passive deception. Further enriching the repertoire of honeytoken applications, Akiyama et al. (2018) harnessed the power of honeytokens to glean insights into the intricate phases of web-based attacks. By strategically disseminating honeytokens across distinct stages of an attack, they successfully harvested invaluable data, providing a comprehensive understanding of the attacker's behavior, tactics, and potential motives. This innovative approach underscores Honeytokens' versatility as a proactive threat intelligence-gathering tool. Venturing into groundbreaking territory, Ja'fari et al. (2021) introduced the "activator" honeytokens concept. This novel paradigm shift extends the notion of deception by introducing a mechanism to uncover relationships between diverse entities within the cyber ecosystem. Specifically, the activator honeytoken acts as a conduit to reveal the intricate connections between the Mirai loader and other bots (Om Kumar and Sathia Bhama, 2019), shedding light on the intricate dynamics of botnet interactions. To comprehensively evaluate the efficacy of these progressive approaches, the Utilization Rate (UR) emerges as the most pertinent metric. Unlike mere access to honeytokens, which might not accurately mirror an adversary's true intent, utilizing honeytokens provides a more authentic measure of their effectiveness. Utilizing a honeytoken signifies a high level of deception, underscoring the significance of UR as a robust indicator of honeytoken's impact in proactively thwarting potential threats.

- **Unveiling the Multidimensional Tapestry of Honeytokens:** With their innovative design and evolving applications, Honeytokens has emerged as a multifaceted cybersecurity strategy that goes beyond traditional deception techniques. This discussion ventures into uncharted dimensions of honeytoken utilization, shedding light on novel approaches that enhance their potency in safeguarding digital landscapes. A groundbreaking concept introduced by Bowen et al. (2009) involves the integration of beacons within honeytokens, giving birth to the ingenious D^3 system. This transformative leap propels Honeytokens into an active role within cybersecurity defenses. The beacon, nestled within each honeytoken, establishes an encrypted communication channel with the D^3 system. This real-time information exchange empowers defenders with insights into the very heart of adversary activities—where and when honeytokens are triggered. This dynamic feedback mechanism redefines honeytokens from passive decoys to living entities, providing unprecedented real-time threat intelligence. By promptly identifying threats and vulnerabilities, the D^3 system elevates honeytoken deployment to a dynamic, proactive defense strategy. Beyond the conventional boundaries of deception, Park and Stolfo (2012) explored the innovative application of beacons to honeytokens in a distinct context. Their approach leverages beacons to trigger alerts upon the compilation or execution of deceptive Java source code, which includes honeytokens. This proactive response mechanism introduces an active element to deception tactics. Generating alerts upon deceptive code execution merges honeytokens into the active defense paradigm. This amalgamation showcases the intricate harmony between honeytokens and real-time threat identification, where honeytokens deceive and trigger responsive actions to thwart potential threats. Expanding the horizons even further, Akiyama et al. (2018) delved into the untapped potential of honeytokens as sources of insight during various stages of web-based attacks. Strategically deployed honeytokens across different attack phases offered invaluable data that uncovered the attacker's tactics, behavior, and underlying motivations. Applying honeytokens as strategic markers throughout an attack enhanced threat intelligence and transformed them into instruments of proactive attack analysis. This strategic approach positions honeytokens as agents illuminating attack narratives, enhancing incident response strategies with a comprehensive understanding of adversary tactics. An innovative stride by Ja'fari et al. (2021) introduced "activator" honeytokens, a revolutionary concept transcending deception's

boundaries. These tokens deceive and serve as vehicles to uncover relationships within intricate cyber ecosystems. The activator honeytoken explicitly explores connections between the Mirai loader and other bots, shedding light on botnet dynamics. This pioneering application showcases honeytokens' transformative potential, from mere deception tools to strategic instruments that unravel the complex web of cyber interactions. Incorporating these discussions in the paper enriches its narrative by showcasing honeytokens' evolving role. These insights underscore how innovation can reshape traditional techniques, highlighting the synergy between cutting-edge strategies and the proactive defense against dynamic cyber threats.

- **Honeytokens: Abundant Web Application Exploits:** The extensive literature surrounding web applications offers a rich tapestry of research focused on implementing honeytokens. These deceptive elements, in the form of various values, are ingeniously woven into the intricate structure of web applications, serving as vital components in the arsenal of cybersecurity strategies. This discussion delves deeper into the multifaceted dimensions of honeytoken deployment within web applications, showcasing their pivotal role in thwarting cyber threats. When strategically integrated into web applications, Honeytokens take on diverse forms tailored to confound and deceive potential adversaries (Qin et al., 2023). These artificial elements manifest as decoy values that include HTTP parameters, URLs, forms, cookies, HTML elements, permissions, and even fabricated user accounts. By seamlessly embedding these deceptive elements within the architecture of web applications, defenders create a labyrinth of false trails that attackers unwittingly follow, ultimately exposing their tactics, objectives, and methods. Deceptive HTTP parameters, seamlessly integrated into the requests and responses of web applications, act as tantalizing bait that adversaries inevitably engage with. These illusory elements subtly guide attackers towards a predetermined path, allowing defenders to understand the adversary's behavior, probing techniques, and even potential points of vulnerability. Similarly, deceptive URLs, often camouflaged as legitimate components of the application's structure, entice attackers into interactions that yield valuable insights into their navigation patterns and exploratory strategies. Incorporating honeytokens as deceptive forms, cookies, and HTML elements adds a layer of complexity to the web application's façade (White et al., 2014). Attackers, drawn to these fabricated entities, unknowingly leave behind a trail of interactions that provide defenders with invaluable clues about their intent and objectives. By engaging with these decoy elements, adversaries inadvertently disclose critical aspects of their modus operandi, enabling defenders to adjust their cybersecurity measures proactively. However, the application of honeytokens extends beyond mere interaction patterns. Researchers have also explored their utility in permissions and user accounts. Defenders guide adversaries toward engaging with seemingly valuable targets by introducing fabricated permissions or user accounts. This interaction exposes the attacker's intent and aids defenders in mapping potential pathways and objectives that adversaries may pursue. The holistic deployment of honeytokens within web applications represents a symphony of deception orchestrated precisely to extract insights from attacker behaviors. Each decoy value, meticulously interwoven within the web application's structure, contributes to a larger narrative that reveals the tactics and motivations of adversaries. The dynamic interaction between innovation, strategic insight, and the digital landscape creates a powerful defense mechanism that bolsters cybersecurity in the face of evolving digital threats (Papaspirou et al., 2021; Jonsson and Marteni, 2022). Here's an expanded description of each honeytoken type, along with more information:
  - Decoy HTTP Parameters: Decoy HTTP parameters are forged data elements strategically inserted into a web application's HTTP requests and responses. These false parameters mimic genuine data and are designed to attract the attention of potential attackers. As attackers interact with these decoy parameters, defenders collect valuable information about the nature of their interactions. This includes details about the target endpoints, the data they seek, and the methods they employ to manipulate the application. The analysis of these interactions assists defenders in uncovering the tactics and objectives of adversaries, enabling them to fine-tune defense mechanisms (Izagirre, 2017).
  - Deceptive URLs: Deceptive URLs are fabricated web addresses that mirror legitimate paths within a web application. These URLs are meticulously crafted to appear integral to the application's structure. Attackers who navigate these deceptive URLs provide defenders with insights into their exploration patterns. This helps defenders understand which application attackers find attractive or potentially vulnerable parts. By analyzing the frequency and sequence of interactions with deceptive URLs, defenders gain a deeper understanding of attacker motivations and potential attack vectors (Sahin et al., 2022).
  - Decoy Forms: Decoy forms encompass synthetic input fields strategically embedded within legitimate web forms. These fields appear authentic, enticing attackers to engage with them. The information submitted by attackers in these forms offers defenders valuable insights into their intent and objectives. Defenders can learn about the specific data points attackers seek, the types of attacks they are attempting, and the tactics they employ to exploit vulnerabilities. This information enables defenders to bolster security measures, fortifying the application against potential threats (Bowen et al., 2009).
  - Deceptive Cookies: Deceptive cookies are fabricated data tokens injected into a user's browser upon interaction with the application. When attackers inadvertently interact with these cookies, defenders gain insights into their behavior. This includes details such as the pages they visit, their actions, and their patterns. By analyzing the information collected from deceptive cookies, defenders can discern attacker motivations, such as reconnaissance efforts or attempts to manipulate session data. This understanding informs proactive defensive measures (Sun et al., 2020).
  - HTML Element Baits: HTML element baits involve strategically inserting synthetic HTML elements within the web application's source code. These elements are typically hidden from regular users but are enticing to potential attackers. As attackers interact with these hidden elements, defenders gain insights into their tactics and techniques. This encompasses details about attackers' methods to explore the application's structure, potentially identifying areas of interest or vulnerabilities. The analysis of interactions with HTML element baits informs defenders about potential attack vectors and guides their defensive strategies (Voris et al., 2013).
  - Fabricated Permissions: Fabricated permissions involve creating synthetic access levels granted to specific roles or users within the application. Attackers who attempt to exploit these fabricated permissions inadvertently reveal their objectives as they engage with these fictitious access levels. Defenders can gain insights into the attackers' intended pathways, potential targets, and the extent of their knowledge about the application's permission structure. This information enables defenders to adopt security measures to counter specific attack strategies and limit potential breaches (Domingue et al., 2014).
  - Fictitious User Accounts: Fictitious user accounts are artificial profiles introduced into the application's user base. These fabricated accounts appear as potential targets to attackers attempting unauthorized access. As attackers interact with these deceptive accounts, defenders gather insights into their methods of infiltration, the paths they take, and their intentions. By analyzing the interactions with fictitious user accounts, defenders can tailor security



(a) Interfering after detection (b) Improving the performance

Fig. 10. The scenarios of the Traffic Redirection technique.

measures to address the specific attack vectors observed and enhance overall system security (Jones, 2016).

In web applications, it is essential to note that our current section, methods, does not cover more advanced strategies, particularly those that involve artificial intelligence (AI). Specifically, techniques such as utilizing deep learning algorithms to generate relational honeydata have not been explored in the existing content. Incorporating AI-based approaches introduces a layer of complexity and adaptability to honeytoken generation. Deep learning models, for instance, can potentially enhance the mimicry of genuine user behavior, leading to more convincing and contextually relevant honeytokens. This method fortifies the honeypot's deceptive capabilities and aligns with the evolving landscape of cyber threats that often leverage sophisticated techniques. Deep learning is a powerful tool for generating relational honeydata, which involves creating fake user interactions within a web application to deceive potential attackers. To achieve this, deep learning algorithms such as recurrent neural networks (RNNs) or long short-term memory networks (LSTMs) can be used to capture sequential dependencies in data, making them suitable for modeling relational aspects. By training on legitimate user interactions, deep learning models can learn the inherent patterns in user behavior, allowing them to generate honeytokens that closely resemble authentic user actions. Further, deep learning models excel in capturing contextual nuances, enabling the generation of honeytokens that maintain coherence within a sequence or relationship, making them more convincing to potential attackers. The term "relational honeydata" implies that the generated honeytokens have been individually crafted to resemble authentic actions and exhibit relationships or dependencies between them. A relational honeydata sequence might include a user logging in, navigating through various pages, and completing a transaction—all intricately linked to form a coherent and believable user journey. One of the strengths of deep learning is its adaptability to new patterns and evolving contexts. As attackers modify their strategies, deep learning models can be retrained to understand and generate honeytokens that adapt to these changes. This adaptability is particularly valuable in the dynamic landscape of web application security, where attack techniques constantly evolve (Mohan et al., 2022).

3.6. Traffic redirection

Traffic redirection in a network containing one or more honeypots commonly takes place in two situations (shown in Fig. 10):

- **Interfering after detection:** In some situations, the adversary is detected in the network. However, we do not block him/her from observing more activities from him/her or waste its time. Hence, we must prevent the adversary from communicating with critical resources in the network. Some honeypots use the traffic redirection technique to change the destination of the adversary's traffic and send it toward themselves. This technique must perform appropriate processes to hide the redirection from the adversary. The primary requirement of this technique is threat detection. A detection mechanism must first be performed to identify the malicious traffic, which can be done by one of these methods:
  - **Intrusion detection systems:** Some researchers use IDSs for the detection process. For example, La et al. (2016) proposed a signaling game model for a network that redirects malicious traffic to a honeypot. The malicious traffic is identified by deploying an IDS. The strategy that helps the network defender to decide which traffic must be redirected is obtained using Bayesian equilibrium in this game. Selvaraj et al. (2016) used a honeypot database in the network, and when an IDS detects malicious traffic, it will be redirected to that fake database for securing the stored data. Furthermore, Park et al. (2019) used a dynamic honeypot, called DVNH, in a Software-Defined Network (SDN) to be the redirection destination of the malicious traffic, which is detected by an IDS.
  - **Other types of honeypots**: Some other researchers try to detect the malicious traffic by another type of honeypot and then redirect it. Among them, we can mention the work done by Ja'fari et al. (2021). They used the redirection technique to detect the loaders of Mirai botnet. They first identify the compromised hosts by placing honeypot decoys in the network, then trace them to find the loaders, and finally redirect the loader traffic to a honeypot system to waste the adversary's time. The work by Biedermann et al. (2012) is also in this field. They used a cloud honeypot to detect brute force and dictionary scanning attacks and then clone a virtual honeypot on the attack target machine for redirection.
  - **Other detection methods:** Finally, we can see some researches that tries to detect malicious traffic using other methods. For example, Sardana and Joshi (2009) proposed a network architecture to mitigate DoS attacks by redirecting flooding flows to the honeypots. Flooding flows are detected by checking the traffic entropy. The redirection maintains the network quality of service for legitimate users. Tian et al. (2015) proposed a USB honeypot that can redirect the data sent from a malicious USB device to a honeypot. Detecting the anomalous USB device is performed by the end-users in this research.
- **Improving the performance:** The redirection process can also be used for increasing the performance of honeypots. For example, Wang and Wu (2019) designed a system, in which the powerful and weak attacks are redirected to high-interaction and low-interaction honeypots, respectively. This technique can help the developers create scalable honeynets and reduce their cost. Fan and Fernández (2017) performed a similar technique to filter more interesting

attack scenarios and send them to the honeypot. In this work, the detection mechanism is performed by the Snort IDS.

WT is the most appropriate metric for measuring this technique's deception power. When the adversary spends so much of his/her time communicating with these honeypots, we can state that the honeypot is correctly performing its task. Moreover, Bedi et al. (2011) proposed a two-player game model, in which the network defender tries to mitigate DDoS attacks by redirecting the adversary's traffic to a honeypot system. This model helps the defender find appropriate parameters to be considered for redirection.

4. Deception in honeynets

Cyber deception techniques to improve honeypot performance encompass innovative strategies applied within cybersecurity to augment the efficacy and capabilities of honeypots. Honeypots, which emulate vulnerable systems or services, are meticulously crafted to attract and mislead potential attackers, diverting their focus from genuine production systems. These deception techniques involve the development of more intricate and compelling honeypot environments. The aim is to not only lure attackers but also to study their behaviors, methodologies, and motives comprehensively. This deeper understanding empowers cybersecurity professionals with valuable insights to refine defensive strategies and enhance threat mitigation. Integrating advanced tactics that foster engagement, prolong interaction, and collect meaningful intelligence is central to the concept. Such techniques can encompass behavior mimicry, where honeypots imitate the actions of authentic users or systems, misleading attackers and generating invaluable data on their approaches. Another technique involves dynamic service emulation, whereby honeypots dynamically simulate various services, rendering the environment more realistic and intricate. This complexity challenges attackers to differentiate honeypots from actual systems. Research contributions in cyber deception have significantly enriched the landscape of honeypot methodologies (Srinivasa et al., 2022). Many studies have proposed diverse strategies, from incorporating deceptive elements into network architectures to developing advanced interaction models. Moreover, investigations into psychological aspects of attacker engagement, such as cognitive biases, have inspired novel approaches to honeypot design. Using game theory principles to optimize deception and engagement has also emerged as a promising avenue of research. These contributions underscore cyber deception techniques' versatility and evolving nature and their pivotal role in refining honeypot performance. By harnessing these innovative methods, cybersecurity professionals aim to bolster the accuracy of threat detection, enhance attacker profiling, and optimize incident response strategies. Ultimately, the synergy between the study of deception and the implementation of cutting-edge techniques furthers the efficacy of honeypots, safeguarding digital assets and fortifying organizations against an evolving landscape of cyber threats (de Nobrega, 2023).

- Innovative Strategies for Enhanced Honeypot Performance
- Diverting Attackers with Simulated Honeypot Systems
- Crafting Convincing and Engaging Honeypot Environments
- Deception Methods: Insights into Attacker Behavior
- Behavior Mimicry: Unveiling Attacker Approaches
- Authenticity and Challenge: Dynamic Service Emulation
- Enriching Honeypot Methodologies through Research Contributions
- Psychology-Driven Design: Advanced Interaction Models
- Strategic Engagement: Applying Game Theory to Honeypots
- Precision Defense: Innovations Enhancing Threat Detection
- From Study to Action: Strengthening Honeypot Effectiveness The subsequent content presents detailed descriptions for each key point covered earlier. These descriptions provide insights into various strategies that contribute to the enhancement of honeypot performance and cybersecurity effectiveness. The discussions encompass a range of innovative techniques, including methods to divert attackers using simulated honeypot systems, the art of crafting convincing and engaging honeypot environments, the use of deception to gain insights into attacker behavior, and the concept of behavior mimicry to reveal attacker approaches. Additionally, the content explores strategies such as dynamic service emulation that introduces authenticity and challenge, enriching honeypot methodologies through research contributions, employing psychology-driven design, applying game theory for strategic engagement, and leveraging innovations for precision defense. The insights shared in each description collectively contribute to a comprehensive understanding of how these approaches strengthen cybersecurity and fortify organizations against evolving threats (Shin and Lowry, 2020).

- Cyber deception techniques to improve honeypot performance involve innovative strategies in cybersecurity for enhancing honeypot effectiveness: Developing cyber deception techniques has emerged as a critical approach to bolstering honeypot performance in the ever-evolving cybersecurity landscape. By leveraging a blend of psychological manipulation, technical innovation, and strategic design, these techniques aim to outmaneuver malicious actors by providing them with seemingly authentic targets. The core objective is to create honeypots that attract attackers and actively deceive and engage them. This involves crafting environments that mirror real systems while embedding subtle inconsistencies that draw attackers further into the deception. As cybersecurity continually faces new challenges, implementing cyber deception techniques offers a dynamic and responsive approach to enhancing honeypot efficacy (Almeshekah et al., 2013; Zhang and Thing, 2021).
- Honeypots are simulated systems that divert attackers from real production systems by attracting and misleading them: Honeypots are one of the ingenious creations in cybersecurity, acting as a tactical diversion to divert attackers from actual production systems. With a deceptively genuine facade, honeypots tempt attackers to interact, providing defenders with a controlled environment to closely monitor and analyze attacker behavior. This diversionary approach is a valuable early warning system, allowing security professionals to detect potential threats before they can breach critical assets. By exploiting attackers' curiosity and desire for vulnerable targets, honeypots are pivotal in gathering intelligence that informs more effective cybersecurity strategies.

These techniques create more convincing and engaging honeypot environments, aiming to attract attackers and gather insights into their tactics: Creating convincing and engaging honeypot environments requires a delicate balance between authenticity and manipulation. Cybersecurity experts strive to create environments that draw attackers in and foster sustained engagement by meticulously mimicking legitimate systems' appearance, vulnerabilities, and interactions. This engagement isn't merely deceiving attackers; it is about crafting an environment that encouraging them to reveal their tactics, preferences, and strategies. This approach grants defenders an unprecedented opportunity to study attackers' decision-making processes, toolkit preferences, and evolving techniques. As attackers interact with honeypots, defenders gain insights that fuel the development of more robust and agile cybersecurity measures (Ackerman, 2020).

- Deception methods not only divert attackers but also enable in-depth study of their behaviors, methodologies, and motives: Deception methods serve as a dual-edged sword in the realm of cybersecurity. While they effectively redirect malicious actors away from valuable assets, their true value lies in the comprehensive insights they provide into the world of cyber adversaries. The information collected from honeypot interactions offers a unique window into attacker behaviors, methodologies, and motivations. This deeper understanding allows cybersecurity researchers to anticipate attackers' next moves, bolster defense mechanisms, and fine-tune incident response strategies (Marble et al., 2015). By carefully analyzing attacker engagements within honeypot environments, security professionals gain a clearer picture of

the evolving threat landscape and the tactics employed by malicious entities.

• Behavior mimicry is a technique where honeypots imitate actions of real users or systems, generating valuable data on attacker approaches (Shi et al., 2012): The technique of behavior mimicry introduces an element of finesse into the realm of honeypots. Cybersecurity experts capitalize on attackers' assumptions and habits by meticulously crafting interactions that mirror the actions of authentic users or systems. Attackers, drawn by the illusion of interacting with genuine assets, engage with honeypots in ways that parallel their typical strategies. This approach offers an unparalleled opportunity to capture data on attacker approaches, tactics, and decision-making patterns. As the attackers unknowingly navigate the simulated environment, defenders gain critical insights into the inner workings of cyber criminals, helping shape proactive cybersecurity strategies.

• Dynamic service emulation involves dynamically simulating various services to create a more authentic and challenging environment for attackers (Badr et al., 2015): Dynamic service emulation takes the concept of honeypots further by incorporating real-time adaptability. Traditional honeypots may offer static environments, but dynamic service emulation mimics actual systems' fluidity. By dynamically altering the services offered, cybersecurity experts challenge attackers to differentiate between real and simulated offerings, making their interactions more authentic and challenging. This approach increases the complexity of the honeypot environment, attracting more sophisticated attackers enticed by the nuanced engagement opportunities presented. As attackers grapple with the authenticity of the environment, defenders gain a clearer understanding of attackers' capabilities and intentions.

• Research contributions in cyber deception have enriched honeypot methodologies, proposing strategies like integrating deceptive elements into network architectures (Steingartner et al., 2021): Cyber deception has seen a surge of innovative research contributions that enrich the methodologies surrounding honeypots. Researchers have explored diverse avenues, from embedding deceptive elements directly into the architecture of networks to devising intricate interaction models that closely mirror real-world scenarios. These contributions represent a concerted effort to elevate honeypot efficacy beyond simple diversionary tactics. By infusing deception into the very fabric of network design, researchers have paved the way for a more comprehensive and strategic approach to cybersecurity. These advances ensure that deception techniques align with evolving attacker tactics, fortifying organizations against an increasingly complex threat landscape.

• Advanced interaction models and psychological aspects like cognitive biases have inspired new approaches to honeypot design: Cybersecurity researchers have delved into the realm of advanced interaction models, harnessing insights from cognitive psychology to refine honeypot design. These models capitalize on psychological phenomena, such as cognitive biases, to create engagements that resonate with attackers' decision-making processes. By exploiting these biases, cybersecurity experts tailor honeypot interactions that mirror real-world scenarios, prolonging attacker engagements and increasing the likelihood of extracting valuable data. This approach aligns with the broader trend of human-centric cybersecurity, recognizing that understanding the psychology of attackers is instrumental in shaping effective defense strategies (Faveri, 2022).

• Applying game theory principles to honeypots optimizes deception and engagement, reflecting the evolving nature of cyber deception techniques: Game theory principles, renowned for their application in strategic decision-making, find a natural fit in honeypots. By applying these principles, cybersecurity experts strategically balance the elements of deception and engagement within honeypot interactions. This delicate equilibrium ensures that honeypots are adaptive and responsive to evolving attacker strategies. As attackers modify their approaches, game theory principles guide the adjustments to honeypot tactics, optimizing deception and engagement dynamics. This dynamic approach reflects the evolving nature of cyber deception techniques, where adaptability is key to staying ahead of attackers (Zhu et al., 2021; Pawlick et al., 2021).

• These innovative methods enhance threat detection accuracy, improve attacker profiling, and optimize incident response strategies in cybersecurity: Integrating these innovative techniques ripples the entire cybersecurity landscape, yielding multifaceted benefits. Firstly, the heightened engagement from advanced techniques translates to more accurate threat detection. The extended interactions enable defenders to gather a more comprehensive dataset for analysis, leading to improved attacker profiling. This profiling, in turn, refines incident response strategies by providing a deeper understanding of attacker motivations, strategies, and potential targets. This integrated approach ensures cybersecurity measures are grounded in real-world attacker behaviors, resulting in more effective and targeted defense strategies (Althonayan and Andronache, 2019).

• The synergy between studying deception and implementing cutting-edge techniques strengthens honeypot effectiveness, fortifying organizations against evolving cyber threats: The synergy between academic research on deception and the practical application of cutting-edge techniques offers a holistic and dynamic defense mechanism. This collaboration equips organizations with honeypots adaptable to emerging threats and strategically designed to deceive and deter adversaries. As the threat landscape continually evolves, this fortified approach ensures a resilient defense, safeguarding organizations against the ever-shifting tactics of cyber adversaries. This collaborative synergy represents the frontline of defense, harnessing the best research insights and practical innovation to create an intelligent and anticipatory security posture.

5. A general mathematical model for analyzing honeynets

A honeynet is a deceptive network of decoys that places several honeypots in different network locations to take the advantage of the collaboration between these honeypots. This community of honeypots is more effective than using a single decoy in a network. It is worth noting that the honeypots in a honeynet are not always in different physical machines. Virtual honeypots on the same machine can also construct the honeynet. The point is that the adversary thinks that they are different systems.

In addition to making single honeypots powerful in deceiving adversaries, the network of honeypots (i.e., honeynet) must also be effective to reach the goal of deception. For example, the number of single honeypots in a honeynet and their placement significantly impact the total deception power.

Most of the researches in this field used gaming models to present a honeynet in which the network defender and the adversary are the main players. Before presenting the deception techniques in this section, we suggest a general representation of honeynets, which can match the current works in this field. All the game models mentioned in this section can be comparable with this new representation. The notations used in this representation are summarized in Table 4. A honeynet, $\mathcal{H}$, can be written as $\mathcal{H} = \{\mathcal{N}, \mathcal{S}, \mathcal{C}, \mathcal{B}\}$, where $\mathcal{N}$ is the network characteristics, $\mathcal{S}$ is the detail of the players strategies, $\mathcal{C}$ is the set of costs that the players may pay, and $\mathcal{B}$ is the set of benefits that the players may obtain.

$\mathcal{N}$ can be written as $\mathcal{N} = \{h, r, z, D\}$, where h, r, z are the number of honeypots, the number of real hosts that are vulnerable to cyber attacks, and the number of other safe hosts in the network, respectively. D is a list of hosts' degrees, where d_i is the i^{th} host degree and d_{max} is the maximum degree value among all the hosts. The number of links connected to a host is considered its degree.

We have $\mathcal{S} = \{ma, ah, sr_h, sr_r, ar_a, sr'_h, sr'_r, ar'_a, oh\}$, where ma is the maximum number of attack attempts that the adversary can perform and ah is the adversary's acceptable number of connections to the honeypots. sr_h and sr_r are the service rates of honeypots and real hosts. ar_a is the adversary's attack rate, and the symbols with a quote are the reduced factor of that symbol after applying the optimal strategy. For

Table 4
The list of notations used for representing a honeynet.

Notation	Description
$\mathcal{H}$	The honeynet
$\mathcal{N}$	The network characteristics
h	The number of honeypots
r	The number of vulnerable real hosts
z	The number of safe real hosts
D	The list of hosts degrees
d_i	The degree of the i^{th} hosts
d_{max}	The maximum hosts degree
$\mathcal{S}$	The players strategies
ma	The maximum number of adversary's attack attempts
ah	The adversary's acceptable number of connections to the honeypots
sr_h	The honeypots service rate
sr_r	the real hosts service rate
ar_a	The adversary's attack rate
sr'_h	The reduced factor of the honeypots service rate
sr'_r	The reduced factor of the real hosts service rate
ar'_a	The reduced factor of the adversary's attack rate
oh	The optimal number of honeypots in the defender's best strategy
$\mathcal{C}$	The set of players costs
pc_h	The adversary's cost of probing a honeypot
pc_r	The adversary's cost of probing a real host
ac_h	The adversary's cost of attacking a honeypot
ac_r	The adversary's cost of attacking a real host
pc'_h	The defender's cost when the adversary probes a honeypot
pc'_r	The defender's cost when the adversary probes a real host
ac'_h	The defender's cost when the adversary attacks a honeypot
ac'_r	The defender's cost when the adversary attacks a real host
cc	The adversary's cost of being caught by the honeypots
dc	The defender's cost of deploying a honeypot
rc	the defender's cost of responding to the adversary's attempts
mc	The adversary's maximum acceptable cost
$\mathcal{B}$	The set of players benefits
ab_h	The adversary's benefit of successfully attacking a honeypot
ab_r	The adversary's benefit of successfully attacking a real host

example, assume that the service rate of a honeypot is 100%, and in the optimal strategy that honeypot must have a service rate of 50%, the reduced factor of sr_h is 0.5 (i.e., the value of sr'_h is 0.5). oh is the optimal number of required honeypots in the defender's best strategies.

The set of costs for each operation can be written as $C = \{pc_h, pc_r, ac_h, ac_r, pc'_h, pc'_r, ac'_h, ac'_r, cc, dc, rc, mc\}$, where pc_h and pc_r are the adversary's cost of probing a honeypot and a real host, respectively. ac_h and ac_r are the adversary's costs of attacking a honeypot and a real host, respectively. The symbols with a quote are the same cost for the defender. cc is the adversary's cost of being caught by the honeypots, dc is the defender's cost of deploying a honeypot in the network, and rc is the defender's cost of responding to the adversary's attempts. Finally, mc is the maximum acceptable attacking cost for the adversary.

For the benefits set, we have $\mathcal{B} = \{ab_h, ab_r\}$, where ab_h and ab_r are the adversary's benefits of successfully attacking a honeypot and a real host, respectively. It is worth noting that some adversaries can compromise the honeypots to control them, and this is considered as a successful attack to a honeypot.

At the following, we present deception techniques that are used to improve honeynets' performance. We also mention the suggested strategies in each research represented with our model, $\mathcal{H}$. We have also simulated comparable works to analyze their performance. A summarization of the related researches in this field is presented in Table 5 (note that some of the names for the researches models are assigned by ourselves).

5.1. Optimizing the honeypots

One of the most significant network parameters in honeynets is the number of deployed honeypots. Placing a few honeypots in the network is insufficient to lure the adversary and protect the entire network. On the other hand, using too many honeypots in the network is costly, and in some cases, it may warn the adversary about the deception mechanism that it is facing. Therefore, finding the optimal number of honeypots for a network is necessary.

One of the first researches in optimizing the number of honeypots was performed by Rowe et al. (2007). This research proposed a mathematical model to calculate the adversary's cost and benefit in attacking a honeynet. If the adversary spends more than the cost threshold on a system, it will ignore that system and try to attack another system. Since this model considers the adversary's tolerance, we call it TBM (short for Tolerance-Based Model). Using this model, the network defender can set an appropriate number of honeypots to increase the attacking cost than the adversary's expected cost. The upper bound of the accepted attacking cost of a single system for the adversary (mc) can be calculated by Equation (1).

$$mc = \frac{(pc_r + ac_r)(ab_r(h + r + z) - h(ab_r + ab_h))}{ab_r(h + r + z + ac_h h)} \tag{1}$$

According to Equation (1), we can conclude that the optimal number of honeypots (oh) can be calculated by Equation (2).

$$oh = \frac{ab_r(r + z)(pc_r + ac_r - mc)}{ab_r mc(ac_h + 1) + ab_h(pc_r + ac_r)} \tag{2}$$

Although the previous model can be used to find the appropriate number of honeypots, the network defender must have information about the adversary's perceived cost and benefit. Hence, another model is proposed by Crouse (2012) to find the optimal number of honeypots without the adversary's perceived parameters. A honeynet is modeled as a URN that contains three types of beads with different colors. We call this model URN. The h honeypots, the r vulnerable hosts, and the other z network elements are shown as red, green, and blue, respectively. A bead is removed from the URN when the adversary attacks a host. In this model, the adversary attempts to attack ma hosts, which leads to eliminating ma beads. This model assumes that the adversary is successful if it can launch an attack against at least one of the vulnerable hosts. Hence, in the URN model, the adversary reaches the goal if at least one of the beads from the ma removed beads is green. We can calculate the probability of a successful attack by this assumption Equation (3).

$$Pr(\text{successful attack}) = \sum_{x=1}^{ma-1} \frac{\binom{r}{x}\binom{z}{ma-x}}{\binom{h+r+z}{ma}} \tag{3}$$

The network defender can calculate the number of required honeypots Equation (3), considering the network situations to quantify the threshold for the adversary's successful attack rate. A simple threshold is to set the maximum success probability as 0.5. In this situation, the number of failed attacks exceeds that of successful ones. Hence, the adversary may avoid attacking the network.

Some adversaries employ intelligent attack strategies in which connecting to a few honeypots cannot reveal their identity. Hence, they accept communicating with a specific number of honeypots toward reaching their goals. As a result, Crouse et al. (2015) proposed a similar URN model that also considers an acceptable number of connections to the honeypots (ah). We call this model URNt (short for URN with threshold). In such conditions, the probability of a successful attack can be calculated as Equation (4).

$$Pr(\text{successful attack}) = \sum_{l=0}^{ah} \sum_{x=1}^{ma-1} \frac{\binom{r}{x}\binom{h}{l}\binom{z}{ma-(x+l)}}{\binom{h+r+z}{ma}} \tag{4}$$

Therefore, the network defender can make the best decision to choose the appropriate deception method by changing the values of the variables in Equation (4) and examining the probability of the adversary's victory.

The previous URN models, do not consider the probing process that the adversary performs before launching the attacks. Therefore, Fraunholz and Schotten (2018b) proposed a game model that also models the

Table 5
Comparing the researches in the field of deception in honeynets.

Research	Purpose	Key Characteristic	Optimal Strategy for the Defender	Model Type
TBM (Rowe et al., 2007)	Optimizing	Considering the adversary's tolerance	Calculate the maximum expected cost for the adversary and try to make the attack cost higher than that. Set the number of honeypots according to Equation (2).	Mathematical model
URN (Crouse, 2012)	Optimizing	Considering the safe hosts in the model	Calculate the probability of a successful attack and try to make it lower. Set the number of honeypots according to Equation (3).	Mathematical model
URNt (Crouse et al., 2015)	Optimizing	Considering the safe hosts and a threshold for connecting to a honeypot	Calculate the probability of a successful attack and try to make it lower. Set the number of honeypots according to Equation (4).	Mathematical model
GBO (Fraunholz and Schotten, 2018b)	Optimizing	Considering the number of probes before the attack	Make a situation that the adversary prefers not to attack. Calculate the number of honeypots according to Equation (5).	General two-player game
HSGp (Píbil et al., 2012)	Diversifying	Assigning each host a numerical importance value	Model the network with the suggested game and find the optimal strategy using Nash equilibrium.	Zero-sum game
HSG (Kiekintveld et al., 2015)	Diversifying	Using network attack graph to find the hosts' importance value	Model the network with the suggested game and find the optimal strategy using Nash equilibrium.	Zero-sum game
DHG (Durkota et al., 2015a)	Diversifying	Assigning limited importance values to the hosts	Find the adversary's optimal strategy and try to make it unreachable with Stackelberg equilibrium.	General two-player game
DHGu (Durkota et al., 2015b)	Diversifying	Assuming that the adversary does not know the honeypot types to be more realistic	Calculate the upper bound for the adversary's cost and approximately solve the game using Stackelberg equilibrium.	General-sum game
DHD (Sarr et al., 2020)	Diversifying	Considering honeypot detection techniques	Diversify the network to reduce the probability of honeypots detection when one of them is detected	Zero-sum game
POSG (Anwar et al., 2019)	Locating	Considering unknown attack graph safety level for the defender	Use approximation methods or POMDP algorithm to solve the game and find the optimal strategy.	Zero-sum game
POSGm (Anwar et al., 2020)	Locating	Letting the defender to place multiple honeypots on the attack graph	Check all strategies and find the optimal one according to the reward function by a progressive algorithm	Zero-sum game
DD (Cai et al., 2009)	Dynamizing	Assuming contiguously located honeypots	Tell lie as much as possible	General two-player game
SGM (Carroll and Grosu, 2011)	Dynamizing	Considering the attacks with and without probing	In the conditions mentioned in Equation (6) to Equation (8), show the probed host as a real host and otherwise, show it as a honeypot	Signaling game
SGMd (Çeker et al., 2016)	Dynamizing	Focusing on the denial of service attacks	Make the real hosts tell the truth in the conditions mentioned in Equation (9) to Equation (14). Make the honeypots tell the truth in the conditions mentioned in Equation (11) to Equation (16). In other situations, tell the lie.	Signaling game
HDG (Garg and Grosu, 2007)	Dynamizing	Considering multiple probes for the adversary	Respond so that the probability of getting the truth and lie are equal.	Extensive form game
CSG (Pawlick and Zhu, 2015)	Dynamizing	Considering adversary's obtained evidence	Find the optimal strategy by perfect Bayesian Nash equilibrium	Signaling game
SGE (Pawlick et al., 2018)	Dynamizing	Considering adversary's obtained evidence based on hosts activities	In dominant state, make all the real hosts and honeypots seem the same. In a heavy state, make all the honeypots seem like the real hosts. In the middle state, make half of the honeypots act like real hosts, and only a few real hosts act like honeypots.	Signaling game
eHDG (Bilinski et al., 2018)	Dynamizing	Assuming that both the adversaries and the legitimate users may probe the network to be more realistic	Lying is necessary for the defender.	General two-player game
BRL (Limouchi and Mahgoub, 2021)	Dynamizing	Using machine learning based on the malicious neighbors to find the optimal strategy	Acting as a honeypot when more than two malicious neighbors are detected	Reinforcement learning
IoTCandyJar (Luo et al., 2017)	Dynamizing	Using machine learning to create an intelligent-interaction honeypot	The optimal strategy is found based on the trained model	Machine learning
MDRL (Huang and Zhu, 2019)	Dynamizing	Using machine learning for finding the optimal strategy, especially for choosing to act as a low or high-interaction honeypot	The optimal strategy is found based on the trained model	Reinforcement learning
DTG (Ren and Zhang, 2020)	Shaping	Assigning different tasks for the nodes with different topological characteristics	Use higher exponent for scale-free networks.	Differential game
VTG (Ren et al., 2021)	Shaping	Considering various network topologies	Keep the greatest hosts' degrees low.	Mathematical model

probing process and it is solved based on the Stackelberg competition, in which the adversary and the defender try to utilize the best strategy after the previous player's strategy. As this model uses game theory concepts for optimizing the number of honeypots, we call it GBO (short for Game-Base Optimizer). This game model tries to find the optimal number of honeypots that are required to be deployed in the network. Two scenarios are defined in this model. In the first scenario, the adversary do not probe the hosts before attacking them. Hence, the model suggests setting the number of honeypots so that the attacking payoff is equal to the non-attacking payoff. In this situation, the adversary is forced not to attack the hosts. According to this condition, the minimum number of honeypots (oh) can be calculated as Equation (5).

$$oh = \frac{r \times ab_r}{dc} \tag{5}$$

In the second scenario, the probing process is also considered. A linear equation is presented to find the payoff of each strategy, and then, a heuristic algorithm is proposed to find the best strategy.

Algorithm 1 The adversary's attack process in the simulations for comparing the researches conducted for optimizing the honeypots.

```
honeypots ← the list of honeypot nodes
hosts ← the list of real host nodes
network ← honeypots + hosts
compromised ← an empty list
checked ← an empty list
payed ← 0
accepted_cost = 20 × host_cost + 4 × honeypot_cost
while True do
    if size(compromised) = 20 then
        log a successful attack
        break
    if size(checked) = size(network) then
        log an unsuccessful attack
        break
    if payed > accepted_cost then
        log an unsuccessful attack
        break
    target ← a random node from network − checked
    add target to checked
    if target ∈ honeypots then
        payed ← payed + honeypot_cost
    else
        payed ← payed + host_cost
        add target to compromised
```

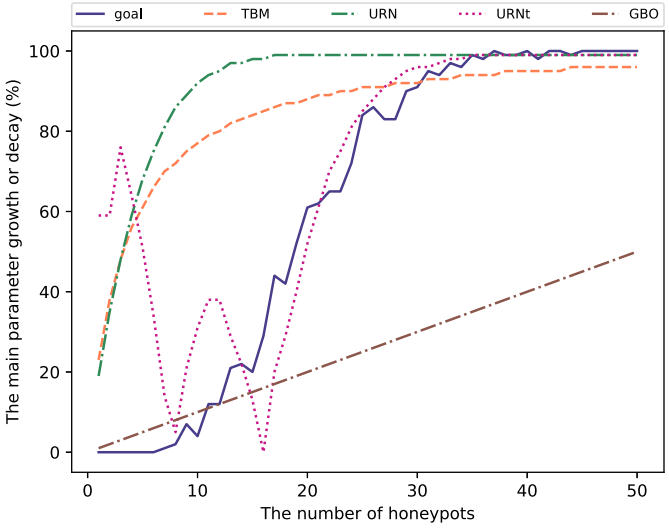


Fig. 11. Comparing the researches about optimizing the number of honeypots.

To compare the performance of the researches mentioned in this section (i.e., TBM, URN, URNt, and GBO), we simulated several random networks with different characteristics in Python, and apply the optimal metric suggested by each model. Each network has a total number of 100 nodes, and the number of honeypots varies from zero to 50 in each scenario. The adversary probes the network and aims to compromise at least 20 real hosts in the network. Connecting to more than 4 honeypots leads to the adversary's failure. The adversary's attacking scenario details are presented in Algorithm 1.

The results are shown in Fig. 11. The changes in the number of successful attacks, which is indicated by "success", and the changes in the four models' metrics, are shown in this figure. The growth and decay of the metrics compared with the adversary's success metric explains that using Equation (4), presented by URNt model, can get better results than the other three models in our scenarios. It seems that Equation (4) gets better results because of using the most important metrics, especially the maximum number of acceptable connections to the honeypots (i.e., ah).

5.2. Diversifying the honeypots

The type of honeypots located in the network is another major parameter affecting the honeynet performance. There are different types of honeypots with various capabilities. Hence, the network defender must place appropriate types according to their deployment cost. Moreover, this diversity helps the defender prevent more honeypot detection attacks. In fact, if the adversary finds a way to detect a honeypot of type t, the other type t honeypots can be detected similarly. But, if the honeypots are of different types, detecting one of the honeypots may not lead to simply detecting all of them.

Píbil et al. (2012) proposed a zero-sum game model, called HSG (short for Honeypot Selection Game) for a honeynet with different types of honeypots. This game assigns a numerical value to each real host and honeypot, indicating its importance in the network. For example, a database server is one of the critical assets in the network and it is likely to be the target of an adversary. The important values for the honeypots are fake as they pretend to be that much important. In HSG, the network defender and the adversary are the game's players. The adversary aims to attack a real host. At the same time, the defender optimally deploys a fixed number of honeypots with different types (different importance values) to increase the probability of a honeypot being attacked. In HSG, the adversary is not able to probe the hosts before attacking them. Hence, another model called HSGp is proposed in the same research to support the probing process. In HSGp, the adversary's resources are limited and it can probe only a specific number of hosts. The information obtained by these probes is not always valid. This research suggests finding the optimal strategy for the defender by Nash equilibrium concepts (Kreps, 1989).

Since the importance of the network hosts is not simply a number, Kiekintveld et al. (2015) used the HSG model with important values calculated from the network attack graph. The attack graph can specify host vulnerabilities, and presenting the most probable attack paths can assign an important value to each host.

The important value of the hosts in a network is selected from a specific list. Therefore, only specific numbers are allowed to be assigned to the honeypots. As a result, Durkota et al. (2015a) proposed a game model for presenting a honeynet with different but limited honeypot types. We name this model DHG (short for Diversifying Honeynet Game). The adversary and the network defender are the players of this game. The defender can place a specific number of honeypots in the network, but their types are custom. The adversary does not know which host is a honeypot. Hence, it creates the attack graph of the honeynet and analyzed it to find the optimal attack path. According to the optimal strategy of the adversary, the best defensive strategy can be calculated by the Stackelberg equilibrium. In this game, the adversary pays a specific cost when it attacks a honeypot, and on the other hand, it gets a reward for successfully attacking a real host.

Since DHG assumes that the adversary knows the types of honeypots, another model is proposed by Durkota et al. (2015b) for a similar scenario, but with assuming that the adversary is unaware of the honeypot types. We call this model as DHGu (short for DHG with an unaware adversary). This model presents a network of different honeypot types with a general-sum game. The defender and the adversary are the two players of this game. The network defender performs the first move to place different types of honeypots in the network optimally. Deploying these honeypots may have different costs and bring different security levels. The adversary knows the number of honeypots, but does not have any information about their type and location. However, it selects a host to compromise by calculating the probability of accessing a honeypot and the cost of its successful attack. This research proposes approximate solutions to find the best strategies for each player, and also presents a linear equation to calculate the upper bound of utility functions in this game.

To analyze the honeypot detection attacks, Sarr et al. (2020) proposed a zero-sum game model, in which the defender attempts to reduce

the chance of successful honeypot detection attacks by increasing the cost of detecting all honeypot types. We call this model as DHD (short for Diversifying to mitigate Honeypot Detection). The defender has to place a specific number of honeypots in the network. However, it is allowed to choose custom honeypot types. The defender pays a specific cost for deploying each type of honeypots. On the other hand, the adversary obtains a reward by detecting a honeypot, but, pays the same deployment cost of a type t honeypot as the detection cost. For example, if the defender utilizes two types of honeypots, low-interaction and high-interaction, the deployment cost of the first and second types are c_1 and c_2, respectively, and the detection must also pay c_1 and c_2 to detect the first and the second types, respectively. The point in this game is if the adversary pays c_1 to detect a honeypot of the second type, he will not be successful. This research suggested not deploying the same type of honeypots and diversifying them randomly in the network to decrease the chance of honeypot detection.

5.3. Locating the honeypots

In addition to the number of honeypots, their location is important in deceiving the adversary and wasting time. Two honeynets with the same number of honeypots may have different defensive performances according to the placement strategy which is deployed for locating the honeypots. The attack graph of a network can be used to find the appropriate location for the honeypots. An attack graph is a directed graph that represents the beginning and the ending states of different intrusions in the network. The edges in this graph show the process of exploiting the vulnerabilities. For example, if the adversary is able to compromise host h_1 only after compromising host h_2, in the attack graph we have h_1 and h_2 as two nodes and an edge exists from h_2 to h_1. The honeypots must be located between two nodes of this graph that are connected with an edge, in such a way that reaching the final state (i.e., the adversary's goal) through them is costless for the adversary. Hence, it would communicate with the honeypots and will be deviated from the attack target. In other words, we try to show the network more vulnerable with the honeypots in appropriate locations. Moreover, placing the honeypots in proper locations can help us trace the adversary. Communicating with a honeypot shows that the adversary has exploited the prerequisite nodes to reach the current one.

Anwar et al. (2019) modeled the problem of placing the honeypots in a network as a zero-sum stochastic game between the defender and the adversary. The game model is called POSG. Both the players know the attack graph in POSG, however, the adversary does not know which network node is a honeypot, and the defender is not aware of which vulnerability has been exploited by the adversary. In each step, the defender can choose to place a single honeypot on a specific edge of the network attack graph or do nothing, and the adversary selects a host to exploit. Exploiting a honeypot costs high for the adversary, and the defender gets a reward. But, if the adversary probes a real host, it gets a reward and the defender must pay a specific cost. The POSG researchers did not mention a specific strategy for the players, and they suggest using approximate methods or POMDP algorithm, which is used for similar game models, to find the optimal strategy.

Since the defender can place only a single honeypot in each step of POSG, Anwar et al. (2020) proposed another zero-sum game model based on the network attack graph, in which the defender can place as many honeypots as it wants in each step. We call this model POSGm (short for POSG with multiple honeypots). The defender must pay a specific cost for locating each honeypot in the network, and gets a specific reward if it could trace the adversary. On the other hand, the cost for the adversary is to be detected and the reward is to exploit a real host vulnerability. In this game model, first, a linear equation is suggested to calculate the reward function of the players, a progressive algorithm is proposed to check all possible game states in future steps and find the best strategies.

5.4. Dynamizing the honeypots

One of the deception techniques that can be utilized in honeynets is changing the behavior of real hosts or honeypots and responding to the adversary dynamically. In this case, when an adversary probes one of the hosts, the network defender decides whether to show that host as a real host or as a honeypot. This technique increases the adversary's uncertainty about the host types. For example, if the adversary finds a host with many open ports, which shows the clue of being a honeypot, it would not be sure of the type of that host. That host may be really a honeypot or it is a real host that pretends to be a honeypot. For this reason, making the honeynet dynamic can lead to higher deception performance. But, there is a trade-off between the deception level obtained by lying and the cost of configuring a related mechanism. Hence, this dynamicity must be created with appropriate boundaries.

Cai et al. (2009) proposed a two-player game model, in which the honeypots are contiguously located in the network's address space. The network defender and the adversary are the players of this game. The adversary probes its desired hosts. If that host is a real one, it gets a normal response. But, if the adversary probes a honeypot, the defender decides to lie or tell truth about the identity of that honeypot. The number of lies in this model is limited. Hence, the defender must adopt an intelligent strategy to lure the adversary. The adversary tries to find the block of honeypots with the lowest possible number of probes, and the defender aims to increase the number of these probes. This research suggests the defenders to use a strategy, called Delay-Delay (DD), in which the honeypots always tell lies until their limitation is exceeded. As the optimal strategy in this model is called DD, we also name the model as DD.

The honeypots are not always located contiguously within the address space. They may be in random addresses. Hence, Carroll and Grosu (2011) proposed another model based on signaling games, in which the network utilizes the honeypots in custom places. We name this model as SGM (short for Signaling Game Model). The network defender and the adversaries are the sender and the receiver in SGM, respectively. A specific number of honeypots are located in the network, but the defender can reply to the adversary's probes with different responses. In other words, if a host is a honeypot and the adversary probes it, the defender decides whether to respond to it as a real host or as a honeypot. The defender tries to conceal the identity of the honeypots and make the adversary attack them. On the other hand, the adversary aims to select a real host for attacking. When the adversary probes a host, the defender can respond with 'h' or 'r' to show the probed host is a honeypot or a real system, respectively. In this situation, the adversary has three choices: attacking that host, attacking the host after probing it, or finishing the game without attacking. SGM suggested that the optimal strategies are to always respond with 'h' or to always respond with 'r'. This research also states that the strategy of randomizing the responses is equivalent to these two strategies. In any of the conditions mentioned in Equation (6), Equation (7), and Equation (8), the optimal strategy is to always respond with 'r'. Moreover, the optimal strategy in other conditions is to always respond with 'h'.

$$\frac{h}{h+r+z} \le \frac{pc_h}{ac_r+cc} \wedge \frac{h}{h+r+z} \le \frac{ab_r-ac_r}{ab_r+ac_r} \tag{6}$$

$$\frac{h}{h+r+z} \ge \frac{pc_h}{ac_r+cc} \wedge \frac{h}{h+r+z} \le \frac{ab_r-ac_r-pc_h}{ab_r-ac_r} \tag{7}$$

$$\frac{h}{h+r+z} \ge \frac{ab_r-ac_r}{ab_r+cc} \wedge \frac{h}{h+r+z} \ge \frac{ab_r-ac_r-pc_h}{ab_r-ac_r} \tag{8}$$

Çeker et al. (2016) proposed a signaling game model for presenting a honeynet, which is nearly similar to SGM. However, this model focuses on preventing DoS attacks in a honeynet. The network defender is the sender in this model and tries to optimally configure the honeynet to still serve legitimate users under a DoS attack. Since this model is like SGM and is used for DoS attacks, we name it as SGMd (short for SGM in DoS attacks). The adversary is the receiver player and performs

one of these actions when communicating with a host: attacking that host, observing, or finishing the game without attacking. This research suggested the real hosts to tell the truth in the conditions mentioned in Equation (9) to Equation (14), and in other conditions, it is optimal to signal the lie. The honeypots also suggest telling the lie in the conditions mentioned in Equation (11) to Equation (16), and the truth in other conditions.

$$sr_r \geq \frac{ac'_r - rc}{sr'_r} \wedge sr_r \geq \frac{ab_r - rc}{sr'_h} \wedge ar_a \leq \frac{ac'_r}{ar'_a} \tag{9}$$

$$sr_r \leq \frac{rc}{sr'_r} \wedge sr_r \leq \frac{rc}{sr'_h} \wedge ar_a > \frac{ac'_r}{ar'_a} \tag{10}$$

$$\frac{rc}{sr'_h} \leq sr_r \leq \frac{rc}{sr'_r} \vee \frac{rc - ab_r}{sr'_h} \leq sr_r \leq \frac{rc + pc'_r - ac'_r}{sr'_r} \tag{11}$$

$$\frac{ac'_r - rc}{sr'_r} \leq sr_r \leq \frac{ab_r - rc}{sr'_h} \vee \frac{rc + ab_r}{sr'_h} \leq sr_r \leq \frac{rc + ac'_r - pc'_r}{sr'_r} \tag{12}$$

$$\frac{rc}{sr'_h} \leq sr_r \leq \frac{pc'_r}{sr'_r} \vee \frac{rc + ab_r}{sr'_h} \leq sr_r \leq \frac{rc + ac'_r}{sr'_r} \tag{13}$$

$$\frac{rc}{sr'_h} \leq sr_r \leq \frac{rc + pc'_r}{sr'_r} \vee \frac{rc + ab_r}{sr'_h} \leq sr_r \leq \frac{rc + ac'_r - pc'_r}{sr'_r} \tag{14}$$

$$sr_r > \frac{ac'_r + rc}{sr'_r} \wedge sr_r > \frac{ab_r + rc}{sr'_h} \wedge ar_a \leq \frac{ac'_r}{ar'_a} \tag{15}$$

$$sr_r > \frac{rc}{sr'_r} \wedge sr_r > \frac{rc}{sr'_h} \wedge ar_a > \frac{ac'_r}{ar'_a} \tag{16}$$

In SGM and SGMd, the adversary can only probe a single host in each step, while in reality, the adversary may probe several hosts before deciding to attack one of them. Hence, Garg and Grosu (2007) proposed another game model, called HDG, to better represent the honeynets in lying scenarios. HDG is an extensive form game with the network defender and the adversary as the first and second players, respectively. The players in HDG move alternatively until the adversary probes a specific number of hosts. At the final step of this game, the adversary decides to attack one of the hosts or not. HDG suggests that the defender to respond in such a way that the probability of getting a true or lie response is equal.

Some clever adversaries, seek for evidence to probabilistically check whether a response is a lie or a truth. For example, a honeypot may simulate mouse movements to act normally. However, the adversary can find out the fake movement with uncommon patterns. Two models are proposed by the researchers that consider the adversary's evidence. The first model is proposed by Pawlick and Zhu (2015). This model is based on cheap-talk signaling games, in which the network defender and the adversary are the sender and the receiver, respectively. We call this model as CSG (short for Cheap-talk Signaling Game). The adversary can find evidence about the deception used in the network, and after getting the message from the defender, decides whether to launch an attack. CSG suggests finding the optimal strategy using perfect Bayesian Nash equilibrium. However, an exact optimal strategy is not defined. Hence, Pawlick et al. (2018) proposed another game that models a honeynet with two types of lie and truth responses. This model is based on cheap-talk signaling games and supposes that the network defender and the adversary are the sender and the receiver in this game, respectively. The message exchanged between the sender and the receiver in this specific game is the activity level of a host. A host with a high activity level is probably a real host, and a low activity level is the main characteristic of a honeypot. Though, the defender can change the activity level to lure the adversary. Since this model is based on a Signaling Game with Evidence, we call it as SGE. The optimal strategy for the defender in SGE is to make the adversary's evidence useless. Therefore, this model considers different states for a honeynet and suggests the defender's optimal strategy in each state. The states and the strategies are as follows:

- Dominant state: In the case that the number of honeypots or the number of real hosts are extremely low (i.e., almost zero), the optimal strategy is to make all the systems (i.e., the honeypots and the real hosts) have similar activity level. In other words, all the systems must be at a high activity level or all of them must have a low activity level.
- Heavy state: When the number of honeypots is not near to zero but it is less than the real hosts, the optimal strategy is to keep the real hosts' activity level high and make the honeypots appear with a high activity level. In the case that the number of real hosts is not near to zero but it is less than the honeypots, the optimal strategy is the same.
- Middle state: If the number of honeypots and real hosts are nearly equal, the optimal strategy is to make half of the honeypots act active, and the other half be in a low activity level. Most of the real hosts must also be kept on high-level activity and the others on low-level activity.

The previously mentioned models in this section assumed that only the adversary probes the network. In real situations, some benign hosts also probe the network and communicate with others. Hence, Bilinski et al. (2018) extended HDG model and proposed a Bayesian game model, in which the first player is the network defender and the second player is a general node. The second player is an adversary with a specific probability and otherwise, it is a benign node. We call this model as eHDG (short for extended HDG). The defender in eHDG can place up to k hosts in the network, and the adversary wins the game if it can perform a successful attack against at least one real host. The main purpose of this game is to show that the lying strategy for the defender is necessary. Without this lying, the defender loses most of the time. The suggested strategy is to balance the number of lies between all the hosts.

Reinforcement learning is utilized by Limouchi and Mahgoub (2021) to optimally dynamize the honeypots in an IoT environment. Since this model uses Bayesian algorithms combined with reinforcement learning, we call this model as BRL (Bayesian Reinforcement Learning). A honeypot can act as a real or fake host in this research. When the honeypot has more than a threshold number of malicious neighbors, it must act as a honeypot. Otherwise, it can switch to a real host. This research shows that the threshold value of 2 leads to the optimal strategy for the defender.

One can consider the dynamizing process as changing the behavior of the honeypots based on their interaction level. Luo et al. (2017) called this honeypots as intelligent-interaction honeypots. In this research, the status of the devices in an IoT network is continuously monitored, and then a machine-learning model is trained to find the optimal strategy. The honeypot architecture proposed in this research, IoTCandyJar, decides based on the learning model to act as a low or high-interaction one. Huang and Zhu (2019) proposed a reinforcement learning model for effectively dynamizing a honeynet by changing the interaction level of the honeypots. The defender in this learning model can perform four different actions: ejecting the adversary's connection, recording all the adversary's information, acting as a low-interaction honeypot, and acting as a high-interaction honeypot. This model uses semi-Markov decision processes to find the optimal strategy. We call this model MDRL (Markov Decision-based Reinforcement Learning).

To compare the main models mentioned in this section (i.e., DD, SGM, HDG, and SGE), we simulated several scenarios in Python. We applied the optimal strategy of each of these four models. Each simulated network is in one of the states introduced in SGE: Dominant, Heavy, and Middle States. Each network contains 100 nodes, where 5, 20, and 50 nodes are honeypots in Dominant, Heavy, and Middle states, respectively. We have simulated four types of scenarios. In the first scenario, the adversary scans the network randomly (RandomScan), and the honeypots are randomly distributed in the network (RandomLocation). This scenario is called RS-RL. In the second scenario, the

Algorithm 2 The process of the simulated scenario for comparing the researches about dynamizing the honeypots.

```
network ← the list of the network nodes
checked ← an empty list
belief ← an empty list
for i from 1 to 40 do
    target ← a random node from network − checked
    add target to checked
    response ← probe(network, target)
    accept ← a random number between 0 and 100
    if accept < true_evidence_rate then
        if response = a real host then
            add target to belief
    else
        if response = a honeypot then
            add target to belief
if size(belief) > 0 then
    target ← a random node from belief
    if target is a real host then
        log a successful attack
    else
        log an unsuccessful attack
else
    log an unsuccessful attack
```

honeypots are sequentially distributed (SequentialLocation). So, we call this scenario as RS-SL. The third scenario uses the sequential scanning (SequentialScan) method with randomly distributed honeypots. So we call it SS-RL. Finally, the last scenario is called SS-SL, because it uses sequential scanning and the honeypots are located sequentially within the network space. The adversary probes up to 40 hosts, and the network defender can lie at most 30 times in our scenarios. Finally, the adversary selects a target among the probed hosts and launches an attack against it. If the target is a real host, the attack is successful, and otherwise, the adversary fails. The detailed process of the simulated scenarios is explained in Algorithm 2. The probe() function mentioned in Algorithm 2 is the target node type, which may be a lie or a truth, according to the dynamizing model.

The results of each scenario are shown in Fig. 12. We can see that on average, DD and SGM have higher performance than HDG and SGE in our scenarios. This may be attributed to the fact that the frequency of changes in the responses in DD and SGM is lower than HDG and SGE.

5.5. Shaping the honeynet

A honeynet topology is another factor to be considered in analyzing its performance. The connections between the hosts can have a significant influence on deception power. For example, in a network with full mesh topology, all the hosts are directly connected, and the propagation of malware in such a network is fast. Hence, the network defender must pay attention to the honeynet topology to get better performance.

The honeynet topology affects the single honeypot's abilities. Ren and Zhang (2020) proposed a differential game between the honeynet and the adversary to analyze this effect. We name this game model a Differential Topological Game (DTG). In this game, the adversary attempts to find the best rate of infecting the network hosts to launch a DDoS attack. On the other hand, the honeynet tries to reduce the propagation rate of the adversary's malware with the lowest cost. It is stated in this research that the degree of each honeypot can significantly influence its performance. For example, higher-degree honeypots can capture more attacks than lower-degree ones. However, the lower-degree honeypots are more appropriate for recovery processes. This research also suggested that a higher exponent is more efficient in preventing attacks for scale-free networks, in which the degrees follow a power law distribution.

In addition to scale-free networks, there may be other topologies that can shape the honeynet. Hence, the influence of some typical topologies on deception performance is investigated in Ren et al. (2021), which proposed a model in which the honeynet is partially infected by malware, and the spreading speed of this malware is checked in different network topologies such as ring, star, tree, and scale-free topologies. The results of this research stated that if the maximal characteristic value of the honeynet adjacency matrix (λ_{max}) is less than the ratio of the recovery rate of the infected hosts to the malware infection rate, the honeynet can experience a high-level performance. They calculated the boundaries of λ_{max} as given in Equation (17), where $N = h + r + z$.

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2} \le \lambda_{max} \le \min(d_{max}, \frac{N-1}{N}\sum_{i=1}^{N} d_i) \tag{17}$$

Considering Equation (17) and the condition mentioned for λ_{max}, the research suggested keeping the greatest host degree low for having an efficient honeynet. Since the model considers Various Topologies to model the Game, we call it VTG.

To analyze the researches performed in the field of honeynet topologies, we have simulated six networks in Python with different topologies infected by malware. These networks are shown in Fig. 13. N_1, N_2, and N_3 use ring, star, and tree topologies, respectively. N_4 uses k-regular topology, in which all the nodes have seven neighbor nodes. Finally, N_5 and N_6 are scale-free networks, where N_5 has a lower exponent than N_6. All the simulated networks have 50 nodes, among which 10 nodes are honeypots, and the malware initially infects five nodes. The honeypots and the initially infected hosts are placed randomly in the network and shown by yellow and red nodes in Fig. 13, respectively. The infected hosts can connect to normal hosts and exploit them with a specific probability. If an infected host connects to a honeypot, it will be recovered. The details of the malware propagation process are mentioned in Algorithm 3.

Algorithm 3 The malware propagation process in the simulations for comparing the researches about shaping the honeynet.

```
honeypots ← the list of honeypot nodes
infecteds ← the list of infected nodes
for each infection interval do
    for i ∈ infecteds do
        neighbors ← the list of node i's neighbor nodes
        flag ← a random number between 0 and 10
        if flag < infection_rate × 10 then
            for j ∈ neighbors do
                if j ∈ honeypots then
                    remove i from infecteds
                    break
                add j to infecteds
```

The results of these simulations are shown in Fig. 14. As shown in Fig. 14, the N_6 network has the lowest number of infected hosts, and it can better prevent the spreading of malware in our scenarios. Since N_6 and N_5 are scale-free networks, and the exponent in N_6 is higher, we can say that the suggestion of the DTG model in our scenarios is acceptable. On the other hand, according to Equation (17), the values of λ_{max} for N_1 and N_4 are two and seven, and for N_2 this value is greater than seven. λ_{max} for N_3 is greater than two and less than seven. However, a direct relation between λ_{max} and the final number of infected hosts is not observed.

5.6. Simulation - masking: elevating honeypot deception to new heights

Simulation - Masking is a deception technique that represents the zenith of sophistication within the realm of honeypot security. Its primary objective is to achieve a remarkable feat – making honeypots virtually indistinguishable from authentic network assets. This technique goes beyond merely inviting attackers; it artfully ensnares them within a web of intricate illusions, where the boundary between reality and deception becomes so blurred that even the most discerning adversaries

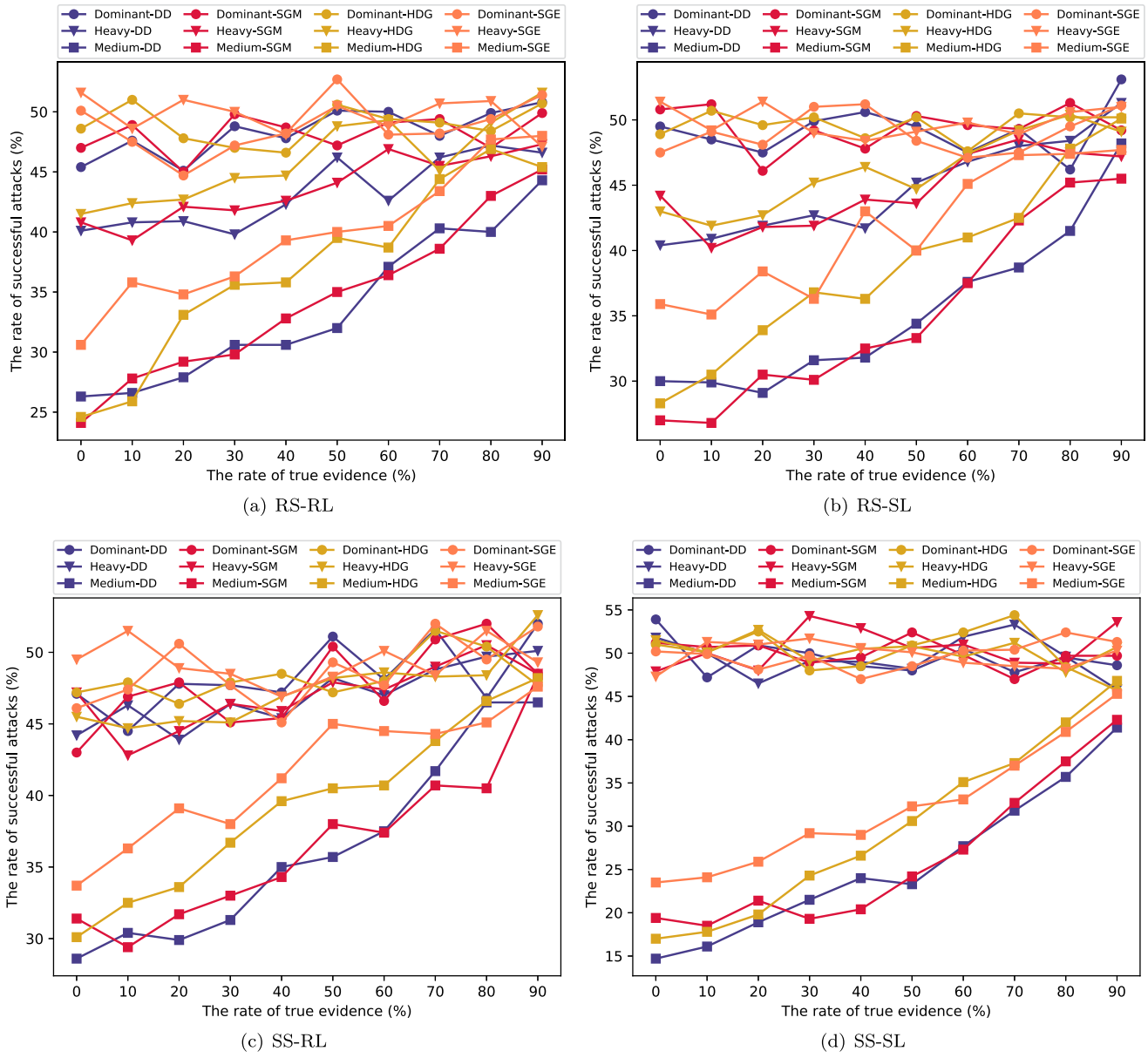
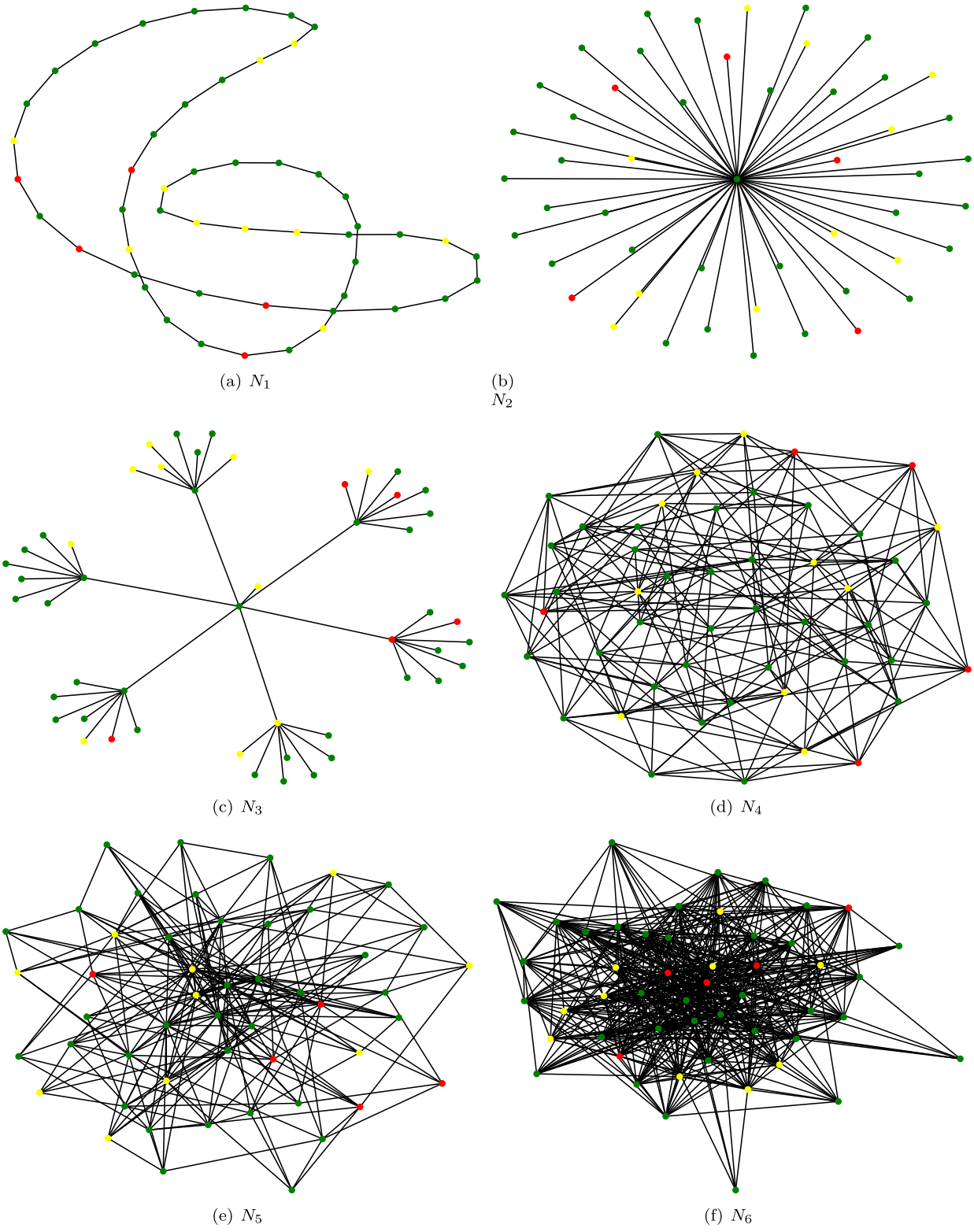


Fig. 12. Comparing the performance of DD, SGM, HDG, and SGE strategies in different scenarios.

find it challenging to differentiate. At its core, Simulation - Masking relies on the art of obfuscation. Honeypots are painstakingly engineered to emulate genuine network assets' characteristics, attributes, and behaviors. This emulation extends to many details, including altering system banners to match those of real services, meticulously replicating service behaviors, and mirroring the protocols commonly found within the network environment. The result is a honeypot that indistinguishably mimics a valuable, legitimate target to an attacker's untrained eye. The success of Simulation - Masking hinges on its ability to invite attackers and convince them of the honeypot's authenticity. The honeypot, shrouded in perfect camouflage, lures attackers into engaging with it, persuading them that it represents a golden opportunity. This deception places attackers in a precarious position as they interact with what they believe to be a legitimate asset. Within this interaction, honeypot administrators gain an intimate understanding of attacker tactics, techniques, and motives. What sets masking apart is the psychological toll it exacts on attackers. Attackers must navigate an environment fraught with uncertainty as they traverse the network. The inability to discern honeypots from tangible assets burdens attackers with cognitive dissonance, causing them to second-guess their every move. This cognitive load often leads to hesitation, mistakes, and detection, providing defenders invaluable response time. Deploying masking techniques within honeypots is an art form that demands precision and an intimate knowledge of network architecture. Honeypot configurations must seamlessly integrate with the broader network environment, creating a seamless illusion that even astute attackers cannot penetrate. Locations like public Wi-Fi zones and open network segments, known for attracting diverse user behaviors, serve as ideal deployment sites for masked honeypots. In these settings, honeypots mimic authentic network behaviors, capturing the most intricate attacker techniques and behaviors. In sum, Simulation-masking is a testament to the unmatched sophistication of honeypot technologies. It embodies the essence of honeypot deception by creating an environment where the line between reality and illusion is painstakingly blurred. When deftly employed by skilled defenders, this technique not only bolsters cybersecurity postures but also exposes the vulnerabilities of human psychology to deception in the constantly evolving landscape of cyber threats (Argyratos, 2021; Graham et al., 2016; Heckman et al., 2015; Wang et al., 2022). Throughout this sec-



(a) N_1 (b) N_2 (c) N_3 (d) N_4 (e) N_5 (f) N_6

Fig. 13. The simulated networks with different topologies. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

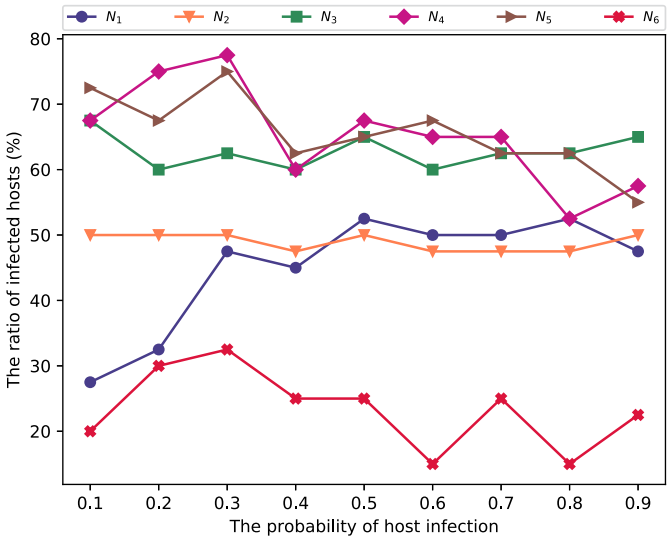


Fig. 14. Comparing the malware propagation process in different scenarios.

tion, we have thoroughly explored honeynet research, covering a broad range of related topics. We began by explaining the fundamental concepts of honeynets, which provides readers with a solid foundation in this area. Following this, we presented a comprehensive model for the systematic presentation of honeynets, which offers a structured framework to enhance the understanding and evaluation of these cyber defense systems. Continuing our exploration, we delved into the realm of deception techniques, which play a pivotal role in augmenting the performance of honeynets. These techniques were thoughtfully categorized into five distinct classes: optimization, diversification, location-based strategies, dynamism, and honeypot shaping. Each category was analyzed within the context of the proposed general model, enabling meaningful comparisons between these techniques and facilitating a deeper understanding of their respective merits and drawbacks.

To provide empirical insights into the efficacy of these techniques, we conducted simulation experiments using Python. These simulations covered a range of scenarios and allowed us to quantitatively assess the performance of various deception techniques within the context of honeynets. Through these simulations, we gained valuable data and observations, which were instrumental in refining our understanding of the most prominent techniques in this field. Using Python as the simulation platform ensured that our experiments were both rigorous and reproducible, contributing to the credibility of our findings.

5.7. Simulation - repackaging

Repackaging is a sophisticated and strategic technique that significantly enhances the effectiveness of honeypots in the relentless battle against cyber attackers. This method capitalizes on attackers' inclination to trust seemingly legitimate resources, often exploiting their curiosity and quest for valuable assets. It is a finely tuned art, demanding defenders to become adept at mastering the craft of embedding subtle yet highly effective deceptive elements within authentic resources. These deceptive elements are carefully constructed and capable of taking various forms, such as concealed scripts, modified executables, or even intricate alterations in data structures. These elements are strategically poised to trigger specific responses or actions when accessed by unsuspecting attackers, effectively ensnaring them in the honeypot's deception.

Resource diversity is a defining characteristic of Simulation - Repackaging. Defenders have the flexibility to transform a wide array of resources into honeypots that attract attackers. This technique extends beyond typical documents and applications to web pages, databases, and even email messages, serving as potential traps. This adaptability allows defenders to lure attackers with diverse, seemingly valuable assets, increasing the likelihood of engagement. The true power of Simulation - Repackaging lies in its early warning capabilities – as attackers interact with these deceptive resources, the technique captures an extensive range of valuable data on their actions during the initial stages of engagement. This data gives defenders a precious window into understanding malicious intent, enabling them to respond proactively before an attack escalates into a full-blown security breach.

Crafting a successful Simulation - Repackaging strategy necessitates meticulous attention to detail and an unwavering commitment to realism. Honeypot environments must be painstakingly designed to seamlessly replicate an organization's legitimate resources and services. This replication ensures that attackers encounter the repackaged resource without raising immediate suspicion, contributing to the effectiveness of the honeypot. Customization is paramount, with each honeypot deployment tailored to align with the unique threat landscape and specific organizational objectives. It is a dynamic process requiring ongoing adjustments.

Nevertheless, navigating the ethical and legal dimensions of deploying honeypots, particularly those employing Simulation - Repackaging, is vital. Defenders must address these concerns carefully and strictly comply with local laws and ethical guidelines. Transparency and adherence to ethical standards are essential to safeguard an organization's reputation and maintain the trust of stakeholders.

Beyond the early warning function, the insights extracted from Simulation - Repackaging should serve as a springboard for bolstering an organization's security posture. This includes timely vulnerability patching, updates to security policies, and the continuous refinement of incident response procedures. In the dynamic landscape of cybersecurity, where attackers continually adapt and evolve their tactics, Simulation - Repackaging emerges as a crafty and proactive guardian, empowering organizations with the foresight and tools necessary to fortify their cybersecurity defenses effectively. By combining the art of deception with meticulous planning and ethical considerations, defenders can proactively outmaneuver cyber adversaries and stay ahead in the ongoing battle for digital security (De Faveri and Moreira, 2016; De Faveri et al., 2018).

5.8. Simulation - repackaging

Simulation - Dazzling is a masterful and intricately designed deception technique that shines brilliantly within the realm of honeypots. It transcends traditional notions of deception by creating honeypot environments that not only trick potential attackers but bedazzle them. Imagine it as a digital spectacle, where the lines between illusion and reality blur, leaving cyber adversaries in awe. At its core, Simulation-Dazzling revolves around the art of crafting honeypot assets that are not just deceptive but visually striking and profoundly attractive to potential attackers. These honeypots are not mere traps; they are digital showcases adorned with elements that make them stand out like rare treasures within the network landscape. They resemble dazzling gemstones, irresistible to those seeking opportunities within the digital domain. A fundamental principle of Simulation-Dazzling is the strategic deployment of decoy assets that are engineered to be exceptionally enticing. These decoys come in various forms, from files bearing alluring names that promise valuable content to directories designed to tempt exploration or network services that radiate an irresistible allure. Placed with precision within the honeypot environment, these assets act as magnetic lures, drawing attackers in with the promise of what seems to be a digital jackpot. The success of Simulation-Dazzling lies not only in capturing attackers' attention but in monopolizing their focus. With their visual and psychological appeal, these dazzling honeypots frequently become attackers' top priorities, erroneously perceived as high-value targets. This diversion of attention away from genuine assets gives defenders a crucial tactical advantage. Moreover, Simulation-Dazzling

isn't solely about visual appeal; it also leverages psychological impact to its advantage. The presence of these striking decoys introduces a profound sense of cognitive dissonance within attackers. They are faced with the perplexity of evaluating the perceived significance of these assets, often leading to hesitation, prolonged interaction within the honeypot environment, and, ultimately, extensive data capture for defenders. Implementing Simulation - Dazzling within honeypots requires an in-depth understanding of attacker psychology and a discerning eye for design. Honeypot administrators must meticulously craft visually striking honeypot assets, positioning them strategically within the network. Public-facing servers, often primary targets for attackers, make ideal locations for deploying dazzling honeypots. The honeypots shine as irresistible beacons in these settings, effortlessly guiding attackers towards them. Harnessing the power of visual allure and psychological intrigue draws attackers into its mesmerizing web of deception. This technique not only diverts attackers' attention away from genuine assets but also equips defenders with a unique vantage point to observe, analyze, and comprehend attacker behavior in the ever-evolving landscape of cyber threats. Through its deceptive brilliance, Simulation-Dazzling illuminates the path toward more robust and proactive cybersecurity (Almeshekah and Spafford, 2014; De Faveri et al., 2018; Heckman et al., 2015).

5.9. Dissimulation - inventing deception beyond expectation

Inventing represents the pinnacle of innovation within the realm of honeypot deception techniques, pushing the boundaries of cybersecurity with its advanced and creatively inspired approach. It stands as a testament to the capacity of human ingenuity and imagination, redefining the very essence of honeypots. Imagine it as an act of crafting a mesmerizing digital narrative where the lines between reality and invention blur into an intricate tapestry of beguiling deception. At its core, Dissimulation - Inventing is an artful process of conjuring honeypot assets that exist solely in the realm of fiction, yet are meticulously designed with an unparalleled level of detail and sophistication. These honeypots transcend the conventional definition of decoys; they are complete stories, fully developed personas, or even entire virtual worlds that potential attackers might stumble upon during their illicit forays into the digital realm. A fundamental principle that underpins Dissimulation - Inventing is the profound craft of storytelling. Honeypot administrators assume the role of digital storytellers, weaving intricate narratives or fashioning fictive personas to engage and seduce potential attackers. These creative constructs manifest in diverse forms, from creating fictitious user accounts bearing compelling backstories to the inception of entirely fabricated digital environments or even the conjuring of intricate security vulnerabilities strategically embedded within the honeypot environment. The true triumph of Dissimulation - Inventing lies in its unparalleled ability not just to ensnare the attention of attackers but to immerse them within a captivating digital world. Attackers, drawn by the allure of the narratives or beguiled by fabricated vulnerabilities, invest substantial time and effort in their interactions with these honeypots, wholeheartedly convinced they have stumbled upon authentic targets. This immersion affords defenders a priceless opportunity to scrutinize attacker methodologies, motivations, and determination closely. Moreover, Dissimulation - Inventing transcends the realm of technical deception to explore the intricacies of human psychology. The engaging narratives and immersive experiences meticulously designed by this technique can evoke emotional responses in attackers, cultivating a sense of attachment and commitment to their interactions within the honeypot environment. This emotional investment often leads to protracted dwell times, extensive data capture, and a deeper understanding of attackers' behavioral idiosyncrasies. Implementing Dissimulation - Inventing within honeypots demands a unique fusion of creative genius and an intimate understanding of attacker psychology. Honeypot administrators must craft captivating digital narratives and ensure these imaginative honeypots align seamlessly with potential attackers' interests and objectives. These inventive honeypots are strategically placed within the network environment, often in areas that align with the natural path of exploration that attackers tend to follow. Dissimulation - Inventing represents the zenith of creativity within the honeypot landscape. Harnessing the enchanting power of storytelling and imaginative world-building draws potential attackers into its captivating web of deception. This technique captures attackers' attention and equips defenders with an extraordinary vantage point to decipher the intricate intricacies of attacker behavior within the ever-evolving and multifaceted world of cyber threats. Dissimulation - Inventing redefines the limits of honeypot innovation, propelling the art of deception far beyond conventional expectations (Cantella, 2021; De Faveri and Moreira, 2016).

5.10. Dissimulation - mimicking

Dissimulation - Mimicking is an extraordinary and supremely effective deception technique that stands as the epitome of excellence within the domain of honeypots. It represents the zenith of emulation, where honeypots are meticulously designed to mimic the appearance and behavior of real network assets to an astonishing degree, akin to crafting a digital doppelgänger that leaves attackers utterly convinced they have stumbled upon authentic resources.

At its core, dissimulation-mimicking involves the painstaking craft of honeypots that are virtually indistinguishable from genuine network resources. These honeypots ascend beyond superficial imitation, venturing deep into the heart of deception by replicating every facet of legitimate assets. This encompasses mirroring system banners, meticulously mimicking service behaviors, and even perfectly recreating network traffic patterns. The result is a honeypot that, to the keen eye of an attacker, appears not just authentic but entirely integral to the network's fabric.

A foundational principle that underlies dissimulation-mimicking is unwavering authenticity. Honeypot administrators engage in meticulous detail to ensure that their mimicked assets are faultlessly accurate. This entails creating honeypots that flawlessly emulate the precise configurations of network services, mirroring protocols to the finest nuances, and generating traffic indistinguishable from legitimate network activity.

The triumphant success of dissimulation-mimicking lies in its unparalleled ability to persuade attackers that they are genuinely interacting with authentic resources. Attackers, as they engage with these honeypots, are often left utterly unable to distinguish them from the tangible assets they fervently seek to compromise. This profound level of deception places attackers in a precarious position as they unknowingly unveil their tactics, techniques, and motives within the honeypot environment, where vigilant defenders stand ready to observe and respond. Furthermore, dissimulation-mimicking frequently affords the psychological advantage of cognitive dissonance for attackers. As they navigate the simulated network, attackers frequently encounter honeypots so compelling in their mimicry that they question the authenticity of everything around them. This cognitive burden can lead to hesitation, mistakes, and detection, providing defenders invaluable response time and insights into attacker behavior. Implementing Dissimulation - Mimicking within honeypots demands unparalleled attention to detail and an exhaustive understanding of network architecture. Honeypot configurations must be meticulously integrated into the broader network environment to create a virtually flawless illusion. These honeypots are most effectively deployed within critical network segments where their authenticity will be most convincing to potential attackers. Dissimulation-mimicking stands as the zenith of honeypot sophistication, embodying the essence of deception by creating an environment where attackers are left in awe of the authenticity of the assets they encounter. This technique not only serves to fortify cybersecurity postures but also endows defenders with an extraordinary opportunity to comprehend and thwart the intricacies of attacker behavior within the

perpetually evolving landscape of cyber threats. It represents the pinnacle of honeypot innovation, where perfect deception is honed to an unparalleled level of mastery (Whaley, 1982; Pawlick et al., 2019; Heckman et al., 2015).

5.11. Dissimulation - decoying

Dissimulation - Decoying is an enchanting and highly sophisticated deception technique that emerges as a captivating masterpiece within the realm of honeypots. It revolves around the intricate art of constructing honeypots that transcend mere deception to become intricate illusions, designed to bewilder and mislead potential attackers. Picture it as orchestrating a grand masquerade ball where the masks donned by the participants are not just disguises but intricate puzzles, leading to a labyrinth of intrigue.

At its heart, Dissimulation - Decoying entails the craft of fashioning honeypots that are deliberately labyrinthine, teeming with decoy elements meticulously positioned to divert and confound attackers. These honeypots are akin to intricate mazes where attackers must navigate a series of elaborate decoys before accessing valuable information or genuine network assets. A fundamental principle that anchors dissimulation-decoying is the art of misdirection. Honeypot administrators invest considerable effort to ensure the honeypot environment is teeming with decoys that convincingly mimic authentic assets. These decoys come in diverse forms, from deceptively named fake files and fabricated credentials to misleading network services. They are cunningly placed within the honeypot environment to tempt, distract, and sidetrack attackers. The resounding success of Dissimulation - Decoying lies in its ability to perplex and entangle attackers, leading them down a tortuous path of deception. Attackers, as they engage with these honeypots, soon find themselves ensnared in a web of decoys, squandering precious time and resources on what they believe to be high-value targets. This strategic diversion provides defenders with a commanding advantage as attackers expend vital assets navigating the labyrinth of deception. Furthermore, dissimulation - deceiving frequently exerts a psychological toll on attackers, invoking frustration as they encounter one decoy after another. This mounting disorientation and discouragement can lead to errors and detection. The psychological impact becomes a potent ally for defenders. Implementing Dissimulation - Decoying within honeypots necessitates a creative aptitude for misdirection and an intricate understanding of attacker psychology. Honeypot administrators must meticulously craft honeypot environments filled with decoys that are not just convincing but perplexing. These honeypots are often strategically placed in network segments where attackers are likely to explore but are not integral to the network's core functionality. Dissimulation - Decoying represents a mesmerizing facet of honeypot deception that captivates attackers with its intricacies, leading them into a bewildering maze of illusions. This technique not only bewilders and entangles attackers but also empowers defenders with a distinctive vantage point from which to observe, analyze, and comprehend the multifaceted intricacies of attacker behavior in the perpetually evolving landscape of cyber threats. It is a testament to the artistry of deception, where honeypots become intricate canvases of misdirection and intrigue (Cantella, 2021; Han et al., 2018; Heckman et al., 2015).

6. Exploring honeypot effectiveness through evaluation

The domain of honeypots presents a unique challenge in assessing their effectiveness, given their dual objective of deceiving automated attacks and manipulating human decision-making. A rigorous evaluation methodology is essential to measure their impact comprehensively. This section delves into the methodologies employed in the literature to evaluate honeypot effectiveness, specifically focusing on the significance of red-teaming experiments.

• The Multidimensional Evaluation Landscape: Evaluating honeypots involves navigating a multifaceted landscape that extends beyond conventional cybersecurity metrics. While technical benchmarks such as intrusion detection rates and attacker engagement duration are critical, the evaluation scope broadens when considering the influence of honeypots on human attackers. Human decision-making, often driven by emotions, biases, and cognitive processes, introduces a unique dimension that necessitates innovative evaluation approaches (Priya and Chakkaravarthy, 2023).

• Red-Teaming Experiments: At the forefront of honeypot evaluation methodologies are red-teaming experiments, which simulate real-world scenarios involving human adversaries. Red-teaming exercises replicate actual attackers' motivations, strategies, and decision-making processes, enabling researchers to assess honeypot effectiveness comprehensively. These experiments bridge the gap between technical capabilities and human psychology, providing insights into how honeypots interact with and influence the behaviors of attackers (Drew and Heinen, 2022).

• Technical and Psychological Dimensions: Red-teaming experiments encompass both technical and psychological dimensions. On the technical front, these evaluations measure how effectively honeypots thwart attacks launched by human attackers (Sethuraman et al., 2023). They also shed light on the strategies attackers employ to navigate honeypot environments. However, what distinguishes red-teaming experiments is their exploration of psychological dynamics. By mimicking the emotional triggers, cognitive biases, and social engineering tactics used by attackers, these experiments reveal the extent to which honeypots manipulate human decision-making.

• Holistic Honeypot Insights: Incorporating red-teaming experiments into the evaluation toolkit offers a holistic perspective on honeypot effectiveness. Such experiments reveal vulnerabilities in the technical aspects of honeypots and their capacity to influence attackers' actions and decisions. This holistic insight equips researchers and defenders with a comprehensive understanding of the evolving threat landscape, enabling them to enhance honeypot strategies that address both technical and human-centric challenges (Maesschalck et al., 2022; Kandanaarachchi et al., 2022).

• Enriching the Honeypot Arsenal: As the cybersecurity landscape evolves, the role of red-teaming experiments becomes increasingly crucial. They bridge the gap between simulated attacks and the complex behaviors of real attackers. By incorporating red-teaming methodologies into the evaluation process, honeypot effectiveness can be fine-tuned to reflect the dynamic interplay between automated attacks and human psychology. This enriched understanding empowers defenders to craft more robust deception strategies that effectively counter multifaceted threats (Chung et al., 2023).

• Incorporating an exploration of evaluation methodologies, particularly red-teaming experiments, in the paper enhances its depth by spotlighting the intersection of technical deception and the intricate psychology of human adversaries. This integration reflects the evolving nature of cybersecurity strategies and underscores the significance of human-centric deception techniques in fortifying cybersecurity postures.

• Unveiling the Role of Human Psychology in Cyber Deception Strategies:

Deception strategies in cybersecurity transcend the realm of technology and delve into the intricate realm of human psychology. Recognizing that the ultimate objective of deception is to influence human decision-making adds a layer of depth to the evaluation of security measures. This awareness becomes particularly pertinent when examining the efficacy of honeypots, as their success hinges on their ability to manipulate the behaviors of human adversaries. While conventional cybersecurity metrics provide invaluable insights, they often fall short of capturing the complex interplay between technology and human nature. The emergence of red-teaming experiments as a prominent evaluation approach signifies a conscious effort to bridge this gap (Gonzalez et al., 2022). These experiments, which replicate real-world scenarios

with actual attackers' motivations and cognitive processes, provide a unique lens to examine the interaction between deception strategies and human psychology. In the context of honeypots, the emphasis on red-teaming experiments reflects the evolving landscape of cybersecurity defenses. Beyond their technical capabilities, these experiments explore how honeypots can exploit cognitive biases, emotional triggers, and social engineering tactics to manipulate human behavior. By unraveling how attackers respond to these psychological manipulations, researchers gain insights to refine honeypot strategies, highlighting the symbiotic relationship between technology and human psychology. As the cybersecurity domain continues to evolve, it becomes increasingly evident that the effectiveness of deception techniques hinges on their capacity to influence and deceive human attackers. By highlighting the human-centric facet of deception, the chosen evaluation methodologies, particularly red-teaming experiments, enrich the understanding of how cybersecurity strategies must adapt to the dynamic interplay between technology and human psychology (Ferguson-Walter et al., 2023). In this perspective, honeypots transcend being mere technical tools and emerge as powerful instruments that harness human vulnerabilities to bolster overall cybersecurity posture. Incorporating this discussion underscores the integral connection between deception, technology, and human behavior, thus providing a holistic view of cybersecurity strategies in an ever-evolving threat landscape (Cranford et al., 2023).

7. Open issues

After reviewing the mentioned techniques and approaches, we identified several research gaps in the field of honeypots and honeynets. The researchers can further investigate these gaps to obtain more efficient honeypots and honeynets. Our suggestions are as follows.

7.1. Evaluating honeypots

Pursuing a precise and practical framework for comprehensively evaluating honeypot systems featuring diverse deception techniques remains a realm ripe for exploration. While we have outlined general metrics in section 3, their exact definitions warrant further refinement. To address this need, we propose utilizing a machine learning model incorporating the suggested metrics as input parameters. By employing this model, developers can objectively assess the performance of their honeypot systems and actively address their shortcomings. Navigating the intricacies of honeypot evaluation brings us to a challenging crossroads: prioritizing metrics in diverse scenarios. Take, for instance, an industrial network. Here, the significance of harvested data might pale compared to the paramount need to mitigate the risk of compromise. Safeguarding the integrity of devices takes precedence over the minutiae of dissecting threats for subsequent analysis. Consequently, researchers confront the pivotal task of identifying pertinent metrics and ascertaining their priorities. Harnessing the power of a machine learning model offers an adaptive framework that responds to contextual nuances. This model's capacity to process multiple metrics and their respective priorities equips developers with a powerful tool to gauge honeypot effectiveness across different scenarios. Yet, the journey to a comprehensive honeypot evaluation framework requires a deeper foray into the behavioral patterns of attackers and contextual dynamics. Continuous research is essential to uncover the ever-evolving strategies of cyber adversaries and the corresponding adaptations demanded in evaluation methodologies. Ultimately, the synergistic integration of machine learning, well-defined metrics, and a keen understanding of contextual subtleties will pave the way for a resilient and adaptable honeypot assessment paradigm. As cybersecurity advances, embracing innovative evaluation techniques is a transformative approach that reshapes how we quantify honeypot efficacy. This approach empowers developers to amplify their honeypot systems' strengths and mitigate their weaknesses actively, thus fostering a more robust cyber landscape.

7.2. Key metrics used for evaluating honeypots

A comprehensive evaluation of honeypots necessitates the application of a diverse range of metrics that collectively shed light on their performance and impact. These metrics serve as quantifiable benchmarks that guide the assessment process. Intrusion Detection Rate (IDR) measures the honeypot's efficiency in promptly identifying and alerting about unauthorized access attempts. The Engagement Rate quantifies the level of interaction between attackers and the honeypot, reflecting its capacity to captivate adversaries. Time to Compromise evaluates how effectively the honeypot deters and delays attackers by extending the duration to breach. Data Captured assesses the richness and volume of information gleaned from attacker interactions (Santhosh Kumar et al., 2023). A False Positive Rate indicates how frequently legitimate users trigger alerts, ensuring operational continuity. Attack Attribution measures the accuracy with which the honeypot identifies attackers, enhancing threat intelligence. The Attack Complexity metric offers insights into the sophistication of attacks attempted. Deception Depth evaluates the honeypot's success in encouraging in-depth engagement by enticing attackers. Interaction Diversity measures the variety of strategies attackers employ within the honeypot environment. Early Warning gauges how swiftly the honeypot detects and communicates emerging threats. Resource utilization assesses the impact of the honeypot on infrastructure and its ability to attract attackers. The Attack Repellent Effectiveness metric reveals how adeptly the honeypot deflects attackers from critical assets. Impact on Attacker Behavior analyzes whether the honeypot influences attackers to adapt their tactics. Threat Intelligence Yield quantifies the value of collected data in informing broader cybersecurity strategies. Lastly, Honeypot Resilience evaluates the honeypot's robustness in maintaining its deceptive façade under attack. These metrics collectively enable a comprehensive evaluation framework that considers both technical and human-centric aspects, enriching the understanding of honeypot effectiveness (Eriksson, 2023). These metrics collectively provide a holistic view of a honeypot's performance, its impact on the threat landscape, and its contributions to improving cybersecurity strategies and incident response. The selection and interpretation of metrics should align with the specific goals of the honeypot deployment and the desired outcomes of the evaluation process. The descriptions for each item can be found in the following.

- Intrusion Detection Rate (IDR): The Intrusion Detection Rate measures the effectiveness of the honeypot in identifying and alerting about unauthorized access attempts. A higher IDR indicates that the honeypot's detection mechanisms recognize suspicious activities, helping defenders promptly respond to potential threats (Raharjo et al., 2022).
- Engagement Rate: The Engagement Rate signifies the level of interaction between attackers and the honeypot. A higher engagement rate suggests that the honeypot successfully lures and captures the attention of attackers, facilitating data collection and deeper insights into their tactics and intentions (Panda et al., 2022).
- Time to Compromise: This metric gauges the time attackers take to breach the honeypot's defenses. A longer Time to Compromise indicates that the honeypot effectively prolongs attackers' efforts, granting defenders more time to identify, analyze, and respond to the intrusion (Hobert et al., 2023).
- Data Captured: Data Captured assesses the quantity and quality of information collected during interactions with attackers. This encompasses network traffic, commands issued, files accessed, and other actions undertaken by attackers within the honeypot environment (Ikuomenisan and Morgan, 2022).
- False Positive Rate: The False Positive Rate calculates the frequency with which legitimate users or automated systems trigger alerts or engage with the honeypot. Minimizing the False Positive Rate ensures the honeypot doesn't impede normal operations or needlessly consume resources (Kandanaarachchi et al., 2022).
- Attack Attribution: Attack Attribution evaluates how accurately the honeypot identifies the origin and identity of attackers. Compelling at-

tribution provides valuable insights into attackers' geographic locations, affiliations, and potential motivations (Crochelet et al., 2022).

• Attack Complexity: This metric measures the sophistication level of attacks directed at the honeypot. Complex attacks may indicate that the honeypot attracts skilled adversaries, while more straightforward attacks might reflect opportunistic attempts from less sophisticated threat actors (Yang et al., 2023).

• Deception Depth: Deception Depth gauges how effectively the honeypot creates an environment that lures attackers into engaging deeply. A high Deception Depth suggests that attackers invest significant time and effort, revealing more about their intentions and techniques (Sumadi et al., 2022).

• Interaction Diversity: Interaction Diversity assesses the variety of ways attackers engage with the honeypot. A broad range of interactions provides insights into attackers' strategies and goals, from probing to attempting various attack vectors (Srinivasa et al., 2022).

• Early Warning: Early Warning measures how quickly the honeypot detects and alerts defenders about emerging threats. Swift detection empowers cybersecurity teams to respond promptly, mitigating potential risks before they escalate (Salimova, 2022).

• Resource Utilization: Resource Utilization evaluates the impact of the honeypot on the underlying infrastructure. High resource utilization might indicate that the honeypot effectively attracts and engages attackers, consuming their time and resources (Abdulqadder et al., 2023).

• Attack Repellent Effectiveness: This metric assesses how well the honeypot redirects attackers from targeting actual production systems. A successful attack-repellent strategy diverts attackers from high-value targets, reducing the risk to critical assets (Yamin and Katt, 2022).

• Impact on Attacker Behavior: Impact on Attacker Behavior analyzes whether the honeypot influences attackers to modify their tactics or techniques. Identifying changes in behavior can inform defenders about evolving threats and attackers' adaptation strategies (Tabari et al., 2023).

• Threat Intelligence Yield: Threat Intelligence Yield quantifies how data collected from the honeypot contributes to the organization's threat intelligence. Valuable insights gained from the honeypot inform overall cybersecurity strategies and decision-making (Tan et al., 2023).

• Honeypot Resilience: It evaluates the honeypot's ability to withstand attacks and maintain its deceptive façade. A resilient honeypot remains operational despite intense scrutiny, continuing to engage and collect data from attackers (Alyas et al., 2022).

7.3. Industrial honeypots

In the ever-evolving landscape of cybersecurity, where threats escalate with unprecedented speed, the strategic deployment of honeypots has emerged as a pivotal defense strategy. Specialized iterations of honeypots have arisen during this dynamic environment to address the challenges posed by rapidly evolving technological landscapes. Wireless honeypots have emerged as formidable tools, strategically crafting simulated Wi-Fi networks to entice potential attackers and illuminate vulnerabilities unique to wireless environments. These honeypots unravel the complexities of rogue access points, eavesdropping endeavors, and unauthorized connections, offering invaluable insights crucial for safeguarding wireless networks. In parallel, the surge in automation and interconnected systems has given birth to industrial honeypots, replicating industrial control systems and supervisory control and data acquisition networks. These virtual constructs beckon adversaries, inviting them to traverse deceptive landscapes that mirror the intricate realm of modern industry. As the narrative unfolds, this paper explores these specialized honeypot types, unraveling the strategies behind their deployment, their distinct advantages, and the profound insights they furnish into the domain of cyber deception and defense (Pashaei et al., 2022).

Diving further into the realm of wireless and industrial honeypots:

• Wireless Honeypots: Navigating Cyber Shadows in the Airwaves

Wireless honeypots have risen as strategic defenses within the backdrop of ubiquitous wireless networks. These honeypots meticulously simulate authentic Wi-Fi networks, strategically attracting attackers and unearthing vulnerabilities inherent to wireless systems. Their role extends to detecting unauthorized access points, thwarting illicit connection attempts, and unveiling instances of wireless eavesdropping. By discerning between legitimate user behaviors and malicious actions, wireless honeypots provide unparalleled insights into the methodologies employed by attackers to exploit the weak points within wireless environments (Soundararajan et al., 2022). Their deployment demands meticulous attention to network configurations, signal interference, and signal strength. Notably, public Wi-Fi zones, bustling with diverse user activity, serve as ideal grounds for deploying these honeypots. By mirroring genuine network behaviors, these honeypots capture and meticulously analyze attack strategies that adversaries might employ to compromise user data or infiltrate organizational systems.

• Industrial Honeypots: Safeguarding the Core of Modern Industry

As industries embrace automation and interconnected systems, a fundamental shift has occurred in industrial environments. This transformation has heralded the necessity for specialized cybersecurity measures, epitomized by the advent of industrial honeypots. These honeypots ingeniously emulate the intricacies of industrial control systems (ICS) and supervisory control and data acquisition (SCADA) networks, inviting attackers to reveal their tactics and maneuvers. Operating as virtual battlegrounds, industrial honeypots show how adversaries might target critical infrastructures. By emulating the unique components of industrial networks, such as programmable logic controllers (PLCs) and human-machine interfaces (HMIs), these honeypots capture instances of intrusion, unauthorized control commands, and other malevolent activities (Conti et al., 2022; Apruzzese et al., 2023). Additionally, they contribute to the cultivation of targeted threat intelligence specific to industrial sectors like energy, manufacturing, and transportation. It is important to note that deploying industrial honeypots necessitates a profound comprehension of industrial processes, protocols, and communication patterns. Security experts must meticulously replicate the intricacies of these systems to construct deceptive environments that accurately mirror the operational technology landscape.

• Bridging Gaps and Amplifying Cyber Defenses

Incorporating in-depth discussions on wireless and industrial honeypots within our paper is an endeavor that bridges the chasm between evolving cybersecurity challenges and the implementation of advanced deception strategies. By highlighting the distinctive attributes of these specialized honeypot types, our work gains heightened relevance, aligning seamlessly with the contemporary focus on safeguarding wireless networks and safeguarding critical industrial infrastructure (Jha, 2023). The exploration of deployment intricacies, attack vectors, and the manifold advantages of these honeypots furnishes our paper with a comprehensive and panoramic view of the versatile realm of honeypot technologies.

7.4. SDN-based honeypots

In the context of Software-Defined Networking (SDN) and cloud computing, efficient resource management and robust security are critical, making honeypot incorporation even more significant (Javadpour and Wang, 2022; Javadpour et al., 2023b). Notably, while several types of research have explored the advanced mimicking technique, as introduced in subsection 3.1, many of these studies have been primarily focused on emulating the functionalities of machines within traditional networks. Unfortunately, they overlook the distinct services and vulnerabilities unique to SDN and cloud computing environments. One of the hallmark features of SDN environments and cloud computing networks is their central controlling component, often called the SDN controller or cloud orchestrator. This centralization presents an intriguing opportunity for deploying deception mechanisms to safeguard the network. We recommend that researchers channel their efforts toward

developing honeypots engineered to faithfully mimic the functionalities of SDN controllers and cloud orchestrators. This strategic approach can facilitate a more comprehensive analysis of attacks targeting SDN controllers and cloud management systems and contribute to developing proactive security measures in cloud computing networks. TWO NOTABLE THREATS STAND OUT within SDN, cloud computing, and their intersection: topology poisoning (Adjou et al., 2022; Khoa et al., 2023) and Distributed Denial of Service (DDoS) attacks directed at the controller or cloud orchestrator. In topology poisoning attacks, adversaries manipulate the topology-related data exchanged between OpenFlow switches and the SDN controller or cloud orchestrator, effectively camouflaging the network topology. To gain deeper insights into the tactics employed in these attacks, we recommend the development of fake OpenFlow switches designed as honeypots. These intentionally vulnerable honeypot switches would willingly expose themselves to topology poisoning attacks, providing researchers with valuable intelligence about these threats. Furthermore, adversaries may launch DDoS attacks against the SDN controller, cloud orchestrator, or cloud resources to incapacitate them by overwhelming their communication channels. To analyze and proactively mitigate such attacks, we suggest implementing an SDN environment with multiple controllers, as exemplified by the work of Javadpour (2020), and multiple cloud orchestrators in cloud computing networks. Subsequently, deploying fake controllers and orchestrators alongside legitimate ones, in the capacity of honeypot controllers and orchestrators, could bolster the network's defenses. These honeypot controllers and orchestrators can craft deceptive fake rules on the switches and cloud resources or induce the switches and cloud resources to transmit fabricated status messages to them. This proactive strategy enables early detection of DDoS attempts and fosters a deeper understanding of the tactics employed by adversaries targeting SDN controllers and cloud orchestrators in cloud computing networks. In conclusion, integrating honeypots tailored to the specific demands of SDN environments, cloud computing networks, and their convergence holds immense promise for enhancing resource management and security in the cloud. By faithfully emulating SDN controllers, cloud orchestrators, and associated components and strategically deploying honeypots to counteract topology poisoning and DDoS threats, researchers and network administrators can gain profound insights into potential vulnerabilities and devise effective countermeasures to fortify cloud computing networks against evolving cyber threats (Anwar et al., 2022).

7.5. 5G-based honeypots

Expanding on using honeypots to bolster 5G network security, it is essential to delve deeper into the specific vulnerabilities associated with each network component and the potential benefits of employing mimicking techniques. First and foremost, the core network forms the backbone of 5G services, comprising an array of critical physical infrastructure. These machines are the lifeblood of 5G connectivity, and any disruption to their operations can have far-reaching consequences. By deploying honeypots that effectively impersonate these essential core network components, we can divert the focus of potential adversaries away from genuine assets. This diversion is a deterrent and provides a unique opportunity to gather intelligence on potential threats, attack methodologies, and adversaries. Moving the radio access network represents a pivotal link in the 5G chain, encompassing wireless connections and interfaces. While there has been some recognition of the need to secure this aspect of the network, detailed implementation strategies and performance assessments have been lacking (Javadpour et al., 2023c; Benzaïd et al., 2022). This underscores the importance of further research and development in this area. Honeypots designed to mimic radio access network elements can be invaluable for safeguarding these components and gaining insights into how adversaries target them. Moreover, the role of client-side honeypots in identifying vulnerabilities within end devices cannot be understated. These devices, often considered the final point of interaction in the 5G network, are susceptible to various security threats. Client-side honeypots can simulate these devices, creating a buffer against potential attacks and gathering data on the tactics employed by malicious actors. The integration of specialized honeypots, tailored to mimic different 5G network components, offers a multi-faceted approach to network security enhancement. By comprehensively addressing vulnerabilities at various levels of the 5G architecture, we fortify the network against potential threats and gain a deeper understanding of the evolving threat landscape. This knowledge can then be used to fine-tune security measures and ultimately ensure the robustness and resilience of the 5G network in an era of rapid technological advancement and increasing cybersecurity challenges (Kheir et al., 2022).

7.6. Honeypots and botnets

In subsection 3.2, we presented the research about the honeypots cooperating with the adversary and pretending to help him/her. However, the mentioned honeypots with the cooperating deception technique can be even more improved. The botnets and complicated threats are growing daily, giving us more information about their behavior. The researchers can use this information to design powerful honeypots that are deceptive in cooperating with adversaries. The honeypots can be designed to provide the adversary with fake help in the different phases of botnets' lifecycle.

It may be challenging to cooperate with a botmaster. Because some botnets are complicated and have different types of members. Hence, we suggest the researchers identify different botnet members in a network and their roles to cooperate with them effectively. For instance, one of the members in the loader-based botnets, such as Mirai, is the loader. The bots probe the whole network and then report the penetrated host username and password. Then the loader will infect that host with the malware script. In such botnets, we suggest the honeypots act like a bot and report the credential pairs of another honeypot to the loader. In this condition, the credentials are valid, and the loader believes the honeypot is on its side.

7.7. Distributed honeypots

The Traffic Redirection technique (presented in subsection 3.6) is widely deployed in different deceptive networks. However, the traffic congestion toward the honeypots is not analyzed. To improve the performance of honeypots in a cost-limited network, we suggest the researchers work on the virtualization mechanisms such as virtual network embedding concepts, which are used by Javadpour and Wang (2022) and Javadpour (2019), to effectively distribute the functionalities of a honeypot and its traffic among different network nodes and paths, respectively. This will help them use the minimum possible amount of resources. The traffic must first be analyzed and then redirected to the appropriate node.

Synchronizing the distributed honeypots and securing their connection is challenging. As a result, the researchers have to work on protected communication channels, such as blockchain, to synchronize the distributed honeypots securely.

7.8. Learning honeynets

Diversifying the honeypots and locating them in a honeynet (mentioned in subsection 5.2 and subsection 5.3) are two deception techniques that we think can be improved by machine learning approaches. We suggest the researchers collect useful information to create a learning model that can predict the services that are commonly targeted by the current threat spread over the network. This prediction helps the honeypots to simulate the services that can attract adversaries at a higher rate.

The first point that must be noted is that some machine learning models are vulnerable to cyber-attacks (Benzaïd and Taleb, 2020). If these models are not designed based on security factors, they may risk the honeypots extra. Therefore, the researchers must consider the protective mechanisms in developing honeypot machine-learning models. The other point is that training the model must not cause additional overhead to a honeynet. The lightweight models are good choices for being used in a honeynet.

The other suggestion is dynamically changing the location of honeypots in a honeynet, which leads to a lower cost of deployment and higher efficiency in wasting the adversaries' time. One can use Moving Target Defense (MTD) concepts to change the optimal set of honeypots. MTD approaches try to change the attack surface by changing the location of the adversary's targets. However, using MTD concepts is challenging because the changes may warn the adversary about abnormal events in the network. Hence, the developers have to pay attention to the shuffling frequency of the honeypots to conceal the changes from the adversary. Machine learning models can be trained to find the optimal shuffling frequency. Another thing that can be considered for honeypots is their placement in the satellite network. In this way, deploying different honeypots can prevent DoS and DDoS attacks. And using MTD methods, the acceptance rate to the network will be reduced.

7.9. Understanding vulnerability types in cybersecurity

In the dynamic landscape of cybersecurity, it is imperative to recognize that not all vulnerabilities are created equal. The significance and attractiveness of a vulnerability to potential adversaries can vary significantly, and this nuanced aspect has profound implications for cybersecurity strategies. While our research has focused on performance and optimization in the context of honeynets, we acknowledge a crucial dimension that needs to be incorporated - the influence of vulnerability types on attacker motivations and behavior. This addition can provide a more holistic understanding of the effectiveness of honeynets in defending against specific types of vulnerabilities (Jones, 2022; McCoy, 2022).

• Integration of Vulnerability Types (Agarwal, 2022): To address this vital aspect, we propose an expansion of our research framework to incorporate vulnerability types as a factor in our analysis. Notably, vulnerabilities such as EternalBlue and Log4j have different levels of appeal to potential attackers due to factors like their exploitability, potential for widespread impact, and financial incentives. Our analysis will aim to distinguish between these vulnerability types, and in doing so, provide valuable insights into how the effectiveness of honeynets may vary in the face of distinct adversary interests.

• Relevance to the Current Study: While our research has laid the groundwork for optimizing honeynets, considering varying adversary interests based on vulnerability types is an essential component for a comprehensive understanding of honeynet performance. This extension not only adds depth to our investigation but also bolsters the practical applicability of our findings. By including this dimension, we aim to provide a more nuanced perspective on honeynets' effectiveness in the ever-evolving cybersecurity landscape (Rich, 2023).

• Incorporating this dimension into our research will involve categorizing vulnerabilities into distinct types based on their attractiveness to potential adversaries. We will then conduct our analysis focusing on these categories, evaluating how honeynets perform and can be optimized differently for each vulnerability type. Our methodology will incorporate real-world data and simulations to substantiate our findings.

• EternalBlue: EternalBlue is a notorious software vulnerability that gained notoriety due to its involvement in the global WannaCry ransomware attack in 2017. Initially identified by the United States National Security Agency (NSA), this vulnerability targets the Windows operating system. EternalBlue allows malicious actors to exploit a Server Message Block (SMB) protocol flaw, enabling them to propagate malware and execute arbitrary code remotely on vulnerable systems. The significant impact and rapid spread of WannaCry shed light on the critical need for timely software patching and effective cybersecurity measures (Riggs et al., 2023; Ibrahim et al., 2023).

• Log4j: Log4j, formally known as Apache Log4j, is a widely-used open-source logging library for Java applications. In December 2021, a severe security vulnerability, often referred to as "Log4Shell" or "Log4j vulnerability," was discovered in Log4j. This vulnerability, tracked as CVE-2021-44228, allows attackers to execute arbitrary code remotely by exploiting the library's ability to process log entries. The Log4j vulnerability raised substantial concerns across the cybersecurity community due to its ubiquitous use in Java applications, which are employed in various critical systems and services. It emphasized the importance of swift patching and vulnerability management in the face of emerging threats (Rossotti, 2022; Feng and Lubis, 2022).

8. Conclusion and suggestions

This survey offers a detailed exploration of honeypot research over the past two decades. We begin by explaining the fundamental concepts underlying honeypots, which serve as a basis for categorizing and analyzing these security mechanisms based on their purposes, modes of interaction, implementation methodologies, operational activities, stakeholders involved, consistency, and uniformity. These categories provide a structured framework for understanding honeypots and offer insights for developers who need to choose the most suitable type for their specific security needs. As we strive to improve the effectiveness of honeypots, we have conducted a thorough investigation into techniques of deception that can enhance the performance of individual honeypots. These techniques can be organized into six groups, each with unique strategies that can be used to avoid detection and attract potential threats. The six groups include advanced mimicking, fake cooperation, manipulation of Deceptive Databases, subtle interruptions, honeytoken baiting, and traffic redirection. To evaluate the effectiveness of security techniques, we propose a set of measurement metrics designed to be robust and practical in various scenarios. We also examine different deception techniques used in honeynets and how they can enhance their performance. These techniques are grouped into categories based on their purpose: optimization, diversification, location, dynamization, and shaping of honeypots within the network. We summarize relevant research and models in each category to enable comparative analysis. To simplify this process, we suggest a comprehensive general model to help select the most appropriate approach for a given context. To explore the practicality of key deception techniques, we conduct simulation scenarios using Python. These simulations provide valuable insights into the potential outcomes and efficacy of deploying deception mechanisms within a network environment. After the survey concludes, we highlight open issues and challenges that require further investigation while providing strategic recommendations for the evolving landscape of honeypots and honeynet deception techniques. This compendium is a valuable resource for researchers and practitioners in the ever-evolving field of cybersecurity, serving to inform and inspire advancements.

CRediT authorship contribution statement

Amir Javadpour: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Forough Ja'fari:** Data curation, Methodology, Supervision, Writing – original draft. **Tarik Taleb:** Resources, Writing – original draft, Writing – review & editing. **Mohammad Shojafar:** Investigation, Writing – original draft, Writing – review & editing. **Chafika Benzaïd:** Data curation, Investigation, Visualization, Writing – review & editing.

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.

Data availability

No data was used for the research described in the article.

Acknowledgement

This research work is partially supported by the European Union's Horizon Europe research and innovation program HORIZON-JU-SNS-2022 under the RIGOUROUS project (Grant No. 101095933). The paper reflects only the authors' views. The Commission is not responsible for any use that may be made of the information it contains.

References

Abay, N.C., Akcora, C.G., Zhou, Y., Kantarcioglu, M., Thuraisingham, B., 2019. Using deep learning to generate relational honeydata. In: Autonomous Cyber Deception. Springer, pp. 3–19.

Abdulqadder, I.H., Zou, D., Aziz, I.T., 2023. The dag blockchain: a secure edge assisted honeypot for attack detection and multi-controller based load balancing in sdn 5g. Future Gener. Comput. Syst. 141, 339–354.

Achleitner, S., La Porta, T.F., McDaniel, P., Sugrim, S., Krishnamurthy, S.V., Chadha, R., 2017. Deceiving network reconnaissance using sdn-based virtual topologies. IEEE Trans. Netw. Serv. Manag. 14 (4), 1098–1112.

Ackerman, P., 2020. Modern Cybersecurity Practices: Exploring and Implementing Agile Cybersecurity Frameworks and Strategies for Your Organization. BPB Publications.

Adjou, M.L., Benzaïd, C., Taleb, T., 2022. Topotrust: a blockchain-based trustless and secure topology discovery in sdns. In: 2022 International Wireless Communications and Mobile Computing (IWCMC), pp. 1107–1112.

Agarwal, Y., 2022. Apache Log4j Logging Framework and Its Vulnerability.

Akingbola, Dahunsi, Alese, Adewale, Ogundele, 2015. Improving deception capability in honeynet through data manipulation. J. Internet Technol. Secur. Trans. 4, 373–379.

Akiyama, M., Yagi, T., Hariu, T., Kadobayashi, Y., 2018. Honeycirculator: distributing credential honeytoken for introspection of web-based attack cycle. Int. J. Inf. Secur. 17 (2), 135–151.

Almeshekah, M.H., Spafford, E.H., 2014. Planning and integrating deception into computer security defenses. In: Proceedings of the 2014 New Security Paradigms Workshop, pp. 127–138.

Almeshekah, M.H., Spafford, E.H., 2016. Cyber security deception. In: Cyber Deception. Springer, pp. 25–52.

Almeshekah, M.H., Spafford, E.H., Atallah, M.J., 2013. Improving security using deception. Center for Education and Research Information Assurance and Security, Purdue University, Tech. Rep. CERIAS Tech Report, vol. 13. p. 2013.

Alosefer, Y., Rana, O., 2010. Honeyware: a web-based low interaction client honeypot. In: 2010 Third International Conference on Software Testing, Verification, and Validation Workshops. IEEE, pp. 410–417.

Althonayan, A., Andronache, A., 2019. Resiliency under strategic foresight: the effects of cybersecurity management and enterprise risk management alignment. In: 2019 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA). IEEE, pp. 1–9.

Alyas, T., Alissa, K., Alqahtani, M., Faiz, T., Alsaif, S.A., Tabassum, N., Naqvi, H.H., 2022. Multi-cloud integration security framework using honeypots. Mob. Inf. Syst. 2022, 1–13.

Anwar, A.H., Kamhoua, C., Leslie, N., 2019. A game-theoretic framework for dynamic cyber deception in Internet of battlefield things. In: Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, pp. 522–526.

Anwar, A.H., Kamhoua, C., Leslie, N., 2020. Honeypot allocation over attack graphs in cyber deception games. In: 2020 International Conference on Computing, Networking and Communications (ICNC). IEEE, pp. 502–506.

Anwar, A.H., Kamhoua, C.A., Leslie, N.O., Kiekintveld, C., 2022. Honeypot allocation for cyber deception under uncertainty. IEEE Trans. Netw. Serv. Manag. 19 (3), 3438–3452.

Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A.V., Di Franco, F., 2023. The role of machine learning in cybersecurity. Digit. Treats Res. Pract. 4 (1), 1–38.

Argyratos, T., 2021. Information theoretic analysis of deception and decisions in networks. Ph.D. dissertation. Aristotle University of Thessaloniki.

Ayeni, O., Alese, B., Omotosho, L., 2013. Design and implementation of a medium interaction honeypot. Int. J. Comput. Appl. 975, 8887.

Badr, Y., Hariri, S., Youssif, A.-N., Blasch, E., 2015. Resilient and trustworthy dynamic data-driven application systems (dddas) services for crisis management environments. Proc. Comput. Sci. 51, 2623–2637.

Bedi, H.S., Roy, S., Shiva, S., 2011. Game theory-based defense mechanisms against ddos attacks on tcp/tcp-friendly flows. In: 2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS). IEEE, pp. 129–136.

Benzaïd, C., Taleb, T., 2020. Ai for beyond 5g networks: a cyber-security defense or offense enabler? IEEE Netw. 34 (6), 140–147.

Benzaïd, C., Taleb, T., Song, J., 2022. Ai-based autonomic and scalable security management architecture for secure network slicing in b5g. IEEE Netw. 36 (6), 165–174.

Bercovitch, M., Renford, M., Hasson, L., Shabtai, A., Rokach, L., Elovici, Y., 2011. Honeygen: an automated honeytokens generator. In: Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics. IEEE, pp. 131–136.

Biedermann, S., Mink, M., Katzenbeisser, S., 2012. Fast dynamic extracted honeypots in cloud computing. In: Proceedings of the 2012 ACM Workshop on Cloud Computing Security Workshop, pp. 13–18.

Bilinski, M., Gabrys, R., Mauger, J., 2018. Optimal placement of honeypots for network defense. In: International Conference on Decision and Game Theory for Security. Springer, pp. 115–126.

Bowen, B.M., Hershkop, S., Keromytis, A.D., Stolfo, S.J., 2009. Baiting inside attackers using decoy documents. In: Security and Privacy in Communication Networks: 5th International ICST Conference. SecureComm 2009, Athens, Greece, September 14-18, 2009, Revised Selected Papers 5. Springer, pp. 51–70.

Bringer, M.L., Chelmecki, C.A., Fujinoki, H., 2012. A survey: recent advances and future trends in honeypot research. Int. J. Comput. Netw. Inf. Secur. 4 (10), 63.

Cai, J.-Y., Yegneswaran, V., Alfeld, C., Barford, P., 2009. An attacker-defender game for honeynets. In: COCOON. Springer, pp. 7–16.

Cantella, E., 2021. Architectural Style: Distortions for Deploying and Managing Deception Technologies in Software Systems. Rochester Institute of Technology.

Carroll, T.E., Grosu, D., 2011. A game theoretic investigation of deception in network security. Secur. Commun. Netw. 4 (10), 1162–1172.

Çeker, H., Zhuang, J., Upadhyaya, S., La, Q.D., Soong, B.-H., 2016. Deception-based game theoretical approach to mitigate dos attacks. In: International Conference on Decision and Game Theory for Security. Springer, pp. 18–38.

Chakraborty, T., Jajodia, S., Katz, J., Picariello, A., Sperli, G., Subrahmanian, V., 2019. Forge: a fake online repository generation engine for cyber deception. IEEE Trans. Dependable Secure Comput.

Chen, T.M., Buford, J., 2009. Design considerations for a honeypot for sql injection attacks. In: 2009 IEEE 34th Conference on Local Computer Networks. IEEE, pp. 915–921.

Chung, M.-H., Yang, Y., Wang, L., Cento, G., Jerath, K., Raman, A., Lie, D., Chignell, M.H., 2023. Implementing data exfiltration defense in situ: a survey of countermeasures and human involvement. ACM Comput. Surv.

Conti, M., Trolese, F., Turrin, F., 2022. Icspot: a high-interaction honeypot for industrial control systems. In: 2022 International Symposium on Networks, Computers and Communications (ISNCC). IEEE, pp. 1–4.

Cranford, E., Ou, H.-C., Gonzalez, C., Tambe, M., Lebiere, C., 2023. Accounting for Uncertainty in Deceptive Signaling for Cybersecurity.

Crochelet, P., Neal, C., Cuppens, N.B., Cuppens, F., 2022. Attacker attribution via characteristics inference using honeypot data. In: International Conference on Network and System Security. Springer, pp. 155–169.

Crouse, M., Prosser, B., Fulp, E.W., 2015. Probabilistic performance analysis of moving target and deception reconnaissance defenses. In: Proceedings of the Second ACM Workshop on Moving Target Defense. ACM, pp. 21–29.

Crouse, M.B., 2012. Performance analysis of cyber deception using probabilistic models. Master's thesis. Wake Forest University Graduate School of Arts and Sciences, Winston-Salem, North Carolina.

Dahbul, R., Lim, C., Purnama, J., 2017. Enhancing honeypot deception capability through network service fingerprinting. In: Journal of Physics: Conference Series, vol. 801. IOP Publishing, p. 012057.

Dalamagkas, C., Sarigiannidis, P., Ioannidis, D., Iturbe, E., Nikolis, O., Ramos, F., Rios, E., Sarigiannidis, A., Tzovaras, D., 2019. A survey on honeypots, honeynets and their applications on smart grid. In: 2019 IEEE Conference on Network Softwarization (NetSoft). IEEE, pp. 93–100.

Dantu, R., Cangussu, J.W., Patwardhan, S., 2007. Fast worm containment using feedback control. IEEE Trans. Dependable Secure Comput. 4 (2), 119–136.

De Faveri, C., Moreira, A., 2016. Designing adaptive deception strategies. In: 2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, pp. 77–84.

De Faveri, C., Moreira, A., Amaral, V., 2018. Multi-paradigm deception modeling for cyber defense. J. Syst. Softw. 141, 32–51.

de Nobrega, K., 2023. Cyber Defensive Capacity and Capability:: A Perspective from the Financial Sector of a Small State.

Domingue, M.J., Lakhtakia, A., Pulsifer, D.P., Hall, L.P., Badding, J.V., Bischof, J.L., Martín-Palma, R.J., Imrei, Z., Janik, G., Mastro, V.C., et al., 2014. Bioreplicated visual features of nanofabricated buprestid beetle decoys evoke stereotypical male mating flights. Proc. Natl. Acad. Sci. 111 (39), 14 106–14 111.

Doubleday, H., Maglaras, L., Janicke, H., 2016. Ssh Honeypot: Building, Deploying and Analysis.

Dowling, S., Schukat, M., Barrett, E., 2018. Using reinforcement learning to conceal honeypot functionality. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 341–355.

Drew, S.K., Heinen, C.W., 2022. Testing deception with a commercial tool simulating cyberspace. Ph.D. dissertation, Monterey, CA; Naval Postgraduate School.

Durkota, K., Lisỳ, V., Bošanskỳ, B., Kiekintveld, C., 2015a. Optimal network security hardening using attack graph games. In: Twenty-Fourth International Joint Conference on Artificial Intelligence.

Durkota, K., Lisỳ, V., Bošanskỳ, B., Kiekintveld, C., 2015b. Approximate solutions for attack graph games with imperfect information. In: International Conference on Decision and Game Theory for Security. Springer, pp. 228–249.

Erguler, I., 2016. Achieving flatness: selecting the honeywords from existing user passwords. IEEE Trans. Dependable Secure Comput. 13 (2), 284–295.

Eriksson, O., 2023. An Evaluation of Honeypots with Compliant Kubernetes.

Fan, W., Fernández, D., 2017. A novel sdn based stealthy tcp connection handover mechanism for hybrid honeypot systems. In: 2017 IEEE Conference on Network Softwarization (NetSoft). IEEE, pp. 1–9.

Fan, W., Du, Z., Fernández, D., 2015. Taxonomy of honeynet solutions. In: 2015 SAI Intelligent Systems Conference (IntelliSys). IEEE, pp. 1002–1009.

Fan, W., Du, Z., Fernández, D., Villagra, V.A., 2017a. Enabling an anatomic view to investigate honeypot systems: a survey. IEEE Syst. J. 12 (4), 3906–3919.

Fan, W., Fernández, D., Du, Z., 2017b. Versatile virtual honeynet management framework. IET Inf. Secur. 11 (1), 38–45.

Fan, W., Du, Z., Smith-Creasey, M., Fernandez, D., 2019. Honeydoc: an efficient honeypot architecture enabling all-round design. IEEE J. Sel. Areas Commun. 37 (3), 683–697.

Faveri, C.D., 2022. Modeling Deception for Cyber Security.

Feng, S., Lubis, M., 2022. Defense-in-depth security strategy in log4j vulnerability analysis. In: 2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS). IEEE, pp. 01–04.

Ferguson-Walter, K.J., Major, M.M., Johnson, C.K., Muhleman, D.H., 2021. Examining the efficacy of decoy-based and psychological cyber deception. In: 30th USENIX Security Symposium (USENIX Security 21), pp. 1127–1144.

Ferguson-Walter, K.J., Major, M.M., Johnson, C.K., Johnson, C.J., Scott, D.D., Gutzwiller, R.S., Shade, T., 2023. Cyber expert feedback: experiences, expectations, and opinions about cyber deception. Comput. Secur. 130, 103268.

Ferretti, P., Pogliani, M., Zanero, S., 2019. Characterizing background noise in ics traffic through a set of low interaction honeypots. In: Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy, pp. 51–61.

Fraunholz, D., Schotten, H.D., 2018a. Defending web servers with feints, distraction and obfuscation. In: 2018 International Conference on Computing, Networking and Communications (ICNC). IEEE, pp. 21–25.

Fraunholz, D., Schotten, H.D., 2018b. Strategic defense and attack in deception based network security. In: 2018 International Conference on Information Networking (ICOIN). IEEE, pp. 156–161.

Fraunholz, D., Krohmer, D., Anton, S.D., Schotten, H.D., 2017. Investigation of cyber crime conducted by abusing weak or default passwords with a medium interaction honeypot. In: 2017 International Conference on Cyber Security and Protection of Digital Services (Cyber Security). IEEE, pp. 1–7.

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 preprint. arXiv:1804.06196.

Ganesarathinam, R., Prabakar, M.A., Singaravelu, M., Fernandez, A.L., 2020. A detailed analysis of intruders' activities in the network through the real-time virtual honeynet experimentation. In: Artificial Intelligence and Evolutionary Computations in Engineering Systems. Springer, pp. 39–53.

Garg, N., Grosu, D., 2007. Deception in honeynets: a game-theoretic analysis. In: 2007 IEEE SMC Information Assurance and Security Workshop. IEEE, pp. 107–113.

Gautam, R., Kumar, S., Bhattacharya, J., 2015. Optimized virtual honeynet with implementation of host machine as honeywall. In: 2015 Annual IEEE India Conference (INDICON). IEEE, pp. 1–6.

Gjermundrød, H., Dionysiou, I., 2015. Cloudhoneycy-an integrated honeypot framework for cloud infrastructures. In: 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC). IEEE, pp. 630–635.

Gonzalez, C., Aggarwal, P., Cranford, E.A., Lebiere, C., 2022. Adaptive cyberdefense with deception: a human–ai cognitive approach. In: Cyber Deception: Techniques, Strategies, and Human Aspects, pp. 41–57.

Graham, J., Olson, R., Howard, R., 2016. Cyber Security Essentials. CRC Press.

Guerra Manzanares, A., 2017. Honeyio4: the construction of a virtual, low-interaction iot honeypot. Master's thesis. Universitat Politècnica de Catalunya.

Han, Q., Molinaro, C., Picariello, A., Sperli, G., Subrahmanian, V.S., Xiong, Y., 2021. Generating fake documents using probabilistic logic graphs. IEEE Trans. Dependable Secure Comput.

Han, W., Zhao, Z., Doupé, A., Ahn, G.-J., 2016. Honeymix: toward sdn-based intelligent honeynet. In: Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, pp. 1–6.

Han, X., Kheir, N., Balzarotti, D., 2018. Deception techniques in computer security: a research perspective. ACM Comput. Surv. 51 (4), 1–36.

Hayatle, O., Otrok, H., Youssef, A., 2012. A game theoretic investigation for high interaction honeypots. In: 2012 IEEE International Conference on Communications (ICC). IEEE, pp. 6662–6667.

Heckman, K.E., Stech, F.J., Thomas, R.K., Schmoker, B., Tsow, A.W., 2015. Cyber denial, deception and counter deception. Adv. Inf. Secur. 64.

Hedayati, R., Mostafavi, S., 2021. A lightweight image encryption algorithm for secure communications in multimedia Internet of things. Wirel. Pers. Commun., 1–23.

Hirata, A., Miyamoto, D., Nakayama, M., Esaki, H., 2015. Intercept+: Sdn support for live migration-based honeypots. In: 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS). IEEE, pp. 16–24.

Hobert, K., Lim, C., Budiarto, E., 2023. Enhancing cyber attribution through behavior similarity detection on Linux shell honeypots with att&ck framework. In: 2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs). IEEE, pp. 139–144.

Huang, L., Zhu, Q., 2019. Adaptive honeypot engagement through reinforcement learning of semi-Markov decision processes. In: International Conference on Decision and Game Theory for Security. Springer, pp. 196–216.

Huang, M., Fan, W., Huang, W., Cheng, Y., Xiao, H., 2020. Research on Building Exploitable Vulnerability Database for Cloud-Native App. 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), vol. 1. IEEE, pp. 758–762.

Ibrahim, A., Tariq, U., Ahamed Ahanger, T., Tariq, B., Gebali, F., 2023. Retaliation against ransomware in cloud-enabled pureos system. Mathematics 11 (1), 249.

Ikuomenisan, G., Morgan, Y., 2022. Systematic review of graphical visual methods in honeypot attack data analysis. J. Inf. Secur. 13 (4), 210–243.

Izagirre, M., 2017. Deception Strategies for Web Application Security: Application-Layer Approaches and a Testing Platform.

Ja'fari, F., Mostafavi, S., Mizanian, K., Jafari, E., 2021. An intelligent botnet blocking approach in software defined networks using honeypots. J. Ambient Intell. Humaniz. Comput. 12 (2), 2993–3016.

Javadpour, A., 2019. Improving resources management in network virtualization by utilizing a software-based network. Wirel. Pers. Commun. 106 (2), 505–519.

Javadpour, A., 2020. Providing a way to create balance between reliability and delays in sdn networks by using the appropriate placement of controllers. Wirel. Pers. Commun. 110 (2), 1057–1071.

Javadpour, A., Wang, G., 2022. cTMvSDN: improving resource management using combination of Markov-process and tdma in software-defined networking. J. Supercomput. 78, 3477–3499.

Javadpour, A., Abharian, S.K., Wang, G., 2017. Feature selection and intrusion detection in cloud environment based on machine learning algorithms. In: 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC). IEEE, pp. 1417–1421.

Javadpour, A., Ja'fari, F., Taleb, T., Shojafar, M., 2022b. A cost-effective mtd approach for ddos attacks in software-defined networks. In: GLOBECOM 2022-2022 IEEE Global Communications Conference. IEEE, pp. 4173–4178.

Javadpour, A., Ja'fari, F., Taleb, T., Shojafar, M., Yang, B., 2022a. SCEMA: an SDN-oriented cost-effective edge-based MTD approach. IEEE Trans. Inf. Forensics Secur. 18, 667–682.

Javadpour, A., Ja'fari, F., Taleb, T., Benzaıd, C., 2023b. A mathematical model for analyzing honeynets and their cyber deception techniques. In: 2023 27th International Conference on Engineering of Complex Computer Systems (ICECCS). IEEE Computer Society, pp. 81–88.

Javadpour, A., Ja'fari, F., Taleb, T., Benzaïd, C., 2023c. Reinforcement learning-based slice isolation against ddos attacks in beyond 5g networks. IEEE Trans. Netw. Serv. Manag.

Javadpour, A., Pinto, P., Ja'fari, F., Zhang, W., 2023a. Dmaidps: a distributed multi-agent intrusion detection and prevention system for cloud iot environments. Clust. Comput. 26 (1), 367–384.

Jha, R.K., 2023. An in-depth evaluation of hybrid approaches in soft computing for the identification of social engineering. J. Soft Comput. Paradig. 5 (3), 232–248.

Jiang, X., Hao, Z., Wang, Y., 2010. A malware sample capturing and tracking system. In: 2010 Second World Congress on Software Engineering, vol. 1. IEEE, pp. 69–72.

Jones, A., 2022. Security Posture: A Systematic Review of Cyber Threats and Proactive Security.

Jones, M.J., 2016. Shady trick or legitimate tactic-can law enforcement officials use fictitious social media accounts to interact with suspects. Am. J. Trial Advoc. 40, 69.

Jonsson, D., Marteni, A., 2022. Multi-Factor Authentication Mechanism Based on Browser Fingerprinting and Graphical Honeytokens.

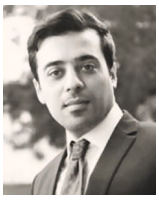
Juels, A., Rivest, R.L., 2013. Honeywords: making password-cracking detectable. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, vol. 11. ACM, pp. 145–160.

Kandanaarachchi, S., Ochiai, H., Rao, A., 2022. Honeyboost: boosting honeypot performance with data fusion and anomaly detection. Expert Syst. Appl. 201, 117073.

Khan, Z.A., Abbasi, U., 2020. Reputation management using honeypots for intrusion detection in the Internet of things. Electronics 9 (3), 415.

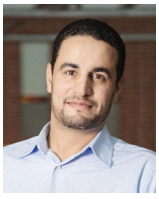
Kheir, N., Abdelrazek, L., Daniel, C., 2022. Demo paper: caught in my radio net-experiment with honeypots in radio access networks. In: 2022 25th Conference on Innovation in Clouds, Internet and Networks (ICIN). IEEE, pp. 1–3.
Khoa, N.H., Do Hoang, H., Ngo-Khanh, K., Duy, P.T., Pham, V.-H., 2023. Sdn-based cyber deception deployment for proactive defense strategy using honey of things and cyber threat intelligence. In: International Conference on Intelligence of Things. Springer, pp. 269–278.
Kiekintveld, C., Lisỳ, V., Píbil, R., 2015. Game-theoretic foundations for the strategic use of honeypots in network security. In: Cyber Warfare. Springer, pp. 81–101.
Kolias, C., Kambourakis, G., Stavrou, A., Voas, J., 2017. Ddos in the iot: Mirai and other botnets. Computer 50 (7), 80–84.
Koziol, J., 2003. Intrusion Detection with Snort. Sams Publishing.
Kreps, D.M., 1989. Nash equilibrium. In: Game Theory. Springer, pp. 167–177.
Kumar, S., Sehgal, R., Bhatia, J., 2012. Hybrid honeypot framework for malware collection and analysis. In: 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS). IEEE, pp. 1–5.
La, Q.D., Quek, T.Q., Lee, J., Jin, S., Zhu, H., 2016. Deceptive attack and defense game in honeypot-enabled networks for the Internet of things. IEEE Int. Things J. 3 (6), 1025–1035.
Lackner, P., 2021. How to Mock a Bear: Honeypot, Honeynet, Honeywall & Honeytoken: A Survey.
Limouchi, E., Mahgoub, I., 2021. Reinforcement learning-assisted threshold optimization for dynamic honeypot adaptation to enhance iobt networks security. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, pp. 1–7.
Luo, T., Xu, Z., Jin, X., Jia, Y., Ouyang, X., 2017. Iotcandyjar: towards an intelligent-interaction honeypot for iot devices. Black Hat, 1–11.
Maesschalck, S., Giotsas, V., Green, B., Race, N., 2022. Don't get stung, cover your ics in honey: how do honeypots fit within industrial control system security. Comput. Secur. 114, 102598.
Marble, J.L., Lawless, W.F., Mittu, R., Coyne, J., Abramson, M., Sibley, C., 2015. The human factor in cybersecurity: robust & intelligent defense. In: Cyber Warfare: Building the Scientific Foundation, pp. 173–206.
McCarthy, A., Ghadafi, E., Andriotis, P., Legg, P., 2022. Functionality-preserving adversarial machine learning for robust classification in cybersecurity and intrusion detection domains: a survey. J. Cybersecur. Priv. 2 (1), 154–190.
McCoy, C.G., 2022. A relevance model for threat-centric ranking of cybersecurity vulnerabilities. Ph.D. dissertation. Old Dominion University.
Mohan, P.V., Dixit, S., Gyaneshwar, A., Chadha, U., Srinivasan, K., Seo, J.T., 2022. Leveraging computational intelligence techniques for defensive deception: a review, recent advances, open problems and future directions. Sensors 22 (6), 2194.
Mokube, I., Adams, M., 2007. Honeypots: concepts, approaches, and challenges. In: Proceedings of the 45th Annual Southeast Regional Conference, pp. 321–326.
Msaad, M., Srinivasa, S., Andersen, M.M., Audran, D.H., Orji, C.U., Vasilomanolakis, E., 2022. Honeysweeper: towards stealthy honeytoken fingerprinting techniques. In: Nordic Conference on Secure IT Systems. Springer, pp. 101–119.
Naeem, N.A., Batchelder, M., Hendren, L., 2007. Metrics for measuring the effectiveness of decompilers and obfuscators. In: 15th IEEE International Conference on Program Comprehension (ICPC'07). IEEE, pp. 253–258.
Naik, N., Shang, C., Jenkins, P., Shen, Q., 2020. D-fri-honeypot: a secure sting operation for hacking the hackers using dynamic fuzzy rule interpolation. IEEE Trans. Emerg. Top. Comput. Intell.
Nazario, J., 2009. Phoneyc: a virtual client honeypot. LEET 9, 911–919.
Nelson, B.A., Wilson, J.O., Rosen, D., Yen, J., 2009. Refined metrics for measuring ideation effectiveness. Des. Stud. 30 (6), 737–743.
Om Kumar, C., Sathia Bhama, P.R., 2019. Detecting and confronting flash attacks from iot botnets. J. Supercomput. 75, 8312–8338.
Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., Rossow, C., 2016. Iotpot: a novel honeypot for revealing current iot threats. J. Inf. Process. 24 (3), 522–533.
Panda, S., Rass, S., Moschoyiannis, S., Liang, K., Loukas, G., Panaousis, E., 2022. Honeycar: a framework to configure honeypot vulnerabilities on the Internet of vehicles. IEEE Access 10, 104 671–104 685.
Papaspirou, V., Maglaras, L., Ferrag, M.A., Kantzavelou, I., Janicke, H., Douligeris, C., 2021. A novel two-factor honeytoken authentication mechanism. In: 2021 International Conference on Computer Communications and Networks (ICCCN). IEEE, pp. 1–7.
Park, B., Dang, S.P., Noh, S., Yi, J., Park, M., 2019. Dynamic virtual network honeypot. In: 2019 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, pp. 375–377.
Park, Y., Stolfo, S.J., 2012. Software decoys for insider threat. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, pp. 93–94.
Pashaei, A., Akbari, M.E., Lighvan, M.Z., Charmin, A., 2022. Early intrusion detection system using honeypot for industrial control networks. Results Eng. 16, 100576.
Pauna, A., Iacob, A.-C., Bica, I., 2018. Qrassh-a self-adaptive ssh honeypot driven by q-learning. In: 2018 International Conference on Communications (COMM). IEEE, pp. 441–446.
Pawlick, J., Zhu, Q., 2015. Deception by design: evidence-based signaling games for network defense. arXiv preprint. arXiv:1503.05458.
Pawlick, J., Colbert, E., Zhu, Q., 2018. Modeling and analysis of leaky deception using signaling games with evidence. IEEE Trans. Inf. Forensics Secur. 14 (7), 1871–1886.
Pawlick, J., Colbert, E., Zhu, Q., 2019. A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy. ACM Comput. Surv. 52 (4), 1–28.
Pawlick, J., Zhu, Q., et al., 2021. Game Theory for Cyber Deception. Springer.
Perevozchikov, V.A., Shaymardanov, T.A., Chugunkov, I.V., 2017. New techniques of malware detection using ftp honeypot systems. In: 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). IEEE, pp. 204–207.
Píbil, R., Lisỳ, V., Kiekintveld, C., Bošanskỳ, B., Pěchouček, M., 2012. Game theoretic model of strategic honeypot selection in computer networks. In: International Conference on Decision and Game Theory for Security. Springer, pp. 201–220.
Popli, N.K., Girdhar, A., 2019. Behavioural analysis of recent ransomwares and prediction of future attacks by polymorphic and metamorphic ransomware. In: Computational Intelligence: Theories, Applications and Future Directions-Volume II. Springer, pp. 65–80.
Priya, V.D., Chakkaravarthy, S.S., 2023. Containerized cloud-based honeypot deception for tracking attackers. Sci. Rep. 13 (1), 1437.
Qin, X., Jiang, F., Cen, M., Doss, R., 2023. Hybrid cyber defense strategies using honey-x: a survey. Comput. Netw., 109776.
Raharjo, D.H.K., Nurmala, A., Pambudi, R.D., Sari, R.F., 2022. Performance evaluation of intrusion detection system performance for traffic anomaly detection based on active ip reputation rules. In: 2022 3rd International Conference on Electrical Engineering and Informatics (ICon EEI). IEEE, pp. 75–79.
Rahmatullah, D.K., Nasution, S.M., Azmi, F., 2016. Implementation of low interaction web server honeypot using cubieboard. In: 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC). IEEE, pp. 127–131.
Razali, M.F., Razali, M.N., Mansor, F.Z., Muruti, G., Jamil, N., 2018. Iot honeypot: a review from researcher's perspective. In: 2018 IEEE Conference on Application, Information and Network Security (AINS). IEEE, pp. 93–98.
Ren, J., Zhang, C., 2020. A differential game method against attacks in heterogeneous honeynet. Comput. Secur. 97, 101870.
Ren, J., Zhang, C., Hao, Q., 2021. A theoretical method to evaluate honeynet potency. Future Gener. Comput. Syst. 116, 76–85.
Rich, M.S., 2023. A Longitudinal Analysis of Cyber Adversarial Tactics and Techniques.
Riggs, H., Tufail, S., Parvez, I., Tariq, M., Khan, M.A., Amir, A., Vuda, K.V., Sarwat, A.I., 2023. Impact, vulnerabilities, and mitigation strategies for cyber-secure critical infrastructure. Sensors 23 (8), 4060.
Rossotti, A., 2022. Anomaly Detection Framework and Deep Learning Techniques for Zero-Day Attack in Container Based Environment.
Rowe, N.C., 2006. Measuring the effectiveness of honeypot counter-counterdeception. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), vol. 6. IEEE, pp. 129c–129c.
Rowe, N.C., Custy, E.J., Duong, B.T., 2007. Defending cyberspace with fake honeypots. J. Comput. 2 (2), 25–36.
Sahin, M., Hébert, C., Cabrera Lozoya, R., 2022. An approach to generate realistic http parameters for application layer deception. In: International Conference on Applied Cryptography and Network Security. Springer, pp. 337–355.
Salimova, H.R., 2022. A virtual honeypot framework. Cent. Asian Res. J. Interdiscip. Stud. 2 (5), 479–486.
Sangaiah, A.K., Javadpour, A., Ja'fari, F., Pinto, P., Zhang, W., Balasubramanian, S., 2023a. A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in cloud of things. Clust. Comput. 26 (1), 599–612.
Sangaiah, A.K., Javadpour, A., Pinto, P., 2023b. Towards data security assessments using an ids security model for cyber-physical smart cities. Inf. Sci., 119530.
Santhosh Kumar, S., Selvi, M., Kannan, A., et al., 2023. A comprehensive survey on machine learning-based intrusion detection systems for secure communication in Internet of things. Comput. Intell. Neurosci. 2023.
Sardana, A., Joshi, R., 2009. An auto-responsive honeypot architecture for dynamic resource allocation and qos adaptation in ddos attacked networks. Comput. Commun. 32 (12), 1384–1399.
Sarr, A.B., Anwar, A.H., Kamhoua, C., Leslie, N., Acosta, J., 2020. Software diversity for cyber deception. In: GLOBECOM 2020-2020 IEEE Global Communications Conference. IEEE, pp. 1–6.
Selvaraj, R., Kuthadi, V.M., Marwala, T., 2016. Honey pot: a major technique for intrusion detection. In: Proceedings of the Second International Conference on Computer and Communication Technologies. Springer, pp. 73–82.
Sethuraman, S.C., Jadapalli, T.G., Sudhakaran, D.P.V., Mohanty, S.P., 2023. Flow based containerized honeypot approach for network traffic analysis: an empirical study. Comput. Sci. Rev. 50, 100600.
Seungjin, L., Abdullah, A., Jhanjhi, N., 2020. A review on honeypot-based botnet detection models for smart factory. Int. J. Adv. Comput. Sci. Appl. 11 (6), 418–435.
Shabtai, A., Bercovitch, M., Rokach, L., Gal, Y., Elovici, Y., Shmueli, E., 2016. Behavioral study of users when interacting with active honeytokens. ACM Trans. Inf. Syst. Secur. 18 (3), 1–21.
Shakarian, P., Paulo, D., Albanese, M., Jajodia, S., 2014. Keeping intruders at large: a graph-theoretic approach to reducing the probability of successful network intrusions. In: 2014 11th International Conference on Security and Cryptography (SECRYPT). IEEE, pp. 1–12.
Shi, L., Jiang, L., Liu, D., Han, X., 2012. Mimicry honeypot: a brief introduction. In: 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing. IEEE, pp. 1–4.

Shin, B., Lowry, P.B., 2020. A review and theoretical explanation of the 'cyberthreat-intelligence (cti) capability' that needs to be fostered in information security practitioners and how this can be accomplished. Comput. Secur. 92, 101761.
Shumakov, I.U., Troitskiy, S.S., Silnov, D.S., 2017. Increasing the attractiveness of false objects of attack on the web-servers. In: 2017 18th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM). IEEE, pp. 195–198.
Siniosoglou, I., Efstathopoulos, G., Pliatsios, D., Moscholios, I.D., Sarigiannidis, A., Sakellari, G., Loukas, G., Sarigiannidis, P., 2020. Neuralpot: an industrial honeypot implementation based on deep neural networks. In: 2020 IEEE Symposium on Computers and Communications (ISCC). IEEE, pp. 1–7.
Soundararajan, R., Rajagopal, M., Muthuramalingam, A., Hossain, E., Lloret, J., 2022. Interleaved honeypot-framing model with secure mac policies for wireless sensor networks. Sensors 22 (20), 8046.
Srinivasa, S., Pedersen, J.M., Vasilomanolakis, E., 2020. Towards systematic honeytoken fingerprinting. In: 13th International Conference on Security of Information and Networks, pp. 1–5.
Srinivasa, S., Pedersen, J.M., Vasilomanolakis, E., 2022. Interaction matters: a comprehensive analysis and a dataset of hybrid iot/ot honeypots. In: Proceedings of the 38th Annual Computer Security Applications Conference, pp. 742–755.
Steingartner, W., Galinec, D., Kozina, A., 2021. Threat defense: cyber deception approach and education for resilience in hybrid threats model. Symmetry 13 (4), 597.
Sumadi, F.D.S., Widagdo, A.R., Reza, A.F., et al., 2022. Sd-honeypot integration for mitigating ddos attack using machine learning approaches. JOIV: Int. J. Inform. Vis. 6 (1), 39–44.
Sun, J., Sun, K., Li, Q., 2020. Towards a believable decoy system: replaying network activities from real system. In: 2020 IEEE Conference on Communications and Network Security (CNS). IEEE, pp. 1–9.
Sun, R., Yuan, X., Lee, A., Bishop, M., Porter, D.E., Li, X., Gregio, A., Oliveira, D., 2017. The dose makes the poison—leveraging uncertainty for effective malware detection. In: 2017 IEEE Conference on Dependable and Secure Computing. IEEE, pp. 123–130.
Suratkar, S., Shah, K., Sood, A., Loya, A., Bisure, D., Patil, U., Kazi, F., 2021. An adaptive honeypot using q-learning with severity analyzer. J. Ambient Intell. Humaniz. Comput., 1–12.
Suryawanshi, B.D., Tayade, P.B., Patil, A.V., Patil, J.B., Rajput, D.V., 2017. Enhancing security using honeywords. In: International Journal of Innovative Research and Creative Technology, vol. 2. IJIRCT.
Tabari, A.Z., Liu, G., Ou, X., Singhal, A., 2023. Revealing human attacker behaviors using an adaptive Internet of things honeypot ecosystem. In: IFIP International Conference on Digital Forensics. Springer, pp. 73–90.
Tan, R.R., Eng, S., How, K.C., Zhu, Y., Jyh, P.W.H., 2023. Honeypot for cybersecurity threat intelligence. In: IRC-SET 2022: Proceedings of the 8th IRC Conference on Science, Engineering and Technology, August 2022, Singapore. Springer, pp. 587–598.
Tian, D.J., Bates, A., Butler, K., 2015. Defending against malicious usb firmware with goodusb. In: Proceedings of the 31st Annual Computer Security Applications Conference, pp. 261–270.
Toor, J.S., Bhandari, E.A., 2017. Honeypot: a deceptive trap. Int. J. Eng. Technol. Manag. Appl. Sci.
Valero, J.M.J., Pérez, M.G., Celdrán, A.H., Pérez, G.M., 2020. Identification and classification of cyber threats through ssh honeypot systems. In: Handbook of Research on Intrusion Detection Systems. IGI Global, pp. 105–129.
Voris, J.A., Jermyn, J., Keromytis, A.D., Stolfo, S., 2013. Bait and Snitch: Defending Computer Systems with Decoys.
Wagener, G., Dulaunoy, A., Engel, T., et al., 2009. Self adaptive high interaction honeypots driven by game theory. In: Symposium on Self-Stabilizing Systems. Springer, pp. 741–755.
Wang, H., Wu, B., 2019. Sdn-based hybrid honeypot for attack capture. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, pp. 1602–1606.
Wang, H., He, H., Zhang, W., Liu, W., Liu, P., Javadpour, A., 2022. Using honeypots to model botnet attacks on the Internet of medical things. Comput. Electr. Eng. 102, 108212.
Wang, K., Du, M., Maharjan, S., Sun, Y., 2017. Strategic honeypot game model for distributed denial of service attacks in the smart grid. IEEE Trans. Smart Grid 8 (5), 2474–2482.
Wang, M., Santillan, J., Kuipers, F., 2018. Thingpot: an interactive Internet-of-things honeypot. arXiv preprint. arXiv:1807.04114.
Wang, S.-H., 2022. The Observation of Smart Camera Security.
Wegerer, M., Tjoa, S., 2016. Defeating the database adversary using deception - a mysql database honeypot. In: 2016 International Conference on Software Security and Assurance (ICSSA). IEEE, pp. 6–10.
Whaley, B., 1982. Toward a general theory of deception. J. Strateg. Stud. 5 (1), 178–192.
White, J., Park, J.S., Kamhoua, C.A., Kwiat, K.A., 2014. Social network attack simulation with honeytokens. Soc. Netw. Anal. Min. 4, 1–14.
Yamin, M.M., Katt, B., 2022. Use of cyber attack and defense agents in cyber ranges: a case study. Comput. Secur. 122, 102892.
Yang, X., Yuan, J., Yang, H., Kong, Y., Zhang, H., Zhao, J., 2023. A highly interactive honeypot-based approach to network threat management. Future Internet 15 (4), 127.
You, J., Lv, S., Zhao, L., Niu, M., Shi, Z., Sun, L., 2020. A scalable high-interaction physical honeypot framework for programmable logic controller. In: 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall). IEEE, pp. 1–5.
Zarras, A., 2014. The art of false alarms in the game of deception: leveraging fake honeypots for enhanced security. In: 2014 International Carnahan Conference on Security Technology (ICCST). IEEE, pp. 1–6.
Zhang, L., Thing, V.L., 2021. Three decades of deception techniques in active cyber defense-retrospect and outlook. Comput. Secur. 106, 102288.
Zhu, M., Anwar, A.H., Wan, Z., Cho, J.-H., Kamhoua, C.A., Singh, M.P., 2021. A survey of defensive deception: approaches using game theory and machine learning. IEEE Commun. Surv. Tutor. 23 (4), 2460–2493.
Zhuge, J., Holz, T., Han, X., Guo, J., Zou, W., 2007. Characterizing the irc-based botnet phenomenon. China Honeynet Technical Report.
Zobal, L., Kolář, D., Fujdiak, R., 2019. Current state of honeypots and deception strategies in cybersecurity. In: 2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). IEEE, pp. 1–9.



Amir Javadpour obtained his MSc degree in Medical Information Technology Engineering from University of Tehran, Iran, in 2014. He received a PhD in Computer Science/Mathematics/Cybersecurity from Guangzhou University, China. In addition, he has published papers with his colleagues in highly ranked journals and several ranked conferences on several topics, including Cloud Computing, Software-Defined Networking (SDN), Big Data, Intrusion Detection Systems (IDS), and the Internet of Things (IoT), Moving Target Defence (MTD), Machine Learning (ML) and optimization algorithms. Additionally, he reviewed papers for several reputable venues such as IEEE Transactions on Cloud Computing, IEEE Transactions on Network Science and Engineering, ACM Transactions on Internet Technology, the Journal of Supercomputing, several journals of Springer and Elsevier, etc. He is a Technical Program Committee (TCP) Member of various conferences.

Forough Ja'fari is a Senior Researcher in cybersecurity and computer science. She received her Bachelor's degree from Sharif University of Technology and her Master's degree in Computer Network Engineering from Yazd University, Iran. She is a visiting scholar researcher at Guangzhou University, China. Cloud computing, software-defined Networking (SDN), cyber deception, Intrusion Detection Systems (IDS), Internet of Things (IoT), Moving Target Defence (MTD), and Machine Learning are some of her research interests. She is currently a Guest Editor (GE) of Cluster Computing (CLUS) Journal and a reviewer for several journals and conferences.



Tarik Taleb Prof. Tarik Taleb is currently a full Professor at the Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, Germany, and a professor at the Center of Wireless Communications, The University of Oulu, Finland. He is the founder and director of the MOSA!C Lab (www.mosaic-lab.org). Between Oct. 2014 and Dec. 2021, he was an Associate Professor at the School of Electrical Engineering, Aalto University, Finland. Prior to that, he was working as a Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd, Heidelberg, Germany. Before joining NEC and till Mar. 2009, he worked as an assistant professor at the Graduate School of Information Sciences, Tohoku University, Japan, in a lab fully funded by KDDI, the second-largest mobile operator in Japan. From Oct. 2005 till Mar. 2006, he worked as a research fellow at the Intelligent Cosmos Research Institute, Sendai, Japan. He received his B. E degree in Information Engineering with distinction, M.Sc. and Ph.D. degrees in Information Sciences from Tohoku Univ., in 2001, 2003, and 2005, respectively. Prof. Taleb's research interests lie in the field of telco cloud, network softwarization and network slicing, AI-based software-defined security, immersive communications, mobile multimedia streaming, and next-generation mobile networking. Prof. Taleb has also been directly engaged in developing and standardizing the Evolved Packet System as a member of 3GPP's System Architecture working group 2. Prof. Taleb served on the IEEE Communications Society Standardization Program Development Board. Prof. Taleb served as the general chair of the 2019 edition of the IEEE Wireless Communications and Networking Conference (WCNC'19) held in Marrakech, Morocco. He was the guest editor-in-chief of the IEEE JSAC Series on Network Softwarization and Enablers. He was on the editorial board of different IEEE journals and magazines. Till Dec. 2016, he served as chair of the Wireless Communications Technical Committee, the largest in IEEE ComSoC. Prof. Taleb is the recipient of the 2021 IEEE ComSoc Wireless Communications Technical Committee Recognition Award (Dec. 2021), the 2017 IEEE ComSoc Communications Software Technical Achievement Award (Dec. 2017) for his outstanding contributions to network softwarization. He is also the (co-) recipient of the 2017 IEEE Communications Society Fred W. Ellersick Prize (May 2017), the 2009 IEEE ComSoc Asia-Pacific Best Young Researcher award (Jun. 2009), the 2008 TELECOM System Technology Award from the Telecommunications Advancement Foundation (Mar. 2008), the 2007 Funai Foundation Science Promotion Award (Apr. 2007), the 2006 IEEE Computer Society Japan Chapter Young Author Award (Dec. 2006), the Niwa Yasujirou Memorial Award (Feb. 2005), and the Young Researcher's Encouragement Award from the Japan chapter of the IEEE Vehicular Technology Society (VTS) (Oct. 2003). Some of Prof. Taleb's research work has been awarded best papers at prestigious IEEE-flagged conferences.

Chafika Benzaïd is currently a senior research fellow at University of Oulu, Finland. Between Nov. 2018 and Dec. 2021, she was senior researcher at Aalto University. Before that, she worked as an associate professor at University of Sciences and Technology Houari Boumediene (USTHB). She holds Engineer, Magister and "Doctorat ès Sciences" degrees from USTHB. Her research interests lie in the field of 5G/6G, SDN, Network Security, AI Security, and AI/ML for zero-touch security management. She is an ACM professional member.

Mohammad Shojafar (Senior Member, IEEE) is a Senior Lecturer (Associate Professor) in network security and an Intel Innovator, a Professional ACM member and ACM Distinguished Speaker, a Fellow of the Higher Education Academy, and a Marie Curie Alumni, working in the 5G & 6G Innovation Centre (5GIC & 6GIC), Institute for Communication Systems (ICS), at the University of Surrey, UK. Before, he was a Senior Researcher and a Marie Curie Fellow in the SPRITZ Security and Privacy Research group at the University of Padua, Italy. Dr Mohammad secured £310k for the ESKMARALD project funded by GCHQ, UK, in 2022. Also, he is a PI of AUTOTRUST, a secure autonomous 5G-based traffic management platform the European Space Agency supported for around €750k in 2021. Also, Mohammad was a PI of the PRISENODE project, a €275k Horizon 2020 Marie Curie project in network security and Fog task/resource scheduling collaborating at the University of Padua. He also was a PI on an Italian SDN security and privacy (€60k) supported by the University of Padua in 2018 and a Co-PI on an Ecuadorian-British project on IoT and Industry 4.0 resource allocation ($20k) in 2020. He contributed to some Italian projects in telecommunications, like GAUChO, SAMMClouds, and SC2. He received his PhD in ICT from Sapienza University of Rome, Rome, Italy, in 2016 with an "Excellent" degree. He is an Associate Editor in *IEEE Transactions on Network and Service Management*, *IEEE Transactions on Intelligent Transportation Systems*, *IEEE Consumer Electronics Magazine*, *IEEE Systems Journal* and Computer Networks.