

Honeypot and cyber deception as a tool for detecting cyber attacks on critical infrastructure

Dmytro S. Morozov¹, Tetiana A. Vakaliuk^{1,2,3,4}, Andrii A. Yefimenko¹,
Tetiana M. Nikitchuk¹ and Roman O. Kolomiets¹

¹Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine

²Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine

³Institute for Digitalisation of Education of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine

⁴Academy of Cognitive and Natural Sciences, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine

Abstract

The constant growth of the threat of cyber attacks on Ukraine's critical infrastructure and industrial IoT networks requires the search for an effective solution to detect and respond to such threats. Ukrainian networks have already become a testing ground for new tactics, methods, and tools for cyber attacks. The study of these attacks, their detailed analysis, and analysis will allow a better understanding of the tools and methods of Russian hackers. Modern approaches to building honeypot/honeynet networks, as well as cyber deception platforms, can be used as an effective source of such information. However, there is no universal solution for such systems, and their effectiveness directly depends on the qualifications of the specialists who deploy them and a deep understanding of their capabilities. The correct use of highly interactive honeypot systems and deception platforms allows you to build a believable honeypot system that will collect information about both the fact of the attack and the actions of the attackers. The analysis of this information will be able to improve both the level of network security and become a source of evidence for further prosecution of cybercriminals. The article presents an overview of the features of using honeypot/honeynet solutions and cyber deception for general-purpose networks and industrial IoT networks.

Keywords

IoT honeypot, cyber security, honeypot, honeynet, cyber deception, security deception

1. Introduction

The number of threats in the field of cyber security has a steady upward trend, and 2022 was no exception. It will be remembered as another year of ransomware attacks, data breaches, attacks on critical infrastructure, and most importantly, a year of global cyber security impact due to Russia's invasion of Ukraine.

doors-2023: 3rd Edge Computing Workshop, April 7, 2023, Zhytomyr, Ukraine


✉ morozovds@ztu.edu.ua (D. S. Morozov); tetianavakaliuk@acnsi.org (T. A. Vakaliuk);
yefimenko.andrii@gmail.com (A. A. Yefimenko); tnitchuk@ukr.net (T. M. Nikitchuk); krt_kro@ztu.edu.ua
(R. O. Kolomiets)

🌐 <https://acnsi.org/vakaliuk/> (T. A. Vakaliuk); <https://ztu.edu.ua/teacher/319.html> (A. A. Yefimenko);
<https://ztu.edu.ua/teacher/132.html> (T. M. Nikitchuk)

🆔 0000-0002-0807-590X (D. S. Morozov); 0000-0001-6825-4697 (T. A. Vakaliuk); 0000-0003-2128-4797
(A. A. Yefimenko); 0000-0002-9068-931X (T. M. Nikitchuk); 0000-0002-9020-938X (R. O. Kolomiets)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

On February 24, 2022, the world of cyber security entered the era of hybrid warfare. Hours before the missiles were launched and the aggressor convoys crossed the border, Russian hackers launched a massively destructive cyber attack against the Ukrainian government, technology companies, and the financial sector. It should be noted that the very beginning of the war in cyberspace as a component of the war on the ground caused significant changes in both the number and the direction of the attacks caused by the war. Along with a significant number of directly or indirectly state-sponsored terrorist groups (APTs), a significant number of threats have emerged as a result of patriotic hacktivism. An example of such activity is the significant surge of DDOS attacks in 2022, “defaces”, sporadic and poorly coordinated attacks on administrative institutions, which mostly had a psychological impact and tried more to cause chaos in society than to cause direct economic or military damage [1].

Such challenges made it necessary to make rapid changes in the work of both state and private institutions in the field of cyber security and to look for effective ways to counter new threats. It is the speed of adaptation of new approaches and the implementation of the best global practices that in many ways made it possible to avoid greater losses and devastating consequences of cyberattacks. The analysis, study, and systematization of information about the algorithms, methods, and technologies used in these attacks and will, with a high probability, be used again – is an important factor in building a flexible and adaptive cyber security system both at the state level and at the level of individual enterprises and organizations

2. Theoretical background

The toolkit and ways cybercriminals penetrate the network are very broad and constantly evolving [2]. That is why modern studies of deception technologies and honeypots of different levels of interactivity are distinguished by a variety of approaches to the construction of research networks with artificial security vulnerabilities to collect information in “field conditions”. The works of Fraunholz et al. [3], Fraunholz and Schotten [4] describe some honeypot studies in university networks. In these articles, the authors analyzed the behavior of both real attackers and network vulnerability testers in a pre-configured vulnerable system. The researchers provided them with an attack system that used honeypot resources and monitored their behavior. These were server-side honeypots: a fake robots.txt file, modified error messages, adaptive latency, and various honey-files. More than a thousand visits by attackers were checked. The attackers’ behavior was further analyzed over six months using a series of honeypots deployed on one client and five web hosting servers. More than ten million visits have been tracked. HoneytokenTP, honey-tokenTPS, FTP, POP3, SMTP, SSH, and telnet protocols were used in honeypot objects. Industrial communication protocols were also simulated to investigate threats to industrial applications.

Cybercriminals are constantly looking for ways to detect the use of deception platforms and honeypots in the network they attack. This causes the need for constant improvement of such systems and increases their plausibility. In the work of Reti et al. [5], the concept of interference honeypot elements is introduced as a plausible extension of existing deception structures, directing attackers to attack honeypot elements. Their models and reference implementations are offered. Behavioral patterns of criminals when interacting with a new type of bait are

analyzed. The advantage of the proposed solutions is to increase the interaction between the attacker and the deployed honeypot elements, which increases the probability of causing the attacker insecurity while losing the attacker's time and resources. The proposed system is capable of improving the intrusion detection process, as well as delaying and hindering current intelligence activities.

In recent years, the problems of building highly interactive honeypot systems for the Internet of Things have attracted the increased interest of researchers, since the security problem in IoT and the improvement of the tools of criminals do not allow the use of traditional approaches of general purpose networks. Fraunholz et al. [6] discusses Falcom, a high-interaction honeypot that provides a full-fledged operating system that maximizes its interaction with attackers and is designed for embedded architectures. Any interaction with this honeypot is suspicious and will be referred for further investigation. Analyzing observed attack parameters can reveal recent trends, new attack vectors, and current intrusion attempts. The paper considers the features of building honeypots for embedded systems, processor architecture, as well as system resources that are chosen for a plausible simulation of embedded devices. In the reference implementation, an authentication mechanism prone to brute-force attacks and dictionary attacks is investigated.

One of the main tasks of deception platforms and honeypots is to collect the evidence base for investigating cyber incidents and subsequently bringing cybercriminals to justice. A honeypot system of medium interaction, which offers telnet and SSH services, is considered by Fraunholz et al. [7]. This honeypot was used to collect information about interactions with them for three months. These data were used for statistical and behavioral analysis. The distribution of attacks and different attacker IP addresses, countries of origin, anonymization services used, adversary skill level, and embedded devices commonly targeted were analyzed. The work uses machine learning methods that can identify unique types of sessions based on issued commands and provided credentials. The collected data were analyzed for characteristics that allowed the classification of types of attackers and sessions.

Differences in the architecture and resources of industrial Internet of Things networks for various purposes lead to the need to find the right honeypot deployment methods for maximum efficiency of their use. Various available honeypot systems for industrial IoT systems and methods of their deployment are considered by Acien et al. [8]. At the same time, the search systems used by criminals to search and identify PLC and SCADA systems and their vulnerabilities at the stage of attack preparation were analyzed. Methods of deploying bait systems using cloud technologies have been studied. The popular ELK stack (ElasticSearch, Logstash, and Kibana) is used as a system for collecting information about interactions with baits. The article demonstrates the technique of deploying a honeypot by conducting a proof-of-concept based on attacks in a controlled environment.

Even though the telnet protocol is quite old, it is still widely used in IoT networks and is often the target of malicious attacks. Šemić and Mrdović [9] investigates the implementation of a honeypot that detects and reports telnet attacks on IoT devices. The considered honeypot allows the detection of both malicious attacks and attacks based on the Mirai botnet. The multi-component design is implemented to achieve sufficient exposure to opposing traffic and security of collected data. The paper explores a flexible honeypot design that allows the honeypot to be easily modified to emulate different IoT devices.

One potentially dangerous area of attack is attackers' attacks on IoT devices that use the

Universal Plug and Play protocol. The U-PoT framework for building a honeypot for Internet of Things devices is proposed and investigated by Hakim et al. [10]. The proposed framework automatically creates a honeypot from UPnP device description documents and can be extended to any type of device or provider that uses UPnP to communicate. Experimental studies have shown that emulated devices can mimic the behavior of an actual IoT device and fool vendor-provided device management programs or popular IoT search engines used by criminals to find vulnerable devices.

That is why the purpose of this study is to investigate the possibility of using honeypot and deception platforms in the networks of critical infrastructure enterprises to increase awareness of possible cyber incidents and minimize damage from the activities of attackers.

3. Results

The directions of attacks, penetration methods, and tools of cybercriminals are constantly evolving rapidly. Along with the threats of attacks on government institutions, commercial enterprises, and user data, we should not forget the global trends in the number of attacks on IoT devices and various embedded systems. Although this direction of cyberattacks is not as threatening in the short term as attacks on critical infrastructure, however, given the increasingly widespread use of the Internet of Things in medicine, industry, and utility infrastructure, it is quite promising for APT groups in terms of the ratio of efforts to implement such attacks to the chaos and damage caused by these attacks.

This is a global trend that has been forming in the last decade. Most cybersecurity threat reviews and studies note that almost every organization will soon face an IoT cybersecurity challenge either directly on their corporate network or through a third party in their supply chain—if they haven't already.

IoT devices are subject to an average of 5,200 cyberattacks per month [11]. Analysts predict that by 2023, there will be 27 to more than 50 billion connected devices, from laptops and medical devices to smart locks, smart appliances, and smart thermostats. Because these devices typically have limited computing power and often lack built-in protections, they are particularly vulnerable to hacker attacks trying to gain access to the network. As the Internet of Things rapidly expands into personal and professional life, the potential attack surface becomes ever larger.

Microsoft's annual report also noted an alarmingly growing list of Internet-connected and Internet of Things devices that are becoming favorite targets for hackers due to the lack of built-in security controls [12]. CommonSpirit Health cybersecurity incident forces IT systems to go offline. According to the report, attacks on remote control devices have increased steadily since June 2021. The nature and direction of these attacks are constantly evolving. Last year saw a significant drop in attacks against common IoT protocols such as telnet, in some cases by 60%. At the same time, botnets have been repurposed by cybercriminal groups and nation-state actors. At the same time, some threats have remained stably high for several years. The persistence of malware such as Mirai highlights the modularity of these attacks and the adaptability of security measures. Mirai, which has been redesigned several times to adapt to different architectures, has infected a wide range of IoT devices, including Internet Protocol cameras, digital security

camera DVRs, and routers, according to Microsoft's Digital Defense Report. The attack vector has bypassed legacy security controls and poses a risk to network endpoints by exploiting additional vulnerabilities and lateral movement.

Such threats require special attention to the protection of networks of enterprises and organizations that use IoT devices in their activities. Software and architectural vulnerabilities of technologies, the complexity of controlling the security level of an IoT network built on devices from different vendors, and most importantly, the adaptability and constant improvement of attackers' attacks on such networks – all these forces us to look for new and effective ways and tools to protect IoT networks.

Systemic problems with the security of Internet of Things devices and the networks that use them lead to an increase in the attack surface of such a network and difficulties in its control by traditional IDS systems. The use of modern approaches to the construction of honeypot networks and deception systems, as their evolutionary offspring, is an effective tool for strengthening control over the actions of attackers at the stage of preparing and conducting an attack on IoT networks.

3.1. Using a honeypot as an attack detection tool

Historically, honeypot systems were designed to find and study the actions of attackers in a compromised system. The term honeypot is used for a system that has been configured to be compromised. Usually, it contains older and vulnerable software with vulnerabilities or security holes related to improper configuration of the program. Due to its location within the DMZ and in the middle of the enterprise network, it should serve as a high-priority target and provide information about the attacker's methods and tools. The honeypot system makes it possible to reduce the number of false positives issued by IDS/IPS systems. Honeypots can be easily used to identify and systematize information about new attack methods and improve the information system about prospective threats [13]. In the early stages, the attacker scans the network for vulnerable computers, then discovers a honeypot that is deliberately vulnerable to attract attacks. If an attacker tries to connect to the honeypot in the future, the system will immediately detect and record the action, because a normal user does not have to interact with the system [14].

The main classification feature of honeypot systems is the degree of their interactivity. Interactivity refers to the level of open network services available to an attacker. Honeypot systems are low-interactive and highly interactive.

A low-interactivity honeypot, as a rule, includes one or more network services that are the objects of an attack. These services are POP3, SMTP, IMAP, FTP, HTTP, and others. Such a honeypot is installed on a computer running MS Windows or GNU/Linux as a regular service. This service immediately secures ports for listening to network activity. The number of open ports is determined by the number of emulated services. In most cases, emulation of services occurs at the surface level – programs do not implement all RFC requirements but only imitate the most frequently called commands [15]. 10-15 years ago this was considered sufficient, but now it becomes one of the main reasons for possible exposure.

Advantages of low-interactive honeypot:

- Relative simplicity of implementation.

- Ease of installation and maintenance.
- Ease of setup.
- Works on top of the standard operating system.
- Many baits scattered across the network can be combined into a system.

Disadvantages of low-interactivity honeypots include:

- A limited number of emulated services.
- Low stealth from detection.
- Low (compared to highly interactive honeypot) efficiency in tracking the attacker's actions.

A highly interactive honeypot is a software package designed to emulate the entire operating system. Unlike a low-interactive honeypot, a highly interactive honeypot allows you to convince a hacker that he is on a compromised machine, uses the command line or a graphical interface, and executes commands on it. Such a system looks much more realistic than a simple emulation of individual services – the attacker realizes that he has partially achieved his goal – one of the network's computers is already hacked. If before that, the main information collected about the hacker was mainly in the protocols of network activity sessions, now the hacker performs all his actions on the honeypot, which allows him to log his activities also at the system level, either using the operating system or by separate programs, which collect all information about his actions.

The functions of highly interactive systems are much wider than low interactive ones:

- Data collection and control (listening to network traffic and keeping logs for further analysis).
- Detection of attacks and their attack sources.
- Identification of the intruder and information about him (IP address, data transfer protocol, port, country, User agent, operating system).
- Control and logging of the attacker's actions.
- Responding to the attacker's actions, in particular, blocking his activity.
- Misleading the attacker by hiding or changing the information, by which he can understand that he is not attacking the real system, but the honeypot, as well as by changing the system configuration.

The advantages of a highly interactive honeypot include:

- Maximum information about the attacker's actions.
- It is more difficult for an attacker to distinguish a highly interactive honeypot from an ordinary node.
- Ability to install any programs containing real vulnerabilities.
- Ability to detect previously unknown system vulnerabilities.

Disadvantages of a highly interactive honeypot include:

- Necessity of deployment by a qualified team of specialists.

- Data analysis problem after honeypot hack.
- Presence of unmasking signs. If an attacker can determine by any means that the system is highly interactive and not real, such a system ceases to be resistant to detection.
- The possibility of an attacker using the system as a hacking tool.

After deployment, highly interactive honeypots require a lot of attention and qualification from the specialists who use them. These people must ensure the quality of system maintenance and ensure that honeypots are not used to attack real systems during capture.

To create a picture of the attacker, it is necessary to determine the information that will be collected by the highly interactive system. A description of an attack on a highly interactive system usually contains information according to the following criteria:

- scale and depth – the scale of the attack is described by the number of compromised machines, and the depth is the level of impact on the system;
- complexity – characterizes the level of knowledge required to execute a specific attack;
- masking – the quality of hiding traces of one's presence in the system by an attacker;
- the source of the attack – the attacker should be identified as much as possible;
- a vulnerability is a flaw in the system/protocol that allows an attack to be carried out;
- tools – tools used in the attack, such as rootkits or backdoors;
- scale and depth can be derived from the frequency of attacks, the degree of impact of the attack, and the degree of infection of the system.

The masking of an attacker is determined by how well he hides the traces of his presence in the system. The vulnerability used by the attacker must be identified for further statistics. This is necessary because usually in a highly interactive system, there are several vulnerabilities at once, and statistics are collected for each of them. In addition, one attacker can simultaneously attack several vulnerabilities and not all of his attack attempts will be successful. The source of the attack can usually be determined using the metadata of network packets, but the source of the attack can be difficult to identify because the attacker will try to hide his presence on the system.

However, most open-source implementations of both low and high-interactive honeypots have long been known to be experienced, attackers. Methods of identification and bypassing allow you to detect such honeypots at the stage of scanning and inspecting the system and not fall into the set traps. Such detection methods include measuring round-trip time, sending damaged packets and analyzing responses to them, researching the completeness of service functionality, anomalies in the behavior of system calls, network traffic analysis, determining hardware anomalies, and others [16].

The main general disadvantage of honeypot systems, which are revealed by attackers to penetrate the network and search for vulnerable systems for further lateral movement, is the lack of plausible network traffic from bait systems. One honeypot that has vulnerabilities, and open ports, but does not interact with the rest of the network, allows you to quickly identify it as a trap. It is precise to reduce this unmasking feature that individual honeypots are combined into networks called honeynets.

3.2. Honeynet as a further evolutionary development of honeypot

The goal of honeynet technology is to simulate a real network as realistically as possible, including production systems, servers, services, etc. [17]. The degree of success of a honeynet lies in the ability to track all the movements and actions of an attacker on the network, rather than on an individual host. All traces left by cyber attackers as a result of their actions and use of tools are analyzed and monitored to be able to know what tactics are used and what is the ultimate goal of the attackers. However, even here criminals do not stand still and their arsenal has evolved following the tools of cyber security specialists. Tools have been created that can identify some networks that use honeypots, such as Shodan's "Honeypot Or Not?" [18].

By creating a dedicated segment, the working network is isolated from the honeynet. This setup allows you to deploy low-interactivity and high-interactivity honeypots to track hackers across multiple systems or on a single host, such as a t-pot [19]. Creating a separate honeynet allows the threat analysis team and security experts to collect data about the network activity of attackers, giving them an attractive target. Honeynet can contain known vulnerabilities, various operating systems, information systems, servers, and much more [20]. Having multiple honeypot instances allows an attacker to advance across a network segment and leave behind more evidence, such as Tactics, Methods, and Procedures (TTP), Indicator of Compromise (IoC), and Indicator of Attack (IoA). By deploying and configuring the honeynet, specialists force the attacker to move in the direction they planned, slowing down and to some extent controlling the speed of the attacker's lateral movement in the middle of the network. At the same time, his actions are recorded and the necessary information is prepared, which will allow the law enforcement officers to identify the perpetrator in the future and, using the collected evidence base, bring him to justice.

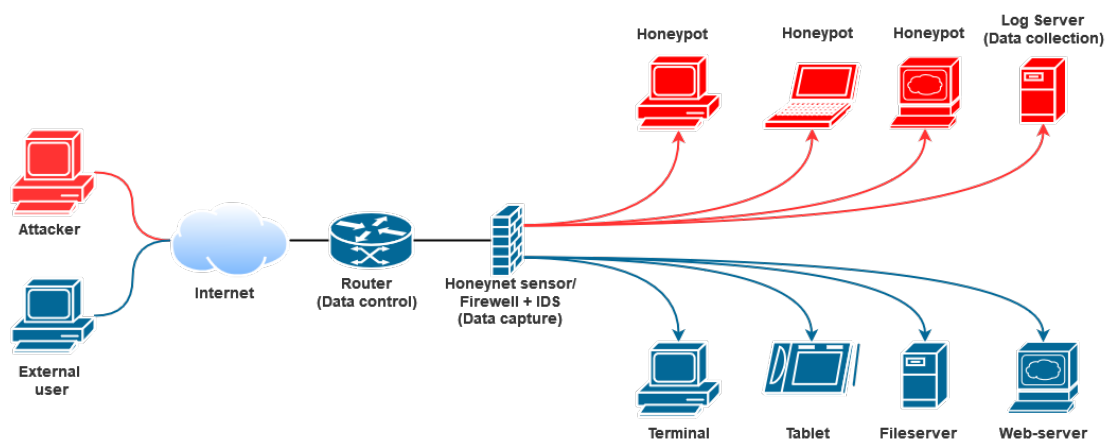


Figure 1: Honeynet architecture.

Collected data can help improve the security of the network, and computer systems, reduce risk, and better protect the organization. A honeynet provides additional protection than classically building walls around a network with a firewall and IDS, as each honeypot will collect and harvest network traffic, IP addresses, zero-day exploits, and other information that

can be used to improve network security.

However, along with the additional advantages of honeynet, some problems complicate their successful implementation. First, honeypots and honeynets can increase the attack surface, which will require more careful monitoring of the state of network security. Second, if misconfigured, the honeypot and honeynet can themselves be exploited, providing attackers with access to the work network. Thirdly, the creation of a honeypot or honeynet, their configuration and maintenance may require highly qualified specialists. With this in mind, many choose open-source solutions. However, while open-source honeynets also have a lower risk of security vulnerabilities because anyone can inspect the code for potential problems, their specifics are well-known to attackers. For example, when deploying a honeynet based on the popular t-pot solution, a specialist should remember that by default all ports will be left open. An error in system configuration makes such a honeypot suspiciously vulnerable and will allow an attacker to quickly determine the presence of a trap.

Another serious debunking feature is the functional segmentation of the honeynet from the enterprise network, which with some time spent by the attacker on analyzing the traffic in the middle of the network, can allow to identify the honeynet and bypass these traps. The solution to this problem is the use of deception solutions to make it difficult for an attacker to detect the very fact of presence of a honeypot or an entire honeynet in the network.

3.3. Advantages of cyber deception over honeypot

Cyber deception (security deception) is a technique used to consistently deceive an attacker during a cyber attack [21]. Deception and honeypot are very related, but not the same thing. Early honeypots were designed to create vulnerable hosts or network segments for potential attackers in predetermined areas of the network, such as the DMZ. Their goal was to reveal the fact of preparation or initiation of an attack. At the same time, a cyber deception is a holistic approach to constantly deceiving an attacker before and during a cyber attack. This can be done using techniques of manipulation, lies, and false information. In addition, deception can be used both at the network boundary and within the network to detect lateral movement [22].

Deception techniques are more sophisticated than traditional security measures and blocking measures, but they support each other. They usually involve honeypots that copy a network or network services and fill them with fake data. Cyber deception is multi-functional. On the one hand, it distracts attackers from your legitimate data. On the other hand, it creates confusion in the minds of opponents, undermining their efforts and slowing down their attacks. The result: the size and complexity of the network increase significantly, forcing attackers to waste resources on useless services or data. By creating false targets or honeypots, as well as luring attackers away from critical data and systems, experts can also control the behavior of attackers. This can help security teams better understand the tactics, methods, and procedures being used against their organization. As a form of threat detection and threat analysis, cyber deception technology is most effective in the way it reveals the psychology of attackers and gathers real-time threat data from adversary activity.

A key advantage of cyber deception technology is that it affects the effectiveness of attackers' actions, making their attacks more resource-intensive. If an attacker spends the time and energy to compromise a decoy server, the defender not only protects valuable assets, but also learns

about the attacker's goals, tools, tactics, and procedures. This is the basic premise of deception tools and technologies. By masking valuable assets in a sea of false attack surfaces, attackers become disoriented and attack the false asset, alerting security teams of their presence in the process. As such, deception tools can be an important defense against Advanced Persistent Threats (APTs).

Deception solutions are designed to trick attackers into thinking they've succeeded and to stealthily lure them into security systems. Deception Distributed Platforms (DDPs) are solutions that create fake systems (often real operating systems but used as victims), decoys (such as fake cookies and browser histories), and honeytokens (fake credentials) on real end-user systems.

The main functions of such systems include:

- Centralized management of real user endpoint decoys and decoy endpoint hosts such as servers and workstation hosts.
- Ability to manage fake services, web applications, and other decoy network integration capabilities.
- Ability to manage endpoint decoys and honeytokens to entice an attacker.
- Ability to administer and distribute deceptive data such as Word documents and tables/records and database files to deception hosts.

Modern DDPs are significantly superior to honeypots, both in terms of functionality and efficiency. Deception platforms include decoys, traps, lures, applications, data, databases, and Active Directory. Modern DDPs can provide extensive capabilities for threat detection, attack analysis, and response automation. Deception is a technique of imitating the IT infrastructure of an enterprise and misleading hackers. As a result, such platforms make it possible to stop attacks before causing significant damage to company assets. Honeypots, of course, do not have such a wide functionality and such a level of automation, so their use requires greater qualifications from employees of information security departments. Thus, different tactics are used: decoys are placed at endpoints to attract the attention of potential attackers. Other decoys are located at the network layer and some work in applications or stored data to target cyber criminals.

4. Discussion

The number of attacks on industrial IoT networks, as well as on elements of critical infrastructure that have IoT devices in their composition, increases by 15-20% every year around the world [23]. Given the constant threat of cyberattacks on elements of Ukraine's critical infrastructure, special attention should be paid to this issue.

The consequences of attacks on such systems can be as follows:

- *Denial of service.* The largest number of attacks carried out lead to denial of service, namely to malfunctions that lead to a partial or complete shutdown of the embedded device.
- *Execution of malicious code.* The consequence of the attack may be the execution of the malicious code entered by the attacker. It also includes various web scripts and SQL injections that can change the behavior of the device.

- *Violation of integrity.* The result of the attacks is a violation of the integrity of some data or the source code of the device's firmware. This includes changing configuration files and settings, as well as applications on the device.
- *Leakage of information.* In some cases, the result of the attack is the unauthorized acquisition of certain information by the attacker.
- *Unauthorized access.* Many attacks result in an attacker gaining unauthorized access to a device. This not only includes cases where an attacker who does not have access to a device can logically gain access to it but also cases where an attacker with access escalates privileges.
- *Decreasing the level of security of the device.* An attacker's actions could cause the device to use weaker algorithms or security policies than those it supports.

The use of specialized deception platforms for IoT is an effective response to these threats. Some deception software allows you to emulate such IT infrastructure objects as databases, workstations, routers, switches, ATMs, servers and SCADA, medical equipment, and IoT. This technology is one of the most effective methods for detecting network threats across all attack surfaces, including hard-to-defend IoT, industrial control systems, point-of-sale terminals, and other devices. Capable of detecting threats that bypass traditional security controls, deception technology is a particularly powerful tool for reducing the amount of time an attacker spends on the network before being detected.

The differences between deception solutions for IoT and deception solutions for general-purpose networks are the use of protocols for communication of Internet of Things devices, including XMPP, COAP, MQTT, HL7, and others. These protocols are used by IoT vendors to support a wide range of applications that enable more consistent machine-to-machine communication and monitoring of critical data and machine health. Accordingly, the IoT deception software is deployed so that it looks like the IoT systems of the enterprise network. Interaction servers and honeypots look like working IoT servers and services, making attackers think they are real. By using honeypots rather than production devices, the attacker reveals, and the platform can quarantine and examine their activities for detailed examination. The analysis engine will analyze the attack methods and the nature of the lateral movement, determine which systems are infected, and provide the signatures necessary to stop the attack. Next, security services can analyze attacks to improve incident response efficiency by automatically or manually blocking and quarantining the attack through integration with third-party prevention systems.

Nowadays, the use of honeypots/honeynet in public networks is a rather controversial and often ineffective practice due to the high probability of their detection. However, the use of specialized honeypots/honeynets in IoT networks is still quite effective [24]. Due to the peculiarities of the architecture and communication protocols with IoT devices, the use of even low-intensity honeypots is an effective marker of the beginning of an attack. Especially if you place such honeypots/honeynets systematically and monitor current security threats of IoT devices for modifying baits. To increase data collection and gain a better understanding of threats, honeypots used different levels of interaction. In addition, their IP addresses must be cycled so that the honeypots are not flagged as honeypots, reducing the number of attacks and the amount of useful information that could be gathered.

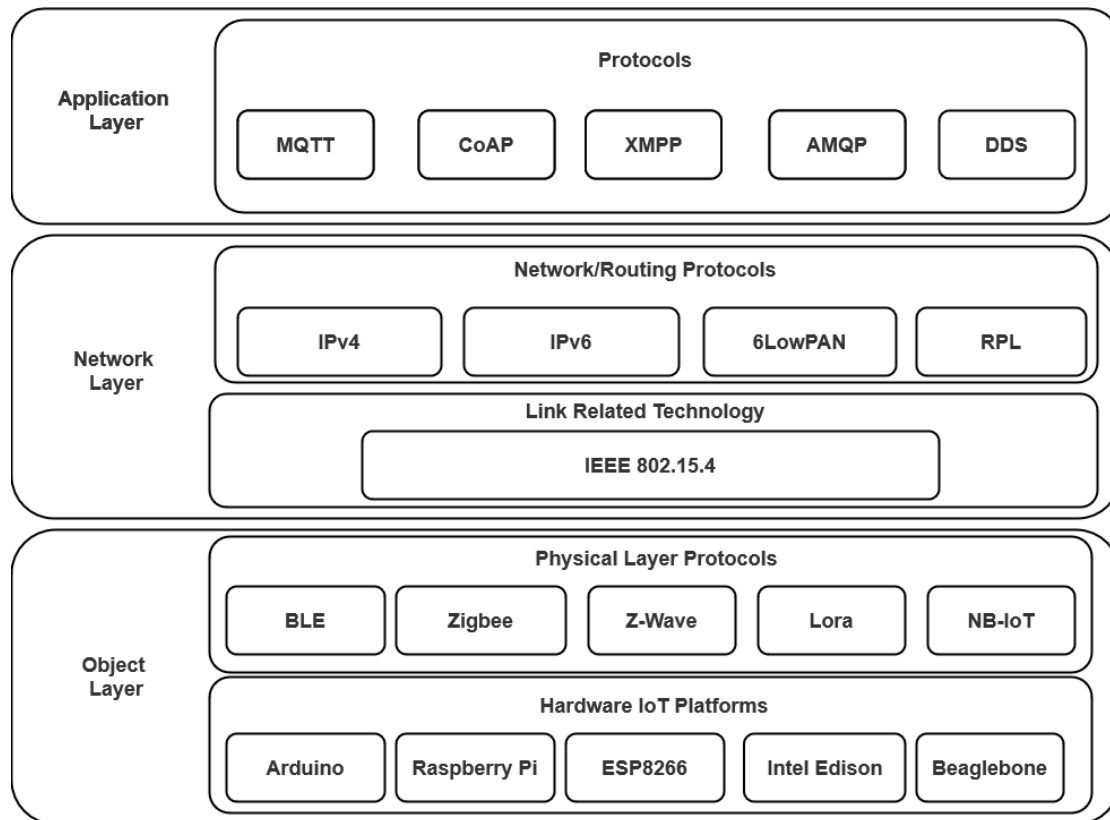


Figure 2: IoT protocols per Operating Layers Table.

Without a doubt, SSH, Telnet, and web servers are some of the most commonly used and accessible services in the Internet of Things, making them an attractive target for attackers. In addition, IoT devices typically use a variety of computing architectures that differ significantly from those used by traditional computer networks. This is why attackers are more likely to launch their software when they gain access to a honeypot without checking what architecture they are using. This allows researchers to trace the sources of the attack tools used by attackers, allowing them to study them much more efficiently later.

IoT devices have certain features that need to be taken into account when creating a honeypot/honeynet. To maximize a hacker's chances of finding and exploiting vulnerabilities, the honeypot must remain anonymous, mimicking a real system to prevent it from being easily identified by attackers. Due to the nature of IoT devices and the inability to fully understand the nature and activities of an attacker, an effective honeypot will require a different approach.

IoT honeypots inherit some characteristics from general-purpose honeypots, including the ability to respond to events as they occur. Although these honeypots are not designed specifically for IoT, they are currently sometimes used for IoT honeypot research. An example is Honeyd, which allows you not only to create virtual media but also to integrate machines. There are several protocols supported by this honeypot, including UDP, TCP, FTP, SMTP, Telnet, IIS, POP,

Table 1

List of IoT honeypots that focus on specific attacks.

Honeypot	Interaction Level	Target attack
U-Pot	Medium	UPnP
HoneyIoT	Low	Reconnaissance
HioTPot	Medium	Attacks on authentication
IoT POT	Hybrid	Telnet
MTPot	Low	Telnet
Phype	Medium	Telnet
Shrivastava	Medium	SSH and Telnet
IRASSH-T	Medium	SSH and Telnet
Honeycloud	High	Fileless attacks
Dowling	Medium	SSH over Zigbee
Pot2DPI	Medium	Attacks on home networks
Siphon	High	Attacks on device characteristics
Metongnon	Low	Attacks on device characteristics
Zhang	Hybrid	Attacks on device characteristics

and telnet. Various studies have investigated whether HoneyD can be used to create effective honeypots that attract attackers. Dionaea [25] is open-source software that allows users to create middleware honeypots that simulate various services (e.g., FTP, HTTP, MQTT, etc.). This program targets attackers who attack hosts on the Internet using vulnerable services. With Cowrie, it is possible to create scalable honeypots of medium and high levels of interaction that can monitor and control various behaviors. As an intermediate interaction honeypot, it records the interaction of an attacker's shell on a simulated UNIX system by emulating multiple commands. As a high-interaction honeypot, it is a proxy for SSH and Telnet to observe the interaction of an attacker on another system. Essentially, it acts as a proxy between the attacker and a group of virtual machines that are configured on the host server, allowing for flexible configuration.

The most versatile IoT honeypots are capable of emulating any device connected to the Internet. With full device emulation, it is harder for attackers to detect the honeypot, which adds more realism to the honeypot. With the ThingPot platform, a complete IoT platform can be emulated and supported at the application level, ensuring that your IoT system is scalable, virtual, open, and scalable. Also worth mentioning is IoT CandyJar [26], which can reproduce the behavior of IoT devices without the risk of being compromised because they are smart and mimic the behavior of authentic IoT devices. They are called lures of intellectual interaction. Conpot [27] is one of the most popular ICS honeypots and has been used by researchers for many years. Conpot supports many industrial protocols, including Building Automation and Control Network, Guardian AST, Kamstrup, Modbus, S7comm, and many others, such as HTTP, FTP, SNMP, Intelligent Platform Management Interface, and TFTP. The kit includes templates for Siemens S7 class PLCs, Guardian AST tank monitoring systems, and Kamstrup smart meters.

5. Conclusions

The threat of cyberattacks on critical infrastructure, as well as the growing importance of IoT systems, requires the search for effective mechanisms for detecting and preventing such attacks. This is a worldwide trend and a solution to this problem must be found now. One of the most promising approaches to detecting attacks on both critical infrastructure objects and industrial CFS and IIoT networks is the use of cyber deception systems and complex honeypot solutions. These systems can be used both to prevent attacks and to obtain complete and up-to-date information about who the attackers are, what tools they have, and how they gain access to these devices. And this, in turn, will make it possible to change security measures more quickly and effectively and prevent further attacks. However, for the effective use of such systems, it is necessary to have a good understanding of their capabilities.

We plan to focus our further research on the deployment of a plausible IoT honeynet network, which will contain typical configurations and settings for IoT networks of Ukraine to collect static information on the vectors and techniques of attackers' attacks. Increasing and improving the functionality of this network in combination with the use of machine learning technologies to generate plausible intra-network traffic will allow to explore the toolkit of attackers for detecting honeypots and honeynets in IoT networks.

References

- [1] C. Talos, Talos Year in Review 2022, 2022. URL: <https://blog.talosintelligence.com/talos-year-in-review-2022>.
- [2] N. M. Lobanchykova, I. A. Pilkevych, O. Korchenko, Analysis and protection of IoT systems: Edge computing and decentralized decision-making, *Journal of Edge Computing* 1 (2022) 55–67. doi:10.55056/jec.573.
- [3] D. Fraunholz, M. Zimmermann, H. D. Schotten, Towards Deployment Strategies for Deception Systems Unsupervised Machine Learning, *Advances in Science, Technology and Engineering Systems Journal* 2 (2017) 1272–1279. doi:10.25046/aj0203161.
- [4] D. Fraunholz, H. D. Schotten, Defending Web Servers with Feints, Distraction and Obfuscation, in: 2018 International Conference on Computing, Networking and Communications (ICNC), 2018, pp. 21–25. doi:10.1109/ICCNC.2018.8390365.
- [5] D. Reti, D. Fraunholz, J. Zemitis, D. Schneider, H. D. Schotten, Deep Down the Rabbit Hole: On References in Networks of Decoy Elements, in: 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 2020, pp. 1–11. doi:10.1109/CyberSecurity49315.2020.9138850.
- [6] D. Fraunholz, D. Krohmer, H. D. Schotten, C. Nogueira, Introducing Falcom: A Multifunctional High-Interaction Honeypot Framework for Industrial and Embedded Applications, in: 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 2018, pp. 1–8. doi:10.1109/CyberSecPODS.2018.8560675.
- [7] D. Fraunholz, D. Krohmer, S. D. Anton, H. Dieter Schotten, 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), 2017, pp. 1–7. doi:10.1109/CyberSecPODS.2017.8074855.
- [8] A. Acien, A. Nieto, G. Fernandez, J. Lopez, A Comprehensive Methodology for Deploying IoT Honeypots, in: S. Furnell, H. Mouratidis, G. Pernul (Eds.), Trust, Privacy and Security in Digital Business, volume 11033 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2018, pp. 229–243. doi:10.1007/978-3-319-98385-1_16.
 - [9] H. Šemić, S. Mrdovic, IoT honeypot: A multi-component solution for handling manual and Mirai-based attacks, in: 2017 25th Telecommunication Forum (TELFOR), 2017, pp. 1–4. doi:10.1109/TELFOR.2017.8249458.
 - [10] M. A. Hakim, H. Aksu, A. S. Uluagac, K. Akkaya, U-PoT: A Honeypot Framework for UPnP-Based IoT Devices, in: 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC), 2018, pp. 1–8. doi:10.1109/IPCCC.2018.8711321.
 - [11] C. Brooks, Cybersecurity in 2022 – A Fresh Look at Some Very Alarming Stats, 2022. URL: <https://www.forbes.com/sites/chuckbrooks/2022/01/21/cybersecurity-in-2022--a-fresh-look-at-some-very-alarming-stats>.
 - [12] Microsoft Digital Defense Report 2022, 2022. URL: <https://www.microsoft.com/en-us/security/business/microsoft-digital-defense-report-2022>.
 - [13] S. Ravji, M. Ali, Integrated Intrusion Detection and Prevention System with Honeypot in Cloud Computing, in: 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), 2018, pp. 95–100. doi:10.1109/iCCECOME.2018.8658593.
 - [14] A. Pashaei, M. E. Akbari, M. Zolfy Lighvan, A. Charmin, Early Intrusion Detection System using honeypot for industrial control networks, Results in Engineering 16 (2022) 100576. doi:10.1016/j.rineng.2022.100576.
 - [15] D. Zielinski, H. A. Kholidy, An Analysis of Honeypots and their Impact as a Cyber Deception Tactic, 2022. arXiv:2301.00045.
 - [16] M. Tsikerdekis, S. Zeadally, A. Schlesener, N. Sklavos, Approaches for Preventing Honeypot Detection and Compromise, in: 2018 Global Information Infrastructure and Networking Symposium (GIIS), 2018, pp. 1–6. doi:10.1109/GIIS.2018.8635603.
 - [17] W. Fan, Z. Du, D. Fernández, Taxonomy of honeynet solutions, in: 2015 SAI Intelligent Systems Conference (IntelliSys), 2015, pp. 1002–1009. doi:10.1109/IntelliSys.2015.7361266.
 - [18] J. Matherly, Honeypot or Not?, 2022. URL: <https://honeyscore.shodan.io/>.
 - [19] T-Pot - The All In One Multi Honeypot Platform, 2023. URL: <https://github.com/telekom-security/tpotce>.
 - [20] J. Franco, A. Aris, B. Canberk, A. S. Uluagac, A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems, IEEE Communications Surveys & Tutorials 23 (2021) 2351–2383. doi:10.1109/COMST.2021.3106669.
 - [21] D. Fraunholz, S. D. Anton, C. Lipps, D. Reti, D. Krohmer, F. Pohl, M. Tammen, H. D. Schotten, Demystifying deception technology: a survey, 2018. arXiv:1804.06196.
 - [22] M. Soria-Machado, D. Abolins, C. Boldea, K. Socha, Detecting Lateral Movements in Windows Infrastructure, CERT-EU Security Whitepaper 17-002, 2017. URL: https://cert.europa.eu/static/WhitePapers/CERT-EU_SWP_17-002_Lateral_Movements.pdf.

- [23] S. Bennett, IoT Security Statistics 2023, 2023. URL: <https://webinarcare.com/best-iot-security-software/iot-security-statistics/>.
- [24] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, C. Rossow, IoT-POT: A Novel Honeypot for Revealing Current IoT Threats, *Journal of Information Processing* 24 (2016) 522–533. doi:10.2197/ipsj.jip.24.522.
- [25] Dionea honeypot, 2021. URL: <https://github.com/DinoTools/dionaea>.
- [26] Intelligent-IoT-Honeypot, 2019. URL: <https://github.com/as2d3/Intelligent-IoT-Honeypot>.
- [27] Conpot, 2022. URL: <https://github.com/mushorg/conpot>.