Computers & Security 114 (2022) 102598

Contents lists available at ScienceDirect

Computers & Security

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

TC 11 Briefing Papers

Don't get stung, cover your ICS in honey: How do honeypots fit within industrial control system security

Sam Maesschalck*, Vasileios Giotsas, Benjamin Green, Nicholas Race

Security Lancaster, Lancaster University, United Kingdom

ARTICLE INFO

Article history:
Received 5 October 2021
Revised 17 December 2021
Accepted 27 December 2021
Available online 30 December 2021

Keywords:
Honeypots
Industrial control systems
ICS
Malware
Security
Critical infrastructure

ABSTRACT

The advent of Industry 4.0 and smart manufacturing has led to an increased convergence of traditional manufacturing and production technologies with IP communications. Legacy Industrial Control System (ICS) devices, now interconnected via public networks, are exposed to a wide range of previously unconsidered threats, which must be considered to ensure the continued safe operation of industrial processes. This paper surveys the ICS honeypot deployments in the literature to date, provides an overview of ICS focused threat vectors, and studies how honeypots can be integrated within an organisations defensive strategy. We discuss relevant legislation, such as the UK Cyber Assessment Framework, the US NIST Framework for Improving Critical Infrastructure Cybersecurity, and associated industry-based standards and guidelines supporting operator compliance. This is used to frame a discussion on our survey of existing ICS honeypot implementations, and the role of honeypots in supporting regulatory objectives. We observe that many low-interaction honeypots are limited in their use. This is largely due to the increased knowledge attackers have on how real-world ICS devices are configured and operate vs the configurability of simulated honeypot systems. Furthermore, we find that environments with increased interaction provide more extensive capabilities and value, due to their inherent obfuscation delivered through the use of real-world systems. Based on these insights, we propose a novel framework towards the classification and implementation of ICS honeypots.

1. Introduction

The increasing adoption of the Internet of Everything (IoE) sees both a shift within industry and consumers, as more and more devices are becoming connected to the Internet. This trend also encapsulates the industrial technologies categorised under the umbrella term of Industrial Control Systems (ICSs) (Bodenheim, 2014), which are used within critical infrastructure and are designed for high reliability. The convergence of traditional ICSs with machine-to-machine (M2M) and IP communications has been characterised as the fourth industrial revolution, or Industry 4.0 (Lasi et al., 2014), which promises to improve operational functionality, manageability, and ease of access. However, ICSs were not designed with Internet connectivity in mind (Ahmed et al., 2017), and often lack basic security features (Mirian et al., 2016), making them vulnerable to cyber-attacks. Moreover, when security features are available for ICS, they are usually 'bolt-on'. These can include vulnerabilities of their own, such as Secure Authentication version 5 (SAv5) for DNP3 (Distributed Network Protocol 3) (Crain and Bratus, 2015), which is vulnerable to single-frame attacks, does not encrypt data between server and outstation (Cremers et al., 2019), and allows for the use of SHA-1 (Rosborough et al., 2019) amongst others. Generally, bolt-on security is considered a weaker option to secure a system (Shiva et al., 2010) and harms both usability and security (Yee, 2004). The risk of ICS attacks is amplified by the fact that these systems are often implemented as part of the critical infrastructure within a country, including water and electricity distribution (Green et al., 2017). Therefore, ICS security is paramount to the safety and economic prosperity of a nation and because these systems present an attractive target for cyber-warfare operations. This can also be seen in the shift of cyberattacks targeting critical infrastructure from initially internal personnel to nation-states in the present time (Miller et al., 2021). This also needs to be taken into account within risk assessment, such as described in the ad-

* Corresponding author.
E-mail addresses: s.maesschalck@lancaster.ac.uk (S. Maesschalck), v.giotsas@lancaster.ac.uk (V. Giotsas), b.green2@lancaster.ac.uk (B. Green), n.race@lancaster.ac.uk (N. Race).

https://doi.org/10.1016/j.cose.2021.102598

versary cost framework (Derbyshire et al., 2021), as a nation-state adversary generally has more resources to its disposal.

Even when manufacturers provide patches for known vulnerabilities, patch deployment times can be significantly higher compared to traditional IT systems, leading to more prolonged exposure times (Dey et al., 2015; Marnerides et al., 2019). Delayed patching can be explained through the requirement for continued operation and minimal downtime. Within an ICS environment system, reliability takes precedence over security (Maglaras et al., 2018), meaning ICS operators may prefer to leave systems unpatched. However, where vulnerabilities exist, their exploitation has the potential to harm operational productivity, reliability, and even human life.

To effectively protect ICSs, it is necessary to develop new methods for attack detection and mitigation. It is no longer sufficient to rely on traditional firewalls and anti-virus solutions, as they are reactive and require updates in order to detect/prevent new forms of malicious traffic (Bilge and Dumitras, 2012). Consequently, zero-day exploits, namely exploits which are not yet publicly disclosed, can potentially penetrate networks and infect systems while remaining undetected. The introduction of "bring your own device" within organisations, and the prevalence of social-engineering (Derbyshire et al., 2018), has rendered conventional perimeter defences inadequate (Wang et al., 2014). Due to the merging of OT and IT, these threats now also apply to ICSs.

One of the ways in which we can aim to mitigate attacks on the network and discover novel attacks is through honeypots. Which are systems with no inherent purpose other than capturing attacks, either on the Internet or within a network, and generally do not receive any legitimate traffic. Both academia and industry have been using and researching honeypots in a range of different use cases. These different use cases tend to have many different setups; whereas academia can deploy many different honeypots, industry tends to focus on honeypots that more closely align with their operations. These differences in purpose and setup are encompassed in the distinction between research and production honeypots. There are many different types of honeypots, ranging from emulating specific services such as SSH to a fully fledged system running several services at the same time. It would be a red flag for attackers seeing an ICS honeypot deployed by an organisation that generally does not use these systems. Contrary to traditional security systems that are often reactive, honeypots enable a more proactive approach to security. Adversaries are encouraged to attack these systems to reveal valuable threat intelligence. Capturing attacks performed by real-world adversaries can be used to discover new vulnerabilities and associated exploits, alongside a broader view of offensive tactics and techniques. The level of encouragement differs depending on the purpose and environment in which the honeypots are deployed, as honeypots could fall into the legal aspect of entrapment. Generally, in a real-world environment you would want an adversary that has entered the organisational network to be more likely to investigate a honeypot than an operational system. In 2020, four zero-day exploits were discovered by ICS honeypots set up for research purposes (Ranger, 2020) proving the viability of these systems in detecting novel attacks.

Research into honeypots within ICS environments has already been done, but this has been fractured. With this , we provide a survey of existing honeypot deployments within the literature to date and provide further background into ICS honeypots. To achieve this, we cover the general aspects of ICSs and honeypots. One important aspect within ICS is the legislative part, as critical infrastructure tends to be heavily regulated; we tackle this by covering both country-specific and international standards and guidelines. Afterwards, we map these onto honeypots and their capabilities to support these guidelines. This provides a strong background to investigate honeypot deployments within the academic space, and combined with the other aspects we aim to show their benefits within industry.

The core contributions of this paper are as follows:

- An survey of existing ICS-focused honeypot implementations.
- Review of ICS standards and guidance, and how honeypots fit within these.
- The introduction of a novel classification scheme for honeypot implementations.
- The introduction of a novel framework supporting honeypot deployments.

Section 2 provides an overview of ICSs, honeypots, and threat vectors. Section 3 discusses how a selection of historic ICS attacks were executed, and their resulting impact. Section 4 explores honeypots within the context of standards and guidance for critical national infrastructure operators, which covers both governmental regulations and guidance published by non-governmental organisations. The section further details international standards and guidelines referenced by the UKs National Cyber Security Centre (NCSC), as appropriate resources to support NIS compliance, and provides an introduction into honeypots and the potential benefits derived through their use in ICSs. This provides a foundation towards a more in-depth exploration of existing honeypot implementation in Section 5, and our novel framework supporting ICS honeypot deployments in Section 6. Finally, we conclude the paper in Section 7, and discuss areas of future work in Section 8.

2. Background

2.1. Industrial control systems

Industrial control systems underpin critical parts of national infrastructure. They control and automate industrial process operations within a variety of industries, including nuclear, water, oil and gas, and electricity (Green et al., 2017; McLaughlin et al., 2016). Due to the organisations that deploy ICSs, it is clear that the impact of an attack on these systems can be considerable. Therefore, appropriate defence mechanisms should be in place to prevent potential damage. The current trend of Internet-connected ICSs opens these systems up for a variety of threats. ICSs were not originally designed to communicate over the Internet (Ahmed et al., 2017). The operating systems (OS) and other software used within these systems can have vulnerabilities which are not regularly patched, and specific protocols used present many difficulties due to their design which adversaries can potentially exploit. The vital function an ICS has within critical infrastructure, combined with the insecure design of ICS protocols, can lead to potentially catastrophic events.

Because of this, novel defence approaches are needed to mitigate emerging threats. ICS devices are built using commercial OS that are highly specialised, and therefore ICS security differs considerably from standard approaches to security (Knapp and Langill, 2014). Existing ICS security solutions aim to minimise disruptions in ICS availability by focusing on protecting the IT infrastructure around the ICS devices (Larkin et al., 2012). Due to the importance of these devices, any interference or additional latency can have significant effects (Jie and Li, 2011). ICSs can be operational without interruptions for up to two decades , unlike IT systems which are regularly updated (Hunter, 2006) or replaced (Frye, 2013). Such a gap between the discovery of a vulnerability and the implementation of the patch allows attackers time to discover and exploit them for years after those vulnerabilities have been published (Marnerides et al., 2019).

Typically ICSs are deployed within a complex environment which consists of several layers of logically-related operational abstractions. One of the most popular representations of these lay-

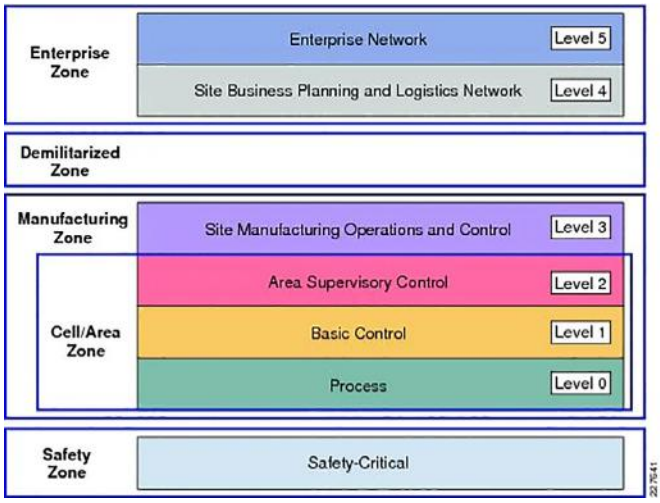

Fig. 1. The Extended Purdue Model which describes the layered architecture of ICS (Didier et al., 2011).

ers is the Purdue model (Fig. 1) which consists of an enterprise zone, demilitarised zone (DMZ), manufacturing zone and safety zone (Didier et al., 2011; Green et al., 2016). The function of the *safety zone* is to house systems that provide predictable fail-safe shutdowns to protect processes, personnel, and the environment. They also monitor the processes running for any anomalies (Obregon, 2015). More recently, CIP Safety (ODVA) allows devices such as safety sensors to operate alongside level 0 systems and safety controllers with controllers on level 1. Within the *manufacturing zone*, there are four levels which contain a set of devices including PLCs (Programmable Logic Controllers), HMIs (Human Machine Interfaces), and RTUs (Remote Transmission Units) which are used to monitor, control and automate processes. Devices within this zone include sensors, human-machine interfaces, remote terminal units and control servers. The *DMZ* is implemented as a boundary between the manufacturing and enterprise zone, and it generally contains IT infrastructure that has the capability to communicate with the OT devices. It presents an interface for further processing on data and facilitates services like remote desktop and remote alarm management. Within the *enterprise zone*, conventional IT devices such as clients and servers are deployed and use data collected via the DMZ to supervise and dictate future strategic planning for the entire infrastructure.

2.2. Honeypots

Honeypots can come in different variants; they can be either virtual or physical and are designed to be exploitable (The Honeynet Project, 2001). The goal of a honeypot is to lure attackers into targeting them. For research honeypots, that are Internet-facing, and are deployed with the main goal of gathering information for research purposes. This is different than for production honeypots which are usually not directly accessible and are deployed inside an organisational network to improve their security. To refrain from entering the legal area of entrapment, honeypots need to be deployed and configured with care. Within an organisation we would expect an aim to direct attackers to a honeypot once they are inside of the network. For research honeypots it is sufficient to make the honeypot accessible from the Internet to capture interest. When honeypots are compromised, they can be used to generate alerts or to deceive the attacker by diverting exploitation efforts away from the systems that need to be protected.

Therefore, the value of the honeypot is determined by the number of attacks it receives (Zhang et al., 2003). Honeypots that are actively attacked provide the most valuable information, but even when they are not exploited honeypots can indicate if a network is being actively targeted. To achieve valuable activity, it is essential to both lure attackers to the honeypots by introducing vulnerabilities whilst also maintaining a reasonable level of security to resemble an operational system (Rowe, 2006). When a system is significantly less secure than others within the same network, it can be seen as an indication of a potential honeypot.

Data gathered through honeypots can be used in many ways. For example, they can provide useful data which can be used to create a timeline of an attack. This is important for accurate threat intelligence, and generally hard to construct (Caravelli, 2019). By implementing honeypots and luring attackers away from real infrastructure, an organisation can both improve its security through the data collected (Caravelli, 2019), and reduce the usage of resources on business systems. Therefore, monitoring of traffic to and from honeypots, and the attackers' actions within them, are crucial aspects of honeypot operations. Due to the nature of a honeypot, by not performing business operations, all traffic to them can be considered malicious. However, this lack of real operational purpose within the network makes it harder to deceive attackers, as there is no actual active traffic between them (Rowe, 2006). Additionally, the level of interaction that an intruder is permitted to have with the honeypot can affect the behaviour of the intruder and therefore, the volume of collected attack data. Generally, the goal of honeypots and their detection capabilities is to gather data to feed into the protection of the network or systems.

In general, honeypots are categorised based on their level of permitted interaction to high-interaction and low-interaction honeypots. A third category, medium-interaction honeypots (Mokube and Adams, 2007), does exist but its characteristics lay close to a low-interaction variant. Low-interaction honeypots pretend to be a specific device such as a Programmable Logic Controller (PLC) and mimic its functions, they run on a standard operating system (e.g. Ubuntu) and provide limited interaction for attackers (Chamotra et al., 2011). For instance, the *Honeyd* honeypot (Provos, 2003) provides a TCP/IP stack emulator which allows an attacker to send network requests which it responds to. However, the attacker cannot have further interaction with the other parts of the system, such as the operating system. This might result in increased identification of the system as a honeypot, and shorter interactions with the system. Using an approach like PCaaD (Green et al., 2021), would also not be possible on a low-interaction honeypot. In contrast, high-interaction honeypots are, in essence, the same device as would be in the operational network, allowing attackers to interact with every aspect of the machine (Spitzner, 2002). These honeypots are generally less easy to identify and allow attackers to perform more actions and increase their time and interaction on the system.

The benefit of high-interaction honeypots is that attackers are less likely to identify them as honeypots, and they can provide considerably more data from the attack (Chamotra et al., 2011). On the other hand, high-interaction honeypots demand significantly more resources, and they entail the risk of the attacker taking over the system due to the high level of interaction they permit (Vetterl, Clayton, 2018). Within an ICS environment, deploying a high-interaction honeypot entails the usage of a real PLC or other ICS device. These devices are expensive, and a single device does not accurately represent a real ICS deployment. Therefore, to achieve full high-interaction multiple devices have to be deployed to transfer data between them.

Improving upon some low-interaction honeypots by integrating additional characteristics of a real system and creating a so-called medium-interaction honeypot, can provide more data whilst still entailing a lower level of risk. However, all features on them are simulated as well. Therefore, we consider such medium-interaction

Table 1
Overview of Honeypot Characteristics.

Level of Interaction	Low	Medium	High
Risk	Low	Low	High
Data Capturing	Basic	Intermediate	Comprehensive
Resource Usage	Low	Low	High
Simulation	Basic	Improved	N/A
Required Knowledge	Low	Advanced	High
Detection	Easy	Easy - Normal	Normal - Hard
Cost	Low	Low	High

honeypots as a sub-category of low-interaction honeypots. For example, they can simulate responses of a specific service, such as a web server. They can lure attackers that scan for particular vulnerabilities or exposed services, without possessing the risk of being exploited. However, medium-interaction honeypots do not run a full operating system (unlike those that are high-interaction) and therefore the data available about potential attacks is more limited (Spitzner, 2002). Looking specifically at an ICS environment, it can be challenging to have a simulated system perform close to a real system. Particularly when taking into account an device, like a PLC, does require input from another system to display data. This could potentially be circumvented by providing the system with static data, which is easy to identify by an attacker. More dynamic data is possible, but requires more effort as this data has to be somewhat realistic, e.g. a water tank cannot go from 100% to 0% in one second, and if there would be an identifiable pattern it could also be detected. Therefore, due to the nature of ICS and if the goal is to capture useful data from knowledgeable attackers, the deployment of a honeypot has to be extremely believable. Overall, medium-interaction honeypot implementations tend to be less frequent than high- and low-interaction variants (Paralax, 2019). Nonetheless, it is essential to note that the distinction between low- and medium-interaction is not always clear and largely depends on the context of the simulated environment.

An overview of general honeypot characteristics can be found in Table 1. These characteristics range from resource usage, and risk involved in deploying them to the knowledge required to set-up and operate the honeypot, and ease of detection.

When talking about risk, we talk about the attacker being able to leverage the honeypot and use it against us. A low level of risk implies that the attacker has limited ways to leverage the honeypot and when correctly deployed, should not be able to use it to pivot into the network. A high level of risk means there is a possibility of the attacker taking control of the honeypot and using it as a way into the network.

Data capturing scales from basic to intermediate and comprehensive. Basic refers to the limited amount of information captured by the honeypot; this tends to be limited to the information included in the IP packet. Intermediate improves upon basic by having the capabilities to capture more IP packets, as the attacker can perform more interactions. Building further upon this is comprehensive, where the attacker has the opportunity to interact with the whole system and can perform any actions that are generally performed in an attack including uploading of data and interacting with other systems on the device.

Resource usage refers to the resources needed to operate a honeypot. Low resource usage means the honeypot can be deployed in a virtual environment; this allows for multiple separated honeypots on the same physical system. High resource usage refers to the need of a physical device set up for one specific honeypot, and typically there is also a need for more software on the system (e.g. keyloggers) and other monitoring systems on the network (e.g. IDS).

Simulation relates to the level of simulation, which can range from limited simulation of services (basic) to a more comprehensive level of simulation, which allows for more interaction (improved). To deploy honeypots, there is a knowledge aspect involved. For a low-interaction honeypot, this is relatively low, as these can generally be installed as an easy to deploy package. An advanced level of knowledge refers to the possible need to adapt and improve the simulation of the honeypot. For a high-interaction honeypot, we would advise a high level of knowledge due to the risks involved, and the more do-it-yourself actions involved in creating a honeypot from a real device.

Detection of the honeypots concerns the ease of detection, which can range from easy to normal for simulated services (depending on the comprehensiveness of the simulation), and normal to hard for high-interaction honeypots (depending on the deployment). To further clarify, for a high interaction honeypot, it is important to strike a balance between security and included vulnerabilities to lower the ease of detection.

Finally, the cost of the honeypot is a combination of the resources required to operate the honeypot and the resources necessary to deploy the honeypot. For low- and medium- interaction this is low as they only require a virtual environment and the honeypot software (which tends to be open-source). A high-interaction honeypot incurs more costs to purchase the device, required software to monitor the device and other systems, such as an IDS.

2.3. ICS Security

We have already mentioned that existing ICS security techniques mainly focus on availability and the IT environment around the infrastructure. However, there have been calls to move the focus from just availability to also focus on the security of the systems against malicious cyberattacks (Cárdenas et al., 2008). Previous studies have found that honeypots can be used to improve the security of SCADA systems (Disso et al., 2013), which are a subset of ICSs. But generally, honeypots are not often considered by ICS security researchers. One of the main drivers for the security of ICSs and critical infrastructure is standards and guidelines in conjunction with regulation. Standards like ISO 27019, IEC/ISA 62,443 and NIST SP 800-82 are generally used within an ICS environment. Therefore, these documents are used by industry when deploying and securing their ICS infrastructure.

It is clear from looking at the ICS security space that there has been a lack of built-in security. This is understandable when looking at it from the perspective that these systems were not designed to be accessible aside from engineers working on them, but as stated before, this has changed in recent times. Bhamare et al. (2020) in their 2020 survey have mainly focused on machine learning (ML) to improve upon ICS security. These approached range from risk assessment based on ML and ML to detect malicious communications within the SCADA environment to Cloud-based computing for attack mitigation. However, to achieve potent ML-based approached data is needed to train the system or feed into the system. We feel that honeypots can greatly improve this as, if deployed within the organisation, they can provide a lot of useful real-time data of threats inside the network. When deployed outside the organisation or deployed with a research focus within the organisation they can provide useful general data.

Generally techniques applied within the ICS space can be categorised as focusing on the architecture, strategy, attack modelling, attack detection and attack categorisation (Ani et al., 2018). Especially with the trend to expose these networks and systems to the Internet, thinking about these six categories becomes even more important. All these areas have to work together and feed into each other. Like within traditional IT environments, security should be encompassed throughout the design. Relying on bolt-on security, like SAv5 for DNP3, should only be viewed as temporary fixes and not be relied on for an extensive period. The inherent vulnerabil-

ities within ICS protocols such as DNP3, Modbus and IEC 61,850 should be tackled with security in mind from the design phase onward.

There is an immediate need for research and development within the ICS security space. Research done with ICS testbeds such as described by Green et al. (2020) will be important for the future of ICS security and the security of our society. ICS security risks come from a socio-technical angle and requires an understanding of how stakeholder decisions from all levels impact the security of these devices. There is a real complexity that stems from the combination of devices and systems within critical infrastructure environments (Rashid et al., 2019). The introduction of honeypots would enable further evaluation of threats and the adversaries behind them.

3. ICS Attacks in practice

Over recent years, several high impact attacks on ICSs have been carried out. Some examples of ICS attacks that happened over the past 30 years are Salt River Project in 1994, Gazprom in 1999, Daimler Chrysler in 2005, Night Dragon in 2009, Rye Brook Dam in 2015 and Triton in 2017. Within this section, we present three important case studies of such attacks to highlight the potential impact that an attack on an ICS device can have in the operation of industrial systems. We have selected these three attacks for their distinctiveness, and the level of coverage Stuxnet and the Ukrainian attack have received. Stuxnet being an attack that originated from within the organisation, the BlackEnergy on the Ukrainian energy systems which exploited MS Word vulnerabilities and used a known piece of malware and Wolf Creek which is a great example of how appropriate security measures are important to mitigate the possible effects of an attack.

3.1. Stuxnet

Stuxnet is widely recognised within the cybersecurity community and viewed as one of the most well-known ICS focused cyber attacks. After the discovery of Stuxnet, it has been stated the world has entered in a new area of warfare and a pivotal moment in cyber security (Langner, 2011). Stuxnet has been used as an argument to improve cybersecurity, to question the current international laws regulating this space (Richmond, 2011), and to explore the future of warfare (Farwell and Rohozinski, 2011).

In 2010 Sergey Ulasen (Kaspersky, 2011) discovered malware that targeted Iranian nuclear facilities, which is widely suspected to be carried out as a joint military attack by the United States and Israeltha (Nakashima and Warrick, 2012). Nevertheless, like most cyberattacks, attributing it to a party is difficult (Farwell and Rohozinski, 2011). Unlike other pieces of malware seen before, Stuxnet was much more complicated and did not have any intention to steal data but instead had the objective to destroy a physical target (centrifuges) and delay the Iranian nuclear program (Collins and McCombie, 2012; Langner, 2011). This made the security world aware that cyberattacks can impact the physical and virtual worlds alike. Further, according to Farwell et al. (Farwell and Rohozinski, 2011), Stuxnet has been the first malware of the 'fire and forget' generation, as it designed to work in a quasi-autonomous manner. This increased the spread rate also lowered the control the adversary has over it. The initial infection happened via a USB drive that was plugged in into a machine in the facility. Then the worm spread automatically over the local network or USB drives connected to the systems with the ambition to further infect Windows computers on the network. Stuxnet exploited four zero-day vulnerabilities (Kushner, 2013). Worldwide, it is believed there were around 100,000 systems infected by the dropper (Langner, 2011).

Table 2
Key improvements between BlackEnergy versions (Khan et al., 2016; Miller et al., 2021).

Feature	v1	v2	Lite	v3
Plugins		X	X	X
Denial of Service	X	X	X	X
C2C Controller	X	X	X	X
Anti Virus Obfuscation	X	X	X	X
Kernel Rootkit			X	X
Bypass Driver Signing				X
Reside in Memory				X
Detect Virtual Environment				X
Detect Countermeasures				X

3.2. BlackEnergy

In December 2015 it was discovered that the BlackEnergy malware was used to attack electricity distribution companies across Ukraine, which resulted in power outages that left more than 225,000 people without electricity (Department of Homeland Security, 2016; Khan et al., 2016). However, this was not the first or only time BlackEnergy was used in an attack. Within the United States, an attack on critical infrastructure using BlackEnergy could have had disastrous effects on the country, if gone undiscovered (ThreatStop, 2016).

In total there are four known versions of BlackEnergy. BlackEnergy version 3 (BE3), which was used in the Ukrainian ICS attacks, exploited vulnerabilities in Microsoft Office and propagated through Microsoft Word documents via spear phishing (US-CERT), and eventually managed to target the breakers of seven substations (Khan et al., 2016). Due to the ongoing political dispute between Ukraine and Russia, it is suspected that the Russian state-sponsored the attack, although such involvement has not been proved (Cherepanov and Lipovsky, 2016).

BlackEnergy is a notable example of how malware evolves over time, rendering traditional defences inefficient as the malware evolves to evade new security measures. Since the first version, it has evolved into a complex multi-purpose piece of malware. Version 2 expanded the espionage, spam and fraud capabilities significantly, and used a modular design which allowed adversaries to use plugins to customise the attack to specific targets (Khan et al., 2016). The latest version (BE3) simplified the method is used to deliver the malware payload (ThreatStop, 2016). Further, it expanded the functionalities it had to evade detection and used different communication protocols. An overview of the BlackEnergy versions and their evolving capabilities can be found in Table 2.

3.3. Wolf Creek

Unlike the Stuxnet and BlackEnergy attacks, the attack on the Wolf Creek Nuclear Operating Corporation (2017) caused no disruptions to the facility itself. As with other nuclear power plants, the operational systems are not part of the business network, and the ICSs are not connected to the Internet (Caravelli, 2019). This shows that an ICS environment that is separated from the IT network is better protected, however, with the current trend this separation is seen less and less. To gain a foodhold in the network emails containing malicious documents sent to senior industrial control engineers, through which the adversaries, supposedly, wanted to map the network for further attacks on the facility (Perlroth, 2017).

Despite the increasing awareness from governments and international agencies, the attack against the Wolf Creek plant highlights the challenge of tackling such threats. Around the same time as the Wolf Creek attack, a dozen of other U.S. power plants were breached by adversaries (Riley et al., 2017). While in this case,

Table 3
Comparison between Stuxnet, BlackEnergy and Wolf Creek.

Characteristic	Stuxnet	BlackEnergy	Wolf Creek
Windows Vulnerability	x	x	x
Phishing		x	x
Zero-Days	x (4)	x	
Propagated Internally	x	x	x
Originated on the Internet		x	x

none of the systems that are part of the manufacturing zone were compromised, there was still a severe threat. If one of the infected devices were to be connected to the network controlling the manufacturing zone, the malware could have spread to the ICSs and have caused catastrophic failure. Therefore it is of uttermost importance that all systems connected to the facility are sufficiently protected and monitored. Once malware is spotted in the network, proper acts of mitigation have to be taken as soon as possible to prevent further breaches.

3.4. Analysis of Stuxnet, BlackEnergy and Wolf Creek attacks

As visible within the discussed ICS attacks, there are many ways for adversaries to gain access to a system. Perimeter defence is, although useful and necessary, ineffective against a range of attacks. There is a need for improved security that goes beyond the usage of traditional tools and devices such as anti-viruses and firewalls. These systems will have to adapt to the tools adversaries use, similar to how adversaries adjust to new security mechanisms. Currently, security is continuously catching-up.

Attacks like Stuxnet leveraged an accomplice to physically enter the plant and plug in a USB drive into a system, which circumvented perimeter security measures and allowed the malware to spread through the network without being detected. Due to its approach, the detection became even harder, as there were no abnormal patterns or traffic from outside the facility on the network. Only deep within the network traffic, there would have been evidence of suspicious code being transmitted. BlackEnergy introduced the word to yet another type of malware, a modular form that can be modified for a specific attack.

The impact of BlackEnergy is undeniable, and its continually evolving nature poses substantial security threats. As mentioned before, traditional security software relies on signatures and constant updates when new forms of malware are detected. Continuous monitoring of application behaviour and network traffic provides a certain level of security, but once malware intrudes the network, it can be challenging to remove it altogether. An effective way to limit the potential impact of an infected system is by blocking connections to the Command and Control (C&C) server, either by quarantining the system or by limiting outside communication. Nonetheless, identifying the new malware variants as quickly as possible is imperative in defending against such attacks. The MS Word vulnerability exploited in the Ukrainian power plan attacks was leveraged by using spear-phishing, for which the security lies with the end-user (Hong, 2012). Traditional security measures are generally ineffective against this form of attack, and the infection of the system on which the file is opened is nearly unavoidable. The spread of the infection through the network was, from that moment, imminent. The attack on the Wolf Creek power plant also used spear-phishing to get into the network, but luckily the threat was limited due to other security measures in place. Although if one of those infected computers were to be connected to the manufacturing zone, it would have spread nonetheless. A general overview of differences can be seen in Table 3.

There are several legal requirements companies have to adhere to regarding the security of their systems. One of these is the European Union's NIS Directive which is discussed in the next chapter. Alongside these legal requirements, there are several international guidelines that can be followed and certifications that can be obtained to prove an organisation has taken necessary steps to protect their systems. However, these do not stem from legal requirements they can be used for compliance with the legal obligations.

4. ICS Cyber security standards and guidelines

Despite the increasing cyber threats against critical ICS infrastructures, many ICS operators appear hesitant to adopt security standards and best practices due to increased cost and management overhead (Knowles et al., 2015). Given the criticality of ICS facilities to national security, many governments decided to mandate security measures and regulate their implementation through legislation (Harrop and Matteson, 2015). For instance, the U.S. Cybersecurity and Infrastructure Security Agency (CISA) operates the Chemical Facility Anti-Terrorism Standards (CFATS) (Department of Homeland Security, 2011) to regulate the security of high-risk chemical facilities, while the security of Nuclear Facilities is regulated by the policies of the U.S. Nuclear Regulatory Commission (NRC) (US Nuclear Regulatory Commission, 2010). Such legislation often describes the high-level security requirements and procedures, but not the actual techniques through which security measures should be realised.

In this section, we provide an overview of some of the most comprehensive international ICS cyber security best practises, and we explore where honeypots are included in these standards. We give an overview of U.S., Canadian, Spanish, French and German best practices and give an in-depth analysis of the U.K. Cyber Assessment Framework (CAF). Although these are best practices, several of these guides are used as a baseline for adherence to regulations such as the EU NIS Directive. Afterwards, we introduce several well-known guidelines of organisations that are often referred to within governmental documentations.

4.1. U.K. Cyber Assessment Framework

In an effort to harmonise cybersecurity regulations across the European Union (E.U.), the European Commission introduced the Network and Information Security (NIS) Directive (The European Commission, 2016). The NIS Directive is EU-wide, which means that every E.U. state has to adopt it in their national legislation. Several countries adopted a different strategy for their critical infrastructure (BMI, 2013; NCSC, 2019) since cybersecurity needs differ from other sectors of the economy. It was adopted in 2016, and all members had to transpose it by 2018 (The European Parliament and The Council of The European Union, 2019). The United Kingdom has produced the Cyber Assessment Framework, one of the most thorough implementations of the NIS Directive. Although the CAF is very extensive, it does not discuss the application of honeypots as a defensive technique.

The U.K. Cyber Assessment Framework is compiled by the U.K. National Cyber Security Council (NCSC), to assess the security of critical national services and infrastructure. The framework further notes that "cyber threats to UK CNI represent an area of particular concern for the government, and consequently the cybersecurity and resilience of the thirteen CNI sectors is a high priority for the NCSC." One of the examples is the civil nuclear sector, which has its own cybersecurity strategy (Department for Business Energy & Industrial Strategy, 2017), SyAPs (ONR, 2021) and Technical Assessment Guides (TAG) such as the Preparation for and Response to Cyber Security Events TAG (Office for Nuclear Regulation). Although the EU NIS Directive does not require this, the U.K. still puts emphasis on the importance of the critical infrastructure sector. This

shows the commitment of the U.K. to assess and advise the industry on their cyber security and demonstrate the critical impact potential incidents might have on the country.

The CAF is divided into four main objectives, which are, in turn, broken down in several principles.

4.1.1. Managing security risk

This objective provides companies with information on how to manage cybersecurity risks. It assists them to have appropriate policies, structures and processes in place to mitigate and manage risks to the systems. It is further divided into Governance, Risk Management, Asset Management and Supply Chain.

A definition for risk has to consist of multiple elements, as risk relies on several factors within cybersecurity. The National Institute of Standards and Technology (NIST), one of the leading regulators regarding cyberspace in the U.S., defines it as "A measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence." (NIST, 2012) The International Organization for Standardization (ISO) and The International Electrotechnical Commission (IEC) state within the ISO/IEC 31010:2019 standard (ISO, 2019): "risk is often described in terms of risk sources, potential events, their consequences and their likelihoods." And the UK NCSC defines risk as: "possible future outcomes that we can describe in terms of their chances of occurrence, and what impact they would have on us." (NCSC, 2018) These three definitions put emphasis on the impact or consequence of an event and its likelihood. Reducing risk should, therefore, focus on reducing the probability of an event occurring and the potential impact it might have on the organisation.

4.1.2. Protecting against cyber attack

Within the second section of the CAF Principles and Guidance, the objective is to protect the business against a cyber attack. It aims to have proportionate security measures in place to protect the systems. This objective is further broken down to Service protection policies and processes, identity and access control, data security, system security, resilient networks and systems and staff awareness and training.

When securing a system, the NCSC discusses three main ways vulnerabilities emerge: flaws, features, and user errors. Keeping software up to date is vital in limiting vulnerabilities through defects in the program. As mentioned before, it can be difficult keeping critical infrastructure up to date as powering off the system can have significant ramifications (Cavusoglu et al., 2008). Protection against cyber-attacks does not stop at the security of data and systems but also covers the actions after a system fails or is compromised. Preparations to ensure critical business functions can continue to have to be in place when accounts are compromised, or systems have been infected.

4.1.3. Detecting cyber security events

Objective C of the Cyber Assessment Framework covers the capabilities an organisation should have to detect cybersecurity events. When all security measures fail, the detection of the malicious user and his actions within the network or system as quickly as possible is key (Mukkamala et al., 2005).

An organisation should monitor the security status of its networks and systems in order to detect potential security problems. After the collection of logs and potential security problems, an organisation should use appropriate tools and analysis to detect any indicators of compromise within them. The NCSC outlines the continuous activity required to maintain the security of the organisation and an effective ongoing change within the operational security of an organisation is vital. Apart from monitoring, a proactive approach to discover cyber events is necessary. Flagging deviations from regular interactions, such as users logging in outside of working hours and unexpected traffic should be a trigger for further investigation.

4.1.4. Minimising the impact of cyber security incidents

The final objective set out in the Cyber Assessment Framework focuses on minimising any adverse effects a cybersecurity incident might have on the network or organisation. As incident are almost unavoidable, even with top-of-the-line security in place (Wells et al., 2014), restoring the regular operation of the business is vital to minimise the financial and reputation losses.

When an incident has occurred, aside from reporting it to the regulator, the organisation should take steps to understand the root causes and make sure suitable mitigating actions have been taken. The aim of investigating the cause of an incident is to be able to prevent the root cause on a business wide-scale rather than only patch the affected system itself.

4.2. U.S. NIST Framework for Improving Critical Infrastructure Cybersecurity

The U.S. National Institute of Standards and Technology (NIST) published the first version of the Critical Infrastructure Cybersecurity Framework in 2014 and revised it in 2017 and 2018 to reflect the evolving cybersecurity landscape and incorporate feedback from organisations (Barrett, 2018). NIST is a well-known organisation for standards and guidance, and several of its guides are referenced in the CAF. Although NIST implementation is not subject to enforcement, organisations such as the North American Electric Reliability Corporation (NERC) does provide Critical Infrastructure Protection (CIP) standards that are enforced. Many of these references back to NIST and their Special publications for further details on actions an organisation can take to incorporate the standards. The NIST Cybersecurity Framework is addressed to the organisations that rely on networked devices, including the sectors of Information Technology, Industrial Control Systems (ICS), Cyber-Physical Systems (CPS), and the Internet of Things (IoT). In addition to the NIST Framework, the Department of Homeland security has also published a recommended practice to improve ICS security (Fabro et al., 2016).

The NIST framework focuses on five functions:

- *Identify*: Understanding of business needs, critical resources and risks.
- *Protect*: Implementation of necessary protection mechanisms to safeguard critical operations and services.
- *Detect*: Detection of anomalous activities and attacks, identification of attack targets and methods, and monitoring of external service providers to determine external threat vectors.
- *Respond*: Actions were taken to stop and mitigate the impact of potential security-related events.
- *Recover*: Recovery to normal operations and restoration of any services impacted by a cybersecurity incident.

While the main framework does not specify how the above functions should be implemented, it refers to NIST Special Publications (S.P.s) that provide implementation details. It is also interesting to note the similarity of these functions to the objectives set out within the CAF.

NIST SP 800-53 (Joint Task Force Transformation Initiative, 2013) provides a comprehensive catalogue of tools that can be used to support the cybersecurity functions of federal information systems and organisations. The tools are divided into 18 different categories of security controls that can be used to satisfy the functions of the NIST

NIST recommends the use of honeypots only by specialised entities using non-operational equipment in highly isolated network partitions because potentially misconfigured honeypots can allow attackers to circumvent other security measures through lateral movement attacks. However, NIST SP-800-160 (Ross et al., 2019) on Developing Cyber Resilient Systems recommends maintaining a full-scale deception environment that encompasses honeypots, honeynets and decoy files. According to NIST, deception to create false targets combined with analytic monitoring to detect traffic to those targets could have hindered the 2015 attackers from opening the substation breakers and disrupt power distribution. Such a measure could have been implemented by developing honeypot Human-Machine Interface (HMI) screens integrated with an Intrusion Detection System (IDS) both for the O.T. and the ICS of the power plant. A similar deception strategy could have been effective in misdirecting the malware in the 2016 attacks from providing the intruders with a Command Line Interface (CLI) and interactive services on HMIs.

The NIST guidelines on the proper use of honeypots underline the importance of understanding the complexities of deploying honeypot-based defences in sensitive ICS. While deception and misdirection can be critical in slowing down catastrophic attacks or preventing them altogether, the potential risks necessitate meticulous planning and expertise.

4.3. Other national guidance and regulation

4.3.1. Public Safety Canada ICS Cyber Security: Recommended Best Practices

Public Safety Canada released TR12-002 to provide SCADA and ICS professionals with both administrative and technical best practices related to the cybersecurity of industrial facilities (Public Safety Canada, 2012). Their best practices start by understanding the risks an organisation faces, which includes cyber threats. To this extent, it is important to gain awareness of these threats and which actors are actively targeting the systems. The areas covered are similar to the ones found within the NIST Cybersecurity Framework and U.K. Cyber Assessment Framework.

4.3.2. Spanish National Cybersecurity Institute

The Spanish National Cybersecurity Institute or INCIBE, has published several guides for industrial control systems. One of them covers protocols and network security in ICS infrastructures (INCIBE, 2017), and another one describes the implementation of low-interaction honeypots for ICS security (INCIBE, 2019). The honeypot implementation guide focuses on the requirements and implementation of ICS honeypots. However, it is limited to low-interaction honeypots. We do feel it is a step in the right direction to introduce ICS professionals to the use of honeypots within their environments and hope for increased guidance from all appropriate bodies.

4.3.3. Portuguese National Cybersecurity Framework

The Portuguese National Cybersecurity Centre published their National Cybersecurity Framework to allow organisations to reach a mature level of cyber security (Portuguese National Cybersecurity Centre, 2020). It highlights the same five domains as the NIST guidance: identify, protect, detect, respond and recover. And similar to the UK CAF it identifies subareas such as risk management, asset management, monitoring, detection, response and recovery. Akin to INCIBE, it also mentions honeypots for anomaly detection.

4.3.4. ANSSI Managing Cybersecurity for ICS

The French Agence National de la Sécurité des Systèmes d'Information (ANSSI) published their guide to Managing Cybersecurity for Industrial Control Systems (Agence nationale de la sécurité des systèmes d'information, 2012). The purpose of the guide is to support organisations by providing good practices when implementing security measures. Within the guide, several myths are examined; one of those is that the isolation of ICS devices means they are protected. Similar to the other guides, it mentions areas such as asset management and risk analysis, monitoring and detection, and incident handling.

4.3.5. German Bundesamt für Sicherheit in der Informationstechnik

The German Bundesamt für Sicherheit in der Informationstechnik (BSI) has several resources for ICS security, which includes general recommendations, recommendations for operators and recommendations for manufacturers (Bundesamt für Sicherheit in der Informationstechnik). The general reference guide for ICS security is the ICS Security Compendium (Bundesamt für Sicherheit in der Informationstechnik, 2013), which establishes a general framework for the industrial sector. It acknowledges the mixture of ICS with traditional I.T. systems and the Internet as a significant change in the operation and security of industrial control systems. There is a focus on lack of monitoring, lack of awareness, malware, maintenance laptops and phishing.

4.4. Other well-Known cyber security guides

The following are examples of other guidelines referenced within the CAF that are ICS focused or are generic guides for system security. We have selected these based on their relevance to the potential use of honeypots within the objectives of the discussed national guidance, which we explore in more detail later in this section.

4.4.1. ISO 27001

The ISO 27000 certification range is one of the most known certifications within cybersecurity, and companies often pursue them. Within these, the 27001 covers the information security management aspect. It has been designed to help organisations with the implementation and continuous improvement of their information security management system. When linking back to the CAF, the ISO 27001 standard fits in with objective A (managing security risks).

4.4.2. ISO 27002

ISO 27002 builds upon ISO 27001 and is designed to aid organisation in the selection of controls to implement an information security management system or to guide organisations into implementing standard security controls. It also focuses on the development of information security management guidelines within an organisation. ISO 27002 is referred to for objectives C and B of the CAF.

4.4.3. ISO 27019

As part of the ISO 27000 range of guidance, ISO 27019 is based on ISO 27002 and provides guidance for process control systems that are used within the energy industry. These systems include PLCs, sensors, field devices, advances metering infrastructure and many others that are used to control and monitor processes involving electricity, gas, oil and heat. It is used within objective B5 of the CAF.

4.4.4. IEC 62443-2-1:2010

The International Electrotechnical Commission's 62443-2-1 standard focuses on the establishment of an industrial automation and control system security program. The IEC recognises the weaknesses in ICS due to the adoption of commercial off the shelf technologies, which tend to be more vulnerable to cyber attacks (Jenney, 2013). This standard can be used within objective A and B (Defending systems against cyber-attack) of the CAF.

4.4.5. NIST Information Security Continuous Monitoring (ISCM)

The focus of the ICSM lays in maintaining the ongoing awareness of information security within the organisation; this relates to vulnerabilities, threats, and management decisions. The Information Security Continuous Monitoring publication fits under Objective C principle C1 of the CAF, which focuses on security monitoring.

4.4.6. NIST Computer Security Incident Handling Guide

The Computer Security Incident Handling Guide is aimed at the handling of cyberattacks that even with proper security measures have been able to succeed. Incident handling is vital in reducing loss and destruction and mitigating exploited vulnerabilities. Linking this guide back to the UK CAF, it can be used within both principles of Objective D (Minimising the impact of cybersecurity incidents).

4.5. Honeypots in the context of ICS standards and guidance

As explored at the beginning of this paper, ICS attacks could have devastating effects. In the previous section, we have explored several pieces of ICS security standards and guidance, and this section further explores where honeypots can fit within these. Although we have surveyed several different documents, only NIST and the Spanish National Cybersecurity Institute mention honeypots, briefly. This is an indication that, particularly within an ICS environment, honeypots are yet to be implemented widely. This is surprising as honeypots can fit within many of the objectives outlined and can provide a unique approach to security. Within this subsection we explore how honeypots can fit within areas mentioned within these guidance.

New approaches to the security of ICS that are able to predict and protect against new attacks are necessary. These approaches should rely on real-time data that can be used to detect malicious traffic automatically. This information can then be used to update firewalls, IDS, etc. A possible concept to gather the data is through the use of honeypots. A recent example of the capabilities of honeypots in an ICS environment is the ICS honeypot deployed by Cybereason, which alerted us of the dangers of multi-stage ransomware (Barak, 2020). The goal of the honeypot was to gather information on tactics, techniques, and procedures used by state-sponsored groups. As evaluated previously, there have been publications from governmental organisations which covers the use of honeypots for ICS. However, this guide focuses mainly on the deployment of low-interaction honeypots, which are generally easier to detect. We agree that low-interactive honeypots are generally used within production environments, however, we feel high-interaction honeypots and other honeypots that would generally be considered 'research honeypots' could be beneficial within an organisation as well. Other guidelines from governments or international organisations do not include honeypots specifically, although they can fit within several areas.

In the example of NIST, Fig. 2 breaks down the security functions and sub-functions of the NIST framework that can benefit from the security controls linked to deception and virtualisation technologies related to honeypots. These security controls include *Concealment and Misdirection* to reduce the targeting capabilities of adversaries, and *Information System Monitoring* to detect events occurring both at the perimeter and within the protected information systems. Nonetheless, NIST does not include honeypot-based security controls in any of the three security baselines defined in the framework, even for high-impact information systems in terms of confidentiality, availability and integrity security objectives.

As stated previously, the NIST framework and the CAF have similar focuses, which are also found within the other national security guides we discussed earlier. We can also use honeypots to strengthen several of the objectives mentioned within these documents. An overview of how honeypots can contribute to the CAF can be found in Fig. 3. Unlike the NIST framework, the CAF does not directly mention honeypots. We will use the CAF as a guide to establish where honeypots can fit within regulation and guidance.

4.5.1. Managing security risk

The first section of the CAF, covering security risk, mainly focuses on policies and processes. Although honeypots can undoubtedly be part of these policies and procedures to, deploying them does not satisfy any portion of this objective. Therefore, we will not go in-depth on this objective.

4.5.2. Protecting against cyber attack

The second objective of the CAF and the protect function of the NIST framework cover security measures that are in place to protect the organisation against cyber attacks. Within the CAF, we have identified three sub-objectives within this area where honeypots can be of value. Honeypots can be leveraged within objectives B.2, B.4 and B.5. Due to their nature, they can aid in the detection of unauthorised access (Fabro et al., 2016), as people within the network could potentially try to access the honeypot. Within the system security, objective honeypots can be used to detect and remove malware, identify attacks that exploit vulnerabilities. A segregated network of honeypots can perform these functions without increasing, and can even be leveraged to reduce, the risk to the critical systems. When a honeypot gets targeted, it can also provide information that allows the organisation to respond to changes in risk.

Therefore, we can say that although honeypots, unlike other security systems, are not capable of protecting against attacks by themselves but require an analysis of their data which can then be used within security systems. Once you know who is attacking your network, and how they are doing it, you can more effectively defend against these threats.

4.5.3. Detecting cyber security events

As explored at the beginning of this paper, one of the primary purposes of a honeypot is to lure adversaries to them. Because of this, we can reasonably argue this objective is exceptionally suitable for honeypots. Honeypots inside the network can aid the organisation with the monitoring of the network, as a well-deployed honeypot should receive both adversaries inside the network as well as automatically mitigating malware. Setting up a separate honeypot network with the purpose of capturing threats to the organisation is a clear example of proactive security.

This objective encompasses the main strength of honeypots, their capability to detect security events. Generally, the more data points available, the more comprehensive the data captured is. Detection should therefore not only lie within the operational network, as this means the adversary is already inside the network, but should also include proactive approaches. Honeypots are one of the most comprehensive forms of proactive event discovery, as they can be made to replicate many different systems and entire networks, which allows an adversary to behave like they were inside a real network. This information can then flow back into the previous subsection and help to inform possible actions that could be taken to improve the protection of the network.

4.5.4. Minimising the impact of cybersecurity incidents

After an attack, it is vital to learn how the attacker got into the system so any exploits can be patched. Although honeypots cannot aid with the response and recovery of systems that have been infected, they can provide a wealth of information relating to the incident. This can include further details on the adversary or

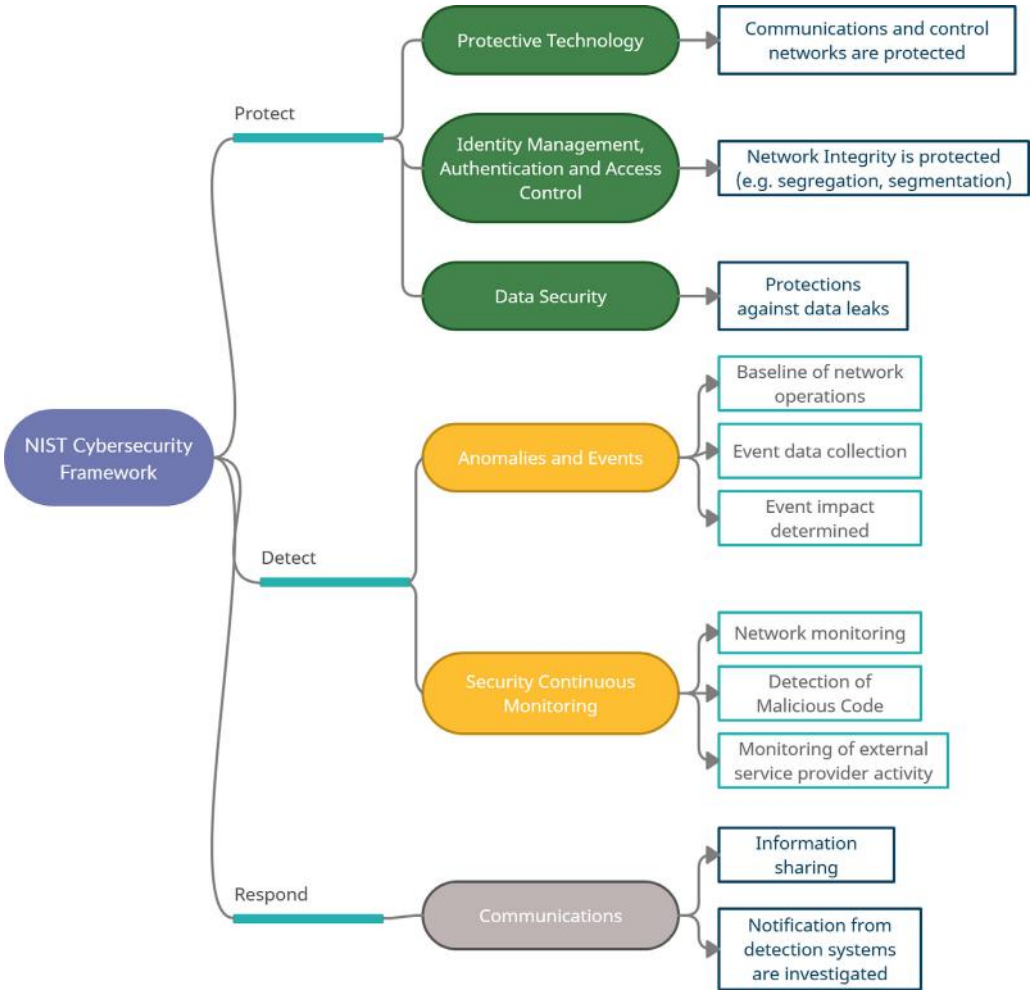

Fig. 2. The cyber-security functions of the NIST framework which can benefit by the deployment of honeypot-related security controls according to NIST guidelines for mission-critical Federal Information Systems and Organizations (Joint Task Force Transformation Initiative, 2013).

even binaries used by the adversary during the attack. Having access to this binary can be of utmost importance. Attacks that happen within the honeypot network can also improve the operational network. Therefore it is essential to learn from attacks happening within this network as well.

A well-deployed honeypot, or network of honeypots, can also be used within digital forensics and incident response training. This further expands the capabilities of honeypots into a training environment and shows their flexibility. Additionally, once an organisation has been compromised, data collected by honeypots can be handed over to appropriate bodies such as law enforcement for further investigation. This means that even a small organisation that does not have the resources or expertise to investigate the logs captured actively can benefit from the deployment of honeypots. Therefore, honeypots are undoubtedly be part of the D2 objective.

5. Existing honeypot implementations

Now that we have covered how honeypots fit within ICS standards and guidance and they can be used within several objectives of the CAF and other guidance, we take a closer look at studies that have been done into ICS honeypots. We aim to look into the real data these studies have gathered and drawbacks of their approach. Several studies have been conducted into honeypot platforms which show that some perform better than others. The main difference between honeypot implementations is which data they can capture, for low-interaction honeypots that are linked to the amount of interaction available to the attacker and the quality of the emulation. When honeypots are poorly implemented, they can easily be identified by more experienced attackers and will therefore not be able to capture data from high-profile attacks or provide useless data (Krawetz, 2004).

Within this section, we give an overview of low-, medium-interaction and high-interaction honeypots. The first subsection has an extensive overview of Conpot and other ICS honeypot implementations. During our research into these implementations, we could clearly see that there is significantly less research into ICS specific high-interaction honeypots than into their low-interaction counterparts.

5.1. Methodology

For the purpose of this survey paper, we have identified a range of ICS honeypots implementations that have been published over the years. Identification has been done by constructing a search query and using it on several academic databases such as IEEExplore and Google Scholar. Following search term was used:

("industrial control systems" or "ics" or "scada") and "honeypot" and "implementation"

The result set consisted of 302 papers; these have then been further distilled by eliminating papers that do not implement the

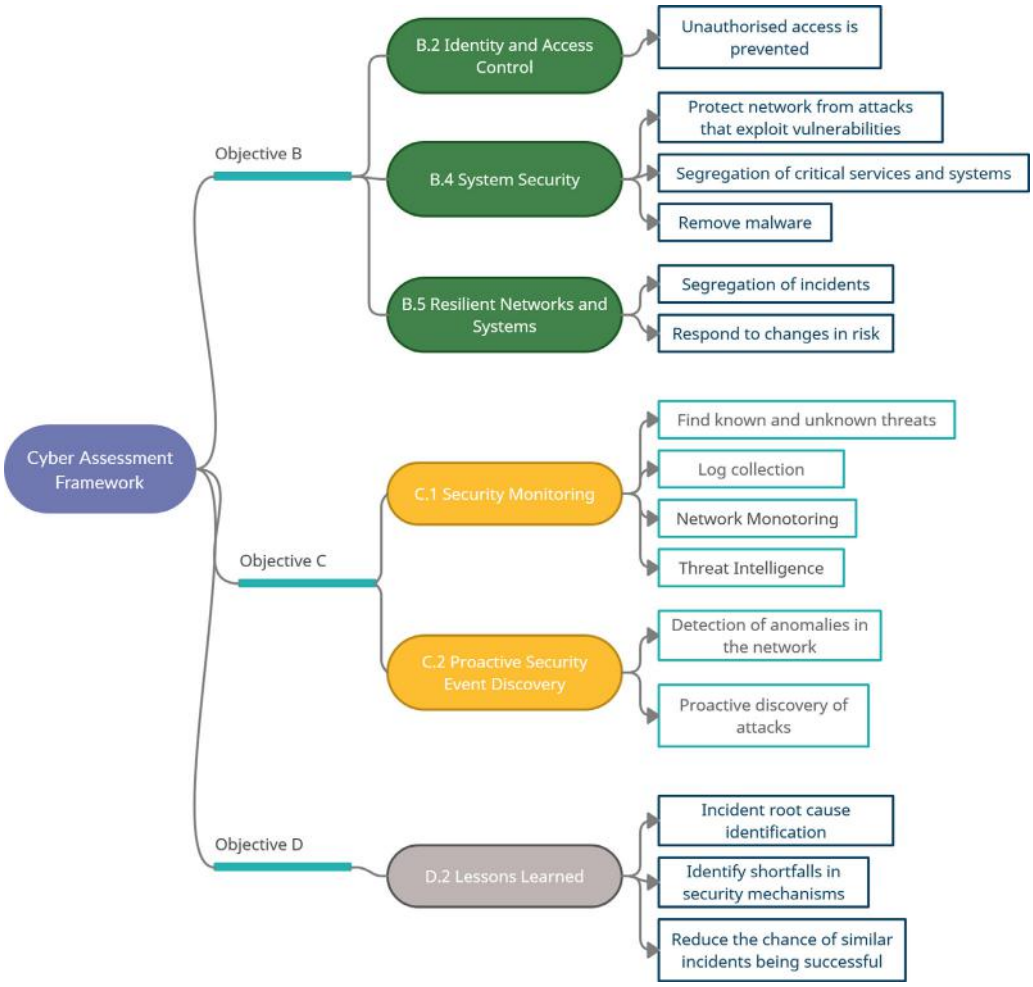

Fig. 3. The cyber-security functions of the CAF which can benefit by the deployment of honeypot-related security controls.

discussed honeypot, are not discussing honeypots in-depth, or are multiple entries of the same paper. This resulted in 60 possible papers. Form these, several papers were not fully accessible, or did not evaluate the honeypot implementation. These steps resulted in 30 papers that were accessible, and implemented and evaluated an ICS honeypot.

5.2. Low- and medium-Interaction

The following 29 honeypot implementations are examples of low- and medium-interaction honeypots we have identified within our result subset. Several implementations use Conpot as a foundation, and others are developed for specific research purposes.

Jicha et al. (2016) performed an analysis of the effectiveness of Conpot by installing it in several separate AWS zones. Within the experiment, all ports were accessible, which would not simulate a real-world environment. We would always suggest configuring honeypots in a similar manner to a production device in order to gain the most accurate information. Overall, 12 Conpot honeypots were deployed, six Siemens S7-200 ICS and six Guardian AST gas pump monitoring system, over five different AWS locations. The authors noted that many more ports, such as 25 and 514, were found to be open through a Nmap scan, which resulted in them to believe Conpot is susceptible to Ubuntu default services. This again leads to an easy method for attackers to detect they are probing a honeypot instead of a real production system.

Another experiment involving Conpot by Kuman et al. (2017) which combined Conpot with OSSEC (a host IDS) and IMUNES (a network emulator) to create a honeynet that simulates an ICS. The combination between a simulated ICS and a simulated network could provide us with the opportunity to have a high-interaction environment with lower risks and without the need for significant investment in infrastructure. During the setup, the authors used the default Conpot template to emulate a Siemens S7-200 PLC and a modified version of the template to mimic an S7-300 PLC which had a vulnerability that was easy to reproduce. The experiment had a duration of two weeks and the bulk of registered activity consisted of port scans. No attempts were made to exploit the S7-300 vulnerability, and the port scans did not result in attacks on the system. The authors identified one possibility of the lack of attacks as the limited time of the experiment, which we noted as well.

The Beijing University of Posts and Telecommunications designed and implemented a more interactive ICS honeypot (Zhao and Qin, 2018) by improving on Conpot by focusing on two aspects, Human Machine Interface and industrial protocols. A simulation is used to provide the honeypot with data and activity to help it disguise itself further. Over 43 days they collected over 13,000 requests, though which they managed to extract 244 IP addresses. Although their focus laid only on the improvement of the S7comm protocol, the study has managed to gain a significant amount of requests. Further improvements could be made, such as the implementation of more protocols and broader deployment of

the honeypot. Overall, we can see movement in the right direction, improving interaction and simulating a production environment to fool attackers.

A Conpot implementation in combination with an IDS and a SCADA implementation was developed and tested by Ponomarev and Atkison (2016). Conpot was used to represent an ICS and pymodbus was leveraged to implement the SCADA part of the ICS. Classification algorithms were used to differentiate between attackers and engineers, which was done at two different client-server separation stages to differentiate between outside and inside traffic. Overall, the purpose of this implementation lays more in the possibility to classify traffic than the perfect replication of an ICS in a honeypot for attackers to attack. Therefore, we cannot scrutinise how well implemented the honeypot is, but we can see where the benefits of a honeypot can lie with regards to data analysing. The IDS achieved an accuracy of 94.3% with no false negatives and 5.7% false positives. This shows a positive trend to implement honeypots combined with machine learning to gather and analyse data in order to warn system engineers when there is a possible attack.

JPCERT, the Japan Computer Emergency Response Team, has implemented Conpot in a honeypot called THS to trace attacks and prevent further infections within the network (Abe et al., 2018). Their design uses Honeyd, to receive and classify packets, and perform actions based on the request it receives. ICS requests are forwarded from Honeyd to Conpot to provide attackers with interaction. The level of interaction, therefore, is limited to what Conpot delivers and does not allow for full PLC interaction. One of their evaluation cases involved the introduction of a computer infected with Havex RAT malware. The malware successfully identified the honeypot, due to the imitative ICS data sent from Conpot. Outgoing scans to the infected PC provided information related to the OS and the open port. THS is able to detect scans directed to the honeypot itself, but would not be able to detect attacks that affect other devices or do not send scan packets. When malware has affected another PLC, which then sends commands to other PLCs, the malicious activity would be revealed by THS by observing commands send within the network. If attackers were to write malicious software to the honeypot, THS would detect the malicious payload and logs created can provide further information to identify the activity. Even actions to overwrite the payload to erase any evidence, would be logged by the honeypot. We view this implementation as promising, and it shows the opportunities honeypots can provide to defend the network. When an infected device enters the network, the honeypot can detect the abnormal activity sent to it and can provide useful information related to malware used to target PLCs. Although, we still have to acknowledge that attackers with a background in ICS would be able to spot the honeypot and likely refrain from interacting with it. Therefore, targeted attacks by experts could still succeed without the honeypot capturing any data related to it.

In an experiment to study the fingerprintability, ways a honeypot can be identified as a honeypot, of ICS honeypots (Zamiri-Gourabi et al., 2019), Zamiri-Gourabi, Qalaei and Azad demonstrate the impact of the flaws found in honeypots by scanning the Internet to detect GasPot and Conpot honeypots. These flaws range from the network delay and performance to further modifications such as keyloggers that can be detected and bypassed by attackers. Further, the limited amount of ICS protocols simulated can flag attackers they are interacting with a honeypot. The primary detection mechanisms are default configurations, missing protocol features, unusual behaviour and the underlying platform. A query ran on Shodan and Censys for one of Conpot its default configurations (PLC name: Technodrome) resulted in 214 hits on Shodan and 185 on Censys. As a side-note, it is possible that many of those are for testing purposes and not actively in use as honeypots. Although it is certainly believable that there are active honeypots amongst those results. Another signature of honeypots was identified as the 'last modified' value of Tue, 19 May 1993 09:00:00 GMT, which returned a combined result of 373 hits. Unusual behaviour is identified as an unnatural pattern; an example listed by the authors included a steady -10% change of a value every hour. Discrepancies in the underlying platform are identified as the support of multiple ICS protocols on a single host when those protocols belong to different devices. The authors state that during their research, they discovered that their method managed to detect GasPots that were configured correctly, which shows that emulated devices can be detected even when not running default configurations. Overall, the tool managed to detect 17 GasPots, and after manual verification, they detected no false positives. This experiment proofs that the configuration and emulation of low-interactive honeypots is key to their operation. When emulation is lacking, protocols are not implemented which can be spotted by attackers and not properly configuring honeypots (keeping default configuration) is a significant error as attackers also know many ICS specific honeypots.

Pliatsios et al. (2019) implemented an interactive ICS honeypot based on Conpot that has the ability to replicate traffic as if it was a real device. Their setup consisted of a real and virtual HMI, a Saitel RTU and Conpot to replicate a Saitel RTU. Both Conpot and the virtual HMI run on a VM. To increase the level of emulation, Conpot uses the traffic data from the real RTU (by feeding pcap files from the real RTU), and the virtual HMI generates requests for the Conpot honeypot. The purpose of the real HMI is to allow an operator to monitor the status of the real RTU. Assessment was done by implementing the honeypot in a real-world hydropower plant. The honeypot managed to emulate the behaviour of the RTU, and the virtual HMI successfully generated realistic traffic. Although the assessment did not include any interaction with the honeypot from an attacker perspective, there were some interesting approaches. Utilising real data to feed into a Conpot honeypot and emulating traffic based on actual ICS traffic undoubtedly makes the honeypot more realistic. The emulation itself would still have flaws, such as not supporting all ICS functions and not allowing for high-interaction; it is a step in the right direction. We could definitely see a similar approach for a honeypot that is situated in a production environment, which possibly could capture malicious data sent to all devices inside the network. It is questionable if an attacker would be deluded that they are interacting with a real device.

A set of Conpot honeypots was deployed by Ferretti et al. (2019) to analyse the interest towards ICS devices on the Internet. The authors note that Conpot is in its default configuration very easy to fingerprint, which lowers their value. Therefore, they have expanded upon and reconfigured Conpot its implementation of ICS protocols to make it more realistic. The implementation was verified by exposing it to Shodan, Shodan did not flag it as a honeypot and gave all instances a Honeyscore below 0.2. Shodan uses Honeyscore to give an indication if a device is likely to be a honeypot and uses characteristics of known honeypots to determine this score, the value is a range between 0.0 and 1 (Shodan). All honeypots were deployed behind a remote endpoint and connected through it over a VPN via a destination NAT rule. This method was used to deploy 11 honeypots. A further 20 honeypots were deployed on the cloud (10 in the US and 10 in Asia). Over the four-month testing period, the honeypots received a total of nearly 5000 connections of which most targeted S7, Modbus or EtherNet/IP. They have captured 1469 distinct IP addresses that specifically targeted ICS protocols and identified 97 distinctive actors amongst them. 17.72% of the IP addresses belonged to Shodan, and a further 6.63% to Censys. Of all the connections made by unknown scanners 60% came from Blackhost, a US hosting service, and nearly 90% of unknown

actors (excluding cloud and hosting services) originated in China. The results of this study reveal that the top 10 actors made 92% of all ICS connections, which show that a handful of actors are responsible for most ICS traffic. Generally, the honeypots received only a handful of different request types, ranging from two to four per protocol. This is generally the case with automated attacks. We can see that the authors put effort into limiting the ability to fingerprint their honeypots, which pays off by gaining a fair amount of connections. The implementation proofs to be useful to gather information on automated attacks, but we feel that the lack of variance in request types show that non-automated attacks are rare.

An evaluation of Conpot is made by Dutta et al. (2020) in which they examine its behaviour towards scanning attacks. A default template of Conpot was deployed inside an organisational network running on an Ubuntu machine. Both Zenmap and Nmap were used to scan for open ports on the machine and gave the same results. Other scanning tools were used but did not manage to produce similar results. None of the scanners was able to list all open ports on the honeypot. Interacting with open ports, such as FTP port 2121, created accurate logs within the honeypot. The authors note that Conpot is not as advanced as a host-based intrusion detection system, but provides adequate results. However, it misses features such as notifying the network administrator when an intrusion occurs. A possible solution to this problem is listed as the implementation of OSSEC, which would complement the logging with extra data. This evaluation of Conpot is relatively limited, mainly because it does not include a real attack on the honeypot itself, but does show that the ICS services can be found by scanning the open ports. The introduction of an intrusion detection system on the Conpot machine itself is a well-thought addition and should, in our opinion, be included in any low-interaction honeypot.

Wang et al. (2019) designed a honeynet-based IDS to capture traffic and study it through machine learning. The system uses several Conpot instances to capture the traffic on the system and extends on Conpot by implementing an intrusion detection module, which uses an SVM trained model to categorise the traffic as malicious or benign. After training the model, it has been implemented into the architecture and verified for accuracy. The model achieved a peak accuracy after being trained with 4% of the data set of nearly 90% with 90s training time and 80s testing time. However, with 1% of the data set training time is reduced to 3s, testing time is at 23.5s and accuracy remains on a similar level of 89.39%. Although we cannot comment extensively on the design of the honeynet, we can say that in a real implementation it would be relatively weak. Conpot, in its default configuration, does not present a wealth of possible interactions or obfuscation necessary to behave like a real PLC. Nevertheless, this paper shows the usefulness of honeypots in a more automated environment. Feeding the honeypot data through a machine learning algorithm and automatically categorising it as malicious or benign will significantly aid system administrators in the security of their network. Although the accuracy of nearly 90% is not perfect, and there is no data given on the false-positives generated, there is still room for further development.

A Gridpot honeypot, which leverages Conpot, to analyse the threats on the smart grid is deployed by Kendrick and Rucker (2019). The honeypot uses GridPot as an open-source honeypot framework, and it uses a honeypot and modelling layer to integrate between GridLAB-D and Conpot. GridLAB-D is a power-distribution simulator that uses algorithms to model and test these systems. The system implements the following protocols for the attackers to interact: HTTP, Modbus, S7COMM, SNMP and IEC 61850. The environment used to deploy the honeypot consisted of an Oracle VirtualBox running GridPot on a Dell XPS desktop with a Windows 10 operating system. A test environment was structured in a similar way but ran on an Ubuntu operating system rather than Windows. The model deployed was the IEEE_13_Node_With_Houses, which consists of 13 nodes and 15 houses. In this setup, a node is a node on the network and a house represents a single family home connected to the smart grid. Conpot was obfuscated to lower the fingerprintability and present as a more enticing target for attackers. The experiment lasted for 19 days with minimal interference, aside from a broken link between Conpot and the modelling layer. In total, more than 9 million packers were captured, of much, the majority consisted of network broadcasts, ARP and other standard traffic. After filtering this data, 1.5 million packets remained. More than 50% of these packets originated from one cloud hosting company located in California, 3.6% from an IP address registered in Russia. Overall, HTTP was the most targeted protocol with Modbus being a distant second. Thirty-nine unique addresses had multiple interactions with the honeypots, ranging from repeat actions to further probing. After analysis with Shodan, the honeypot had a HoneyScore of 1.0/1.0, meaning that according to Shodan there is a very strong possibility the address hosts a honeypot. From the HoneyScore, we can see that the implementation was subpar to other honeypot deployments and did not manage to deceive Internet scanners. As a result, we can safely say that traffic to the honeypot will be of much lower value. Well-trained attackers will generally refrain from attacking a system like this, and automated tools might engage with it depending on the script running. More effort has to be put in the obfuscation of the honeypot, which should result is higher quality data.

In an aim to create resilient cyber-physical systems (CPS) Bou-Harb et al. (2017) propose an approach that unites both cyber and physical environments. The main aim of the study is to leverage real threat intelligence into the security of CPS, and the proposed architecture consists of both a cyber and physical layer. Within the cyber layer, dynamic malware analysis as an active measure and a Conpot honeypot as a passive measure. The physical layer is similar to a generic CPS environment. However, it is extended upon by implementing a CPS monitor to tap, gather and amalgamate data flows and coordinate with the threat detector to react to an attack. A cyber-physical threat detector is implemented and receives data from both physical and cyber layers to monitor all data and detect attacks to the systems. We will focus on the honeypot implementation of this framework. To enhance Conpot and provide a realistic CPS, the authors have implemented further emulation to include CPS protocols. After a one month deployment, Conpot managed to capture about 500 unique attackers which generated thousands of events. Further analysis showed that 10 000 packets contained TCP and UDP scanning attempts and 2000 were TCP DoS attacks on CPS protocols. Two of the three case studies done within the environment included the honeypot as an attack vector. The first one consisted of an attacker attempting a privilege escalation attack on the honeypot by exploiting the HMI session manager. Successful mitigation of the attack was done by blocking traffic originating from the IP address. In the second case study, the attacker exploited the SNMP to gain an overview of all operational services. When this was successful, the attacker generated malicious Modbus request to cause damage to the system. The countermeasure to this attack was dropping the malicious requests and blocking traffic from the IP address that exploited the system. We can clearly see that the honeypot managed to capture a significant amount of data, which could contain valuable threat intelligence, during its short deployment. Further, we can see that attackers trying to exploit the honeypot were successfully caught, and the threat was mitigated. This is a perfect example of how honeypots can be of extreme value when combined with other systems. Although the honeypot implementation was fairly basic, the combination with the other aspects of the system would make it an enticing target.

A honeypot proposed and evaluated by Buza et al. (2014), but we will focus on the more improved implementation of Holczer et al. (2015), is based on a Siemens ET/200S PLC which is emulated on a Ubuntu virtual machine. The PLC generally offers three primary services, STEP7, HTTP(S) and SNMP. Both STEP7 and HTTP(S) services are emulated in one emulated service, and SNMP emulated as a separate service. They deployed the honeypot on a public network in an attempt to obtain real traffic to the honeypot. It is noted that due to the deployment of the honeypot on a university network, attacks explicitly targeting ICS would be rare. After an eight-day test, no traffic was observed on either STEP7 or SNMP ports. Attempts were made to access the SSH ports but were blocked by the firewall. No specific PLC attacks were detected. The one-month extended test produces similar results, except for a limited amount of traffic to the STEP7 port. A second short test, again, provided no different information than the other tests. This implementation provides us with interesting information. The use of a university network has a negative impact when trying to entice attackers and make them think they attack a real organisation.

Serbanescu et al. (2015) deployed a research ICS honeypot over the public Internet using the Amazon EC2 cloud. They used software emulations of ICS/SCADA devices. However, they admit the goal of the emulation was not to mimic the devices perfectly but rather obtain information on the overall threat landscape. We would argue that to gain an in-depth and accurate overview of the threats to ICS devices. One should portray their honeypots as closely as possible to the device they want to mimic. Especially within the ICS environment where knowledgeable attackers closely scrutinise the devices, they attack. The deployment within the Amazon cloud infrastructure can also result in less valuable data, as attackers should be aware that PLCs would not be deployed in those environments. We agree with their statement that the cloud provides benefits in terms of scalability, but efforts have to be made to obfuscate the use of the cloud. It could also be argued that the deployment of multiple devices within one network, instead of some spread over several Amazon EC2 regions, can aid to entice attackers as a real production system would have multiple systems such as PLCs as well. Eighteen honeypots were deployed over eight EC2 regions, with most regions having either two or three instances. Overall, 1092 Modbus connections and 1040 requests were made, around 22% of both originated from Shodan, and more than 70% of other attacks originated from one other server. All IEC-104 requests and 23 out of 34 port scans were conducted by Shodan as well. These results show a lack of real threats probing and attacking the honeypot deployments.

Jaromin (2013) designed and implemented an industrial control emulator that acts as a decoy field device. The PLC emulator is implemented on a Gumstix single board computer running a Linux distribution and emulates a web service on port 80, Modbus on port 502 and a host automation products (HAP) protocols service on port 28784. Several iptables rules have been set up to filter and queue the packets. Due to the focus on higher-level protocols, only web and HAP protocols were evaluated for an accurate representation of the target PLC. In general, we see that efforts have been made to emulate a PLC accurately, but there are multiple shortcomings to make us confident that the honeypot is a detailed representation of a PLC. The results showed a high level of packet-level accuracy for the implemented services; the honeypot also performed well in the scenarios conducted to evaluate the accuracy at both scanning and attack levels. An area where the emulator fails to perform similar to the PLC is response times, which were more than 98 times slower for some workloads on the Gumstix compared to the PLC. In standard honeypot scenarios, the slower response of the Gumstix was negligible. Based on this research, we can identify a possible well-performing honeypot if further development is conducted.

The Symbolic Cyber-Physical Honeynet or SCyPH was presented by Redwood et al. (2015). It is designed to entice attackers, enhance the screening and coalescence of attack events and more. They emulate cyber-physical systems (CPS), implement a SCADA HMI, and provide logging and anomaly detection. The primary target is the HMI, which is integrated with the CPS and uses a web interface to allow interaction. The whole framework is modular and allows for the implementation of other systems. All actions taken on the HMI are represented on the CPS so that the attacker can see the result. Gridpot was used in conjunction with the framework to demonstrate its capabilities. MMS, GOOSE and Modbus protocols in IEC 61,850 were emulated as services for the attacker to interact with. An attack was launched with specifically written malware, which successfully changed the state of an intelligent electronic device switch in one of the simulated substations. One interesting action taken by the authors was the exploitation of a vulnerability within the HMI software, which shows the importance of correctly emulating software within honeypots as attackers would have knowledge of such exploits and actively try to use them. All actions and binaries used within the system are logged and available for a forensic replay of attacks. This framework is promising and adaptive due to its modular nature. For proper emulation of devices, a significant amount of work has to be done, and non-discovered exploits would not be able to be implemented. This might allow attackers to fingerprint the honeypot. In general, we view this framework as promising but note the hard work needed for proper implementation.

To investigate the extent of malware and attacks targeting critical infrastructure, Berman and Butts (2012) presented a honeypot based on Gumstix technology which emulated an ICS field device. The Gumstix was programmed to support Modbus communication following RFC specifications to incorporate standard function codes. When the honeypot receives a message, it will respond in an appropriate manner, e.g. valve closed. Unrecognised function codes are responded to by an unrecognised function code error code. All non-Modbus traffic (any port aside from 502) is logged but not responded to. Nmap scans returned the expected results but failed to recognise the honeypot was running on a Linux OS. This approach has its merits and provides a low-cost and low-maintenance PLC honeypot, but may not be effective against attackers with ICS knowledge, which the authors also acknowledge. It could be useful to capture scanning attacks and malware that automatically propagates throughout the network aiming to infect PLCs. To accurately implement all the necessary function codes to respond to Modbus communication, the configuration would require a significant amount of time. It has to be evaluated in the time spend to implement the honeypot is worthwhile compared to the information it would capture.

You et al. (2019) explored the use of honeypot data to characterise Internet-wide automated ICS attacks. To this regard, they have implemented a minimal interaction honeypot, MirrorPot. The honeypot generates a response based on the request, but the honeypot never parses the incoming requests. Their goal is to capture the actions of automated scripts, not specific attacks executed on the devices themselves. They deployed seven instances over the world hosted on static ISP addresses with 26 ports running ICS services. The most extended deployment lasted for 477 days, the shortest for ten days. Five out of seven honeypots were active for less than 40 days, which we deem insufficient to gain usable long-term threat intelligence. For example, it can take Shodan several days to fully index devices, with the initial scan potentially taking days (Bodenheim et al., 2014). At the end of the experiment, 2.6% of all requests (56 643 490) contained payloads, and only 5.3% of attacks were targeted at ICS-related ports. Less than 20% of IP addresses were spotted more than twice, which shows that the number of targeted attacks was low. Although the experiment resulted

in some impressive results, such as more than 20 common ICS-related attack patterns, we still believe that the experiment overall lacks on the implementation side. Many ICS attacks are targeted, and attackers with knowledge of ICS easily spot weak emulations. Due to the short lifespan of most honeypots (less than 40 days), we feel that there was not enough time for those honeypots to be appropriately indexed and scanned. There is also a discrepancy in the data, showing 477 days as the most prolonged duration, but the authors claim a 418 days duration. Further, they claim they ran seven honeypots for 418 days when the majority of those ran for less than two months.

Mimepot (Bernieri et al., 2019) is, according to its developers, a cyber-physical honeypot that is able to simulate physical processes and leverages SDN to provide a future-proof approach. It consists of two modules, Mime Plant and Mime E&C. MimePlant is the network node which simulates the PLC, whereas Mime E&C simulates a SCADA workstation which regulates the plant behaviour. Both modules have to be implemented separately, so they are able to produce network traffic similar to a production network. SDN is used to redirect malicious traffic to Mimepot, and to obfuscate the IP addresses of the real devices to fool the attacker into thinking they are attacking the real device. Their evaluation consisted of attacking Mimepot and changing the configuration. Attacks are successfully redirected to the honeypot, and attackers are able to see changes based on their requests. The introduction of SDN shows an excellent example of how other techniques can benefit honeypots, the redirection and obfuscation could fool an attacker to a certain level and protect the infrastructure. We believe that the general evaluation is lacking as the attacks were not carried out by unknowing attackers. Therefore they do not correctly replicate a real attack. The authors know what functions are supported by the honeypot and would have shown a different approach. More research has to be done into the robustness of Mimepot, as it is not clear how in-depth the implementation of the ICS protocols is.

HosTaGe (Vasilomanolakis et al., 2016) is a honeypot developed by Vasilomanolakis et al. and is categorised as a lightweight low-interaction honeypot for mobile devices aimed at detecting malicious devices. They have adapted HosTaGe to support ICS networks by supporting the emulation of ICS protocols (Modbus, S7, HTTP, SNMP Telnet, 5MB and SMTP). The implementation of these protocols looks extensive, but we still doubt that the implementation would have been perfect and therefore, attackers could notice missing features, vulnerabilities and others. HosTaGe has a built-in detection mechanism to detect attacks directed at a single protocol, multiple protocols (based on the same source) and based on the payload sent to the device. It captures the packets and connection requests it receives and generated signatures when it detects an intrusion. Those signatures can be sent to a Bro IDS. In comparison with Conpot, HosTaGe generally received more traffic over the same period on all protocols except Modbus and was also able to detect unique malicious IP addresses. We would like to point out that the experiment was set up without any firewalls between the honeypots and the Internet, which would be a significant red flag for experienced attackers. However, the goal of the experiment was to identify automated attacks towards ICS devices and therefore, it might not have made a significant difference. The authors note that Shodan was able to scan the honeypots during their tests but only managed to identify Conpot as a honeypot. This approach presents us with an interesting aspect, the generation of signatures and evaluation of malicious data on the honeypot, which is subsequently sent to a connected IDS. The purpose of honeypots is to capture data. Because of this, they are only as useful as their data and how the data is used. When data can be evaluated in real-time, it is at is the most-valuable point, as administrators are then able to mitigate an attack while it is happening.

iHoney (Navarro et al., 2018) is an ICS honeypot designed by Navarro, Balbastre and Beyer to mimic a real ICS infrastructure as close as possible. The authors decided to simulate a water treatment plant, and therefore the design of the infrastructure followed the same process as it would have done in an actual plant. iHoney consists of three modules, the ICS system, simulation system and monitoring infrastructure. The ICS system consists of a SCADA server, PLC control network, and the associated ICS protocols. The plant is simulated within the simulation system to generate realistic outcomes in real-time and allows for interaction as a real plant operator would have. Monitoring is done by a Network Intrusion Detection System (NIDS) and Host-based Intrusion Detection System (HIDS) on the network and exposed SCADA server respectively. To further lower the fingerprintability, the HIDS is hidden behind a legitimate program which checks the communication status of the PLCs. The authors note that several concessions had to be made; for example, they had to be a balance between the complexity and realistic simulation of the system. This shows one of the main drawbacks of low-interaction honeypots, but this emulation looks well-thought and well-implemented. The honeypot was exposed on the Internet for over two years and was attacked on a daily basis. Most of the attacks were automated, which we would expect on a low-interaction honeypot, and some attacks involved social engineering, which is a sign of advanced attackers. Between July 2015 and September 2016, the initial exposure period, monthly IDS alerts mostly ranged between 1000 and 10000. However, August, September and October peaked at more than 600 000 alerts monthly (with August generating over 1.3 million), the authors identified a problem with the remote desktop protocol used which was substituted with a Virtual Network Computing service afterwards. We can see a considerable amount of data gathered by iHoney, which looks like a well-designed honeypot. The capturing of social engineering attacks show that attackers were fooled by the honeypot and dedicated time to infiltrate it, which proofs the detailed design and implementation was beneficial.

In an attempt to discover which actors are scanning the Smart Grid, Mashima et al. (2019) deployed five low-interaction honeypots on geographically different AWS instances. Their emulation was relatively limited and included TCP listeners and simple server programs for IEC 60870-5-104 and IEC 61,850 emulation. It was verified that Shodan did not categorise the systems as honeypots. This shows a lack of vetting on Shodan its part, as the limited amount of interaction and services in combination with the deployment on an AWS IP address should be a significant red flag. The honeypots were active for over six months and captured 6 GB of ICS network traces. Weekly packet count was in the same range with the exception of some spikes due to some of the honeypots receiving a significant amount of traffic for a short period. Multiple attacks showed an identical approach, which leads to the conclusion that a similar tool must be used. In our opinion, this is also an indicator of automated attacks with limited intelligence behind it. The honeypots received a denial-of-service attack, and DNP3 and Modbus TCP scans. An interesting result of this experiment is that there is little correlation between the daily traffic intensity for the different geographical locations. This could be an indicator that scanning tools scan the Internet in geographical segments, but this brings us little more information unless further analysis is conducted. Overall we think this implementation is lacking effectiveness and the data analysis provides little results to be used to increase ICS security.

A honeynet that is capable of emulating an entire smart grid field communication infrastructure was designed by Mashima et al. (2017). Substations are simulated on a virtual machine connected to a substation LAN, and OpenMUC is used to implement the communication interface. The substation network is emulated through Mininet, which runs on its own virtual ma-

chine. Mininet virtual hosts are camouflaged by making them look like IED devices, for this all packets are forwarded to a SoftGrid virtual machine. Physical components of the power grid are simulated on a PowerWorld simulator. Traffic latency is close to a real implementation, and frequency changes are handled dynamically to emulate realistic changes to attackers. The honeynet is deployed on an Amazon IP address and utilises a ring topology as it is most frequently used. Fingerprintability was tested through means of Nmap to detect the operating system, and deployment of multiple virtual machines on the same hosts to evaluate the effects a high-load on one machine has on the others. The authors note that running multiple virtual machines on the same hardware can lead to the discovery of the honeypot and potential mitigation for this is to host virtual machines with their own dedicated processor core. It does not seem that the implementation was utilised for data capturing purposes. We can clearly see the authors aimed to create a honeypot that is comprehensive and difficult to detect. Implementing a honeynet, instead of a single honeypot, dramatically increases the interaction and emulation, which in turn should be more enticing for attackers. A statement was made that the deployment on an Amazon IP address does not increase the detectability of the honeypot, we disagree with this statement as real ICS infrastructure would not be deployed in the cloud. The use of AWS may result in data from automated attacks, but these are generally of lower complexity than targeted attacks. Our advice is to deploy ICS honeypots in an IP range that would generally see ICS deployments.

IMUNES is a well-known network simulator and was used by Haney and Papa (2014) to simulate a SCADA honeynet. The honeynet framework consists of two areas, the honeynet and the organisational network, and the former includes a honeywall and the latter a standard firewall. Both areas are therefore strictly separated. Within the honeynet, there is a system dedicated for log collection and an IMUNES on FreeBSD Cluster running virtual PLC/RTU nodes. Each of those nodes runs a JAMOD PLC simulator, honeyd and Sebek for data collection. JAMOD is a Java Modbus library that supports TCP, UDP and serial connections and is used to emulate the PLC functions. To further enhance the emulation honeyd is leveraged to simulate other operating systems and services such as a login shell, and a management interface over HTTP(s). Snort is deployed to monitor and capture traffic within the honeynet, and generate alerts when signatures match a known attack. Sebek is deployed within the honeypots to capture keystrokes and system interactions, to further analyse how attackers are exploiting the system. This proposed framework presents us with some interesting approaches. First, using IMUNES to simulate the honeypots and second to emulate a PLC through Java. Using IMUNES adds an extra level of virtualisation that, in our opinion, can increase the fingerprintability of the honeypot and we would encourage anyone implementing honeypots to refrain from adding layers of virtualisation when it is not necessary. Especially when working within ICS environments. Although Java is an excellent programming language, in this case, the authors are doing something difficult when there is a tested alternative available. Honeypots like Conpot are designed to emulate ICS devices and used in a wide range of implementations already. Further experiments have to be done to verify the capabilities if the honeynet performs well, it would open possibilities for large honeynets with a small footprint.

TrendMicro (Wilhoit and Hilt, 2015) implemented a GasPot honeypot to evaluate the attractiveness of SCADA honeypots for attackers. GasPot is designed to not look like a honeypot. Every deployed instance is unique, which makes it harder to fingerprint. It supports six commands and allows users to change values in the simulation. They deployed GasPot instances on physical IP addresses (no cloud services) in seven countries: US, Brazil, UK, Jordan, Germany, UAE and Russia. Some deployments were designed to be detected by Shodan to collect data on automatic attacks. After deployment, they discovered a Pastebin post where people shared information about the vulnerable instances deployed in this research. This is very interesting and shows that there is a real community amongst hackers. Attackers managed to exploit the honeypots, change their names and perform other actions. Looking at the source of the attacks, they were spread all over the world. The implementation shows a lot of promise as useful data was captured and presents possible avenues such as purposely leaking information about a honeypot within hacker communities to gain more data. All the honeypots were configured to resemble a real device, and although these were emulated, the attackers did thoroughly engage.

In an effort to evaluate the efficiency of honeypots in the detection and evaluation of advanced threats, Wade (2011) implemented a test system based on the Digital Bond framework. The honeypot is deployed on the university network, behind a Honeywall, and a Nmap scan correctly identified the services running on the open ports (21/tcp, 80/tcp and 502/tcp). The services running on the honeypot include HTTP, FTP, Telnet, SNMP, VxWorks Debugger and Modbus. VxWorks Debugger is a real-time operating system for embedded systems and designed to run on top of another operating system. HTTP, FTP, SNMP and Telnet services are partially implemented to lower the vulnerability of the system. On the network, Snort is deployed for intrusion detection. The honeypot was running for 38 days, with seven days non-consecutive of malfunctioning resulting in 31 days of data. Nearly 2 million Snort alerts were generated over the deployment period, the majority of the alerts (97%) were generated by UPnP malformed advertisements which only affects older Windows systems. Of all other alerts, the majority were UDP port scans and port sweeps (2.8% of total alerts). The experiment concluded that there was no specific interest in the SCADA services, and most attackers were not aware of what operating system was running. The first area we would look in an aim to discover there was no ICS specific interest would be the deployment of the honeypots on a university network, which are well known for hosting research honeypots and not known for hosting ICS devices. Further, the emulation was adequate and should have presented itself in a similar way than other low-interaction ICS honeypots.

Antonioli et al. (2016) have proposed a realistic virtual ICS honeypot that, according to them, allows for high-interaction by the attacker. We can see that the honeypot mimics a real ICS architecture near perfectly, with a simulated network environment, PLC, HMI and processes. Within the honeypot, they allow the attacker to fingerprint the device (e.g. Nmap, xprobe2) and obtain necessary information, including IP addresses and ports, and protocols used. There is no prevention for the attacker to obtain escalated privileges on the system, but the authors have limited their attacker model only to include reasonable scenarios. The implemented honeypot is able to capture Man-in-the-Middle attacks, port scanning and DoS attacks amongst others. Within this honeypot, we can see the benefits of emulating a real environment, as attackers have more ways to interact and engage. Overall, we can see this is a comprehensive implementation of a honeypot that should be able to fool even higher level threat actors. However, some decisions such as the use of weak SSH credentials and plaintext telnet authentication might be a red flag. Nevertheless, we would not call this an actual high-interaction honeypot, based on its virtual nature, but it does result in comprehensive attack data. We would like to see a similar honeypot based on real devices to achieve true high-interaction, with a possibility to emulate the processes and network.

Simões et al. (2015) propose an emulated PLC honeypot that is situated between operational PLCs (physically or logically). It aims to divert attackers to attack it and actively report on the suspicious

activity targeting it to the IDS. The honeypot focuses on Modbus and runs both simulated and complete services that are generally found on a PLC. Central to this honeypot is the Modbus API simulator, which provides the necessary Modbus functionality. Other modules included are FTPD, SNMP and a module to detect probing activities (port scans) on the other TCP/IP ports. The location of the honeypot is undoubtedly beneficial in our opinion as they would be in an environment that has real PLCs and the honeypot should be in a prime position to capture malware that is propagating throughout the network. Due to the programmability and configurability of Modbus API, they are able to mimic a wide range of real PLCs and provide unique behaviour. There is also an event monitor integrated within the honeypot to analyse the data captured. A module for remote management is also included to allow security staff to monitor the honeypot and allow for remote actions. To prevent the attacker from using the honeypot as an attack vector, a firewall is implemented to deny connections from the honeypot to the other systems but allow all incoming connections. We can see this is a comprehensive emulation and provides a lot of useful features. The location of the honeypot is one of its key strengths, especially with the lowered risk to other systems due to the firewall. Because there is no evaluation, we cannot comment on the attractiveness of the honeypot and if attackers would be able to detect it.

In an attempt to develop a new medium- to high-interaction PLC honeypot Lau et al. (2016) developed XPOT to simulate a Siemens S7 314C-2. They achieved high interactivity by supporting more than 100 MC7 different instructions and allow an adversary to load PLC programs on the honeypot. The authors note that due to the fact that XPOT has to compile the bytecode, adversaries can easily spot the delay in execution compared to a genuine Siemens PLC. An evaluation of the honeypot was done by allowing six participants of a PLC programming and hacking course to distinguish a real PLC from XPOT. Although participants were not always sure what the correct behaviour of the PLC should be, all participants correctly identified the XPOT when given access to all tools and features. We can be reasonably sure that experienced attackers would be able to identify XPOT as a honeypot without many issues. This, again, shows the need for a near-perfect implementation and the benefits of using a real PLC as a honeypot to trick attackers into believing they managed to penetrate a real device.

5.3. High-interaction ICS honeypots

The following three honeypots honeypot implementations are high-interaction honeypots we have identified within our result subset. High-interaction ICS honeypot implementations are clearly less common than low-interaction variants, which can be explained by the high costs of equipment, and the level of knowledge required to deploy and maintain them.

In a recent research paper by Trend Micro (Hilt et al., 2020), a fake factory consisting of honeypots was set up to attract and capture real threats. Within the company, there were cellular routers, protocol gateways, servers and HMI (virtual), and physical industrial control systems. Four PLCs were implemented, one Siemens S7-1200, two Allen-Bradley MicroLogix 1100, and one GE Fanuc. They also included a Phoenix ContactILC 131 inline controller. One comment we have to this approach is that it is unlikely for companies to implement many different brands of PLCs which, for us, would be a clear indicator that the network might consist of honeypots. Nonetheless, the honeypots managed to gain attention and capture comprehensive attacks which resulted in system shutdowns, fraud and more. We can see quite a bit of non-ICS related activity, such a fraud, crypto miners and ransomware. ICS specific attacks are generally attackers playing with the HMI and the factory infrastructure. This might be linked back to our previous point. The implementation does show that even when we set up a comprehensive ICS honeypot, we have to take into account other actors and their malicious intents.

Aside from an emulated PLC honeypot, Simões et al. (2015) propose a high-interaction honeypot architecture where the attacker interacts with a real PLC. However, it is not linked with any industrial processes. All traffic to the PLC is forwarded to an IDS, which is more accessible due to the unencrypted nature of the Modbus protocol and generates security events. The main advantage the authors' list is the implementation of the real infrastructure for the attackers to interact. The cost and complexity of the implementation, especially when there are multiple honeypot deployments, is listed as one of the main disadvantages. Real PLC honeypots make it harder for attackers to spot the honeypot if they never interact with the monitoring systems themselves. Attackers can interact with the PLC in the same ways they would when they attack real production infrastructure, which lowers the suspicion they might be targeting a honeypot. The architecture is proposed in limited form and not evaluated.

An operational technology honeypot designed and implemented by Piggin and Buffey (2016) consisted of four major components: control systems and process simulation, situational awareness and forensics platform, the attacker's infrastructure and the remote monitoring infrastructure. They have designed a high-interactive honeypot that allows the attacker for detailed interaction with the honeypot and specifically designed it for forensic investigations. The honeypot was designed to attract attackers and capture valuable, for which they developed an application to resemble real automated processes. After deployment, the honeypot managed to capture ICS attacks related to the disruption of PLC data communications, an anonymous attack against the PLC originated from the TOR network and password attacks using default vendor credentials to delete directories on the SCADA PC, amongst others. This implementation shows the need to lure attackers to honeypot implementations that closely resemble production systems.

5.4. Summary of discussed implementations

We can identify the essential characteristics of ICS honeypots that are important to adhere to in order to capture valuable data. Looking into the low-interaction honeypots, we can see that the data gathered by them is of relatively low value. The data captured by these honeypots are generally limited to Internet-wide scans or initial reconnaissance. When low-interaction honeypots are connected to the Internet, they can be used for high-level threat intelligence purposes and to gain information on how ICS devices are scanned on the Internet, which can enable organisations to limit the exposure of the organisation to these scanners. Within the network, they could be able to spot automatic propagation of malware that has already managed to infect a device on the network. However, when deploying low-interaction honeypots for any purpose it is important to obfuscate the default signatures that may be included in the platform used. For Conpot, the deployment that is most popular, there are a multiple of signatures we have evaluated in previous work which range from signatures on the HTTP emulation to information seen on the S7Comm protocol (Maesschalck et al., 2021). Due to the limited information that is generally available within the papers we have evaluated we cannot provide an in-depth overview of the fingerprintability of the platform itself. Which is why, within this survey, we have mainly focused on the environment the honeypots were deployed in.

High-interactive honeypot deployments are rarer than low-interaction ones. This is mainly because of the higher cost, maintenance and development time. Nevertheless, they are able to provide a more realistic environment for attackers to exploit. Because

Table 4
High-level Overview of ICS Honeypot Implementations. This is a high-level overview of some of the protocols implemented within the discussed honeypots, this list is by no means exhaustive and could contain errors. Some implementations do not specifically discuss all implemented protocols and these have been deducted from background knowledge.

Author(s)	Interaction	HTTP(S)	Telnet/SSH	(T)FTP	SNMP	Modbus	IEC-104	IEC 61850	S7Comm	Data Captured
Jicha et al., 2016	Low	✓	✓	✓	✓	✓			✓	Basic
Kuman et al., 2017	Medium	✓		✓	✓		✓		✓	Basic
Zhao and Qin, 2018	Low	✓		✓	✓	✓			✓	Basic
Ponomarev and Atkison, 2016	Low	✓		✓	✓	✓			✓	Basic
Abe et al., 2018	Medium	✓		✓	✓	✓			✓	Basic
Pliatsios et al., 2019	Medium	✓		✓	✓	✓	✓		✓	No Eval.
Ferretti et al., 2019	Low	✓	✓	✓	✓	✓	✓		✓	Basic
Dutta et al., 2020	Low	✓		✓	✓	✓			✓	-
Wang et al., 2019	Low				✓	✓			✓	No Eval.
Kendrick and Rucker (2019)	Low	✓			✓	✓	✓		✓	Basic
Bou-Harb et al., 2017	Medium	✓			✓	✓			✓	Intermediate
Holczer et al. (2015)	Low	✓			✓				✓	Limited
Serbanescu et al., 2015	Low	✓	✓	✓	✓	✓	✓			Basic
Jaromin (2013)	Low	✓	✓	✓						No Eval.
Redwood et al., 2015	Medium					✓		✓		Intermediate
Berman and Butts, 2012	Low					✓				-
You et al., 2019	Low		✓		✓	✓				Basic
Bernieri et al., 2019	Medium					✓				Intermediate
Vasilomanolakis et al., 2015	Low	✓	✓	✓		✓				Basic
Navarro et al., 2018	Medium					✓			✓	Intermediate
Mashima et al., 2019	Low					✓	✓	✓		Basic
Mashima et al., 2017	Medium		✓				✓	✓		No Eval.
Haney and Papa, 2014	Medium	✓	✓			✓				No Eval.
Wilhoit and Hilt, 2015	Low									Advanced
Wade (2011)	Low	✓	✓	✓	✓	✓				Limited
Antonioli et al., 2016	Medium	✓	✓		✓	✓				Comprehensive
Hilt et al., 2020	High	✓	✓	✓					✓	Comprehensive
Simões et al., 2015	Medium			✓	✓	✓				No Eval.
Piggin and Buffey (2016)	High	✓	✓		✓	✓				Comprehensive
Lau et al., 2016	Medium				✓				✓	Basic

✓: one or all of the protocols listed in the column are implemented. Limited, basic, intermediate, advanced and comprehensive relate to the quality of data captured related to the attacks on the system E.g. no specific ICS data would be classified as limited and extensive ICS interaction on the honeypots would be classified as comprehensive.

attackers can perform the same actions as on real systems, they are less likely to notice they are attacking a honeypot and are more likely to use all the tools they have to attack it. This should result in higher-value data, as we can see from the University of Toulouse experiment. The main downside, aside from cost, is the risk they pose when they are successfully exploited. When this happens, the attacker can use the system to exploit other devices on the network. If the honeypot is deployed within the same subnet as the operational network, the production devices can be compromised as well. Therefore, we would advise not to deploy high-interaction honeypots within an operational environment or internal business network but to deploy them in an isolated network within the IP range.

When looking to deploy honeypots, we can find some foundational recommendations in existing literature (Dodson, 2020), we build upon this with the following suggestions. The aim of the configuration should be to mimic a real device as closely as possible. This does also include how the honeypots are portrayed to the outside. From the discussed implementations, we can see that honeypots deployed on university or AWS IP addresses captured less valuable data.

Because of the specific knowledge required to exploit ICS devices successfully, we believe that the people who are targeting these devices are generally more aware of how these devices are implemented. This also links back to the necessity of a realistic configuration and functionality of the device. To extend on the honeypot itself, further development and the implementation of more data collection/monitoring systems is encouraged. Most monitoring systems should be implemented within high-interaction honeypots by default (e.g. HIDS and NIDS), and these can improve data collection on low-interaction honeypots as well.

We would argue that because low-interaction honeypots generally capture less valuable data, it is crucial to extract as much data from them as possible. The deployment of monitoring systems should always accompany the deployment of honeypots. These systems should be implemented both on the honeypots themselves and around them. An overview of the discussed honeypots and their characteristics can be found in Table 4.

5.5. Enhanced honeypot classification

Based on this study on the level of interaction of honeypots, we can see that high-interaction honeypots provide a more significant data set than low- and medium-interaction ones. What we do observe is that emulated honeypots that are deployed in a more interactive environment do provide more data than stand-alone or weakly implemented ones. With this research as a foundation, we provide a new form of classification for honeypots. Aside from the standard low, medium and high classification of the honeypot itself, we focus further on the environment where it is deployed. This environment that we call a honeynet (Spitzner, 2003) should be described with the same categories to avoid the usage of many different categories. Honeypots, disregarding their level of interaction, can be situated within a low-, medium- or high interactive network. This is determined not only by the number of honeypots in the honeynet, but also the number of different services, and how closely it represents a real organisational network. For example, deploying multiple high-interactive PLC honeypots in an environment with HMI (Human-Machine Interface) honeypots, sensors, and other systems commonly found within these networks would be a high-interaction honeypot within a high-interactive honeynet. A limited amount of high-interaction PLC honeypots in a

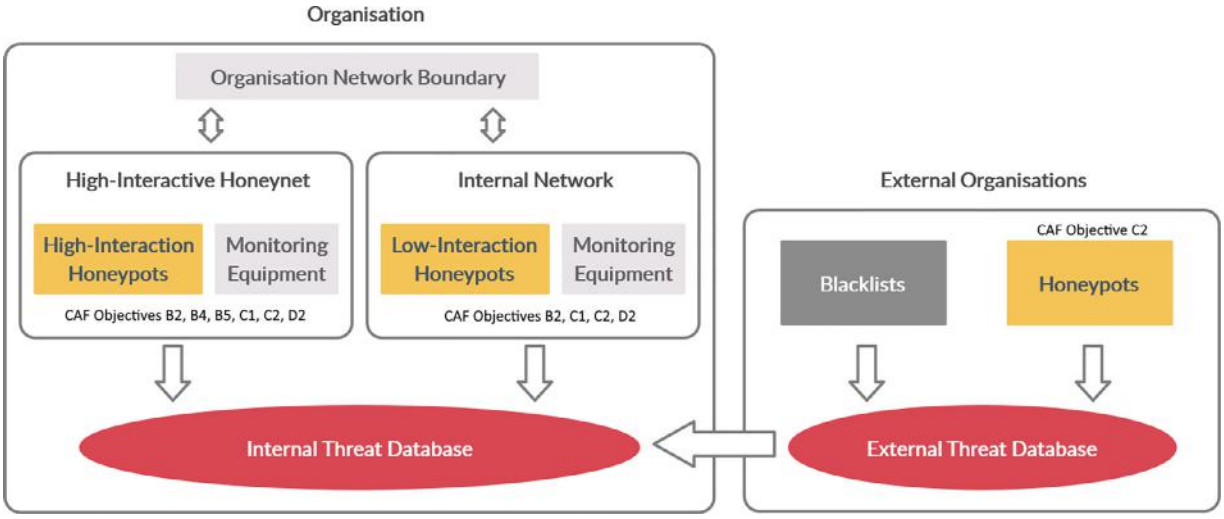

Fig. 4. Overview of HoneyPlant.

stand-alone network would be a high-interaction honeypot within a low-interactive honeynet. Although we discuss the interactivity of honeypots, depending on the monitoring systems on, around, and within the honeypots, the value and amount of data might differ.

This further classification provides us with more information about the honeypot implementation at first glance. It also allows for a better understanding of how extensive the amount of data captured by the honeypot implementation might be. We feel that the environment in which the honeypot is deployed gives a better understanding of the honeypot itself, and this classification can limit misunderstandings about the extent of the implementation.

6. Proposed honeypot framework: HoneyPlant

As we have identified in the previous sections, industrial control systems can be deployed in highly regulated environments such as energy and utility sectors or in environments that require a high amount of up-time. It is clear that honeypots have the capabilities of improving the security of such environments. However, for those highly regulated environments, honeypots can provide another benefit as they can be leveraged to (partially) comply with several of these regulations and guidance. We have yet to see a clear framework to deploy honeypots and use them to be compliant with legislation.

This section presents HoneyPlant, our proposed high-level framework for the implementation and deployment of honeypots within an ICS context. This framework can be used in line with regulations, such as the Cyber Assessment Framework, and as a part of implementing guidance, such as ISO 27002. The main aim of this framework is to support ICS operators with the deployment of honeypots within their environment. To provide a clear overview how their deployment can assist with regulations and provide context on which type of honeypot fits a certain environment to provide increased security with limited risks. Further, we investigate if the implementation of our framework could have made a difference in the protection against historic ICS attacks. For avoidance of doubt, the main goal of the honeypots lie within their detection capabilities. However, we will investigate if detection of events by honeypots deployed according to the framework could have lead to protection against these attacks. Although this framework can be used as a basis for any honeypot network, we mainly focus on ICS environments and regulation. The security of these environment differs from traditional security in several objectives (Neitzel and Huba, 2014) For ICSs, security should not affect the availability and integrity of the device, unlike IT, where confidentiality is the key focus. We do not want to block any legitimate traffic to the devices. There are differences in physical components, network topology a segmentation between ICS and IT environments that have to be taken into account when deploying an ICS honeypot.

6.1. Overview of HoneyPplant

The proposed framework can be broken down in two parts, the organisation's network and the external organisations' networks. The latter contains the blacklists and honeypots situated all over the Internet whilst the former one is owned by the organisation itself. These external blacklists are leveraged to gather intelligence on threats on the Internet which have yet to reach the organisation. This will be part of proactive threat discovery and will aid in the creation of resilient network and systems, and general system security. This proactive security aspect is also in-line with the DHS recommendations for ICS security (Fabro et al., 2016), as reactive approaches can be expensive and disruptive to the operations. Within the organisation, there are two distinct parts, a separate honeypot network and the default internal organisational network to which all employees connect. By placing the honeypots inside the organisation its network, we aim to make it look indistinguishable from the other parts of the organisation network and encourage attackers that have entered the network to attack the honeypots. This part of the framework focuses on the threats an organisation faces. It can be used in relation to objectives B, C, and D of the CAF and the protect, detect and respond functions of the NIST framework. Aside from these, it can be used as part of implementing guidance such as ISO 27002, ISO 27,019 and NIST Guide to Intrusion Detection. A high-level graphical overview of the framework can be seen in Fig. 4.

Several considerations have to be made within the organisation when looking to deploy a honeynet according to this framework. First, the high-interactive honeynet has to mimic a field site (Cell/Area Zone (level 0 - 2)) like the ones deployed within the organisation as closely as possible. This includes the type and amount of devices that are specific to the levels within this zone, and should also take the deployment of commercial ICS software packages (such as ClearSCADA or WinCC) in considera-

tion (Green, Derbyshire, Knowles, Boorman, Ciholas, Prince, Hutchison, 2020). Some examples of architectures can be found in the ICC working paper on Establishing Zones and Conduits from the Industrial Cyber Security Center and Kaspersky Lab (Kaspersky Lab Industrial Cyber Security Center, 2019). Further, to enhance the honeynet, other characteristics such as latency and external IP address have to be taken into account to create an environment that resembles a field site as closely as possible.

6.1.1. External organisations

The external organisation part of the framework consists of several honeypots deployed and maintained by several external organisations with the purpose of monitoring and capturing malicious traffic on the Internet. These honeypots will capture large Internet-wide scans and attacks. When malicious traffic is detected, threat databases will be updated with newly discovered infected IP addresses and signatures of malware. IT departments can then leverage this data to update their security measures. These honeypots can be located within the same industry as the organisation or can reside in organisations that focus on threat intelligence and honeypots such as BadPackets. Therefore an organisation deploying honeypots according to our framework can also be an external organisation within another organisation their deployment, and vice versa.

The organisation has no control about this part of the framework, but it can still be an important way of gathering threat intelligence. Cybersecurity can and should be a collaborative effort between multiple organisations. If one organisation experiences an attack, other organisations should learn from it and be prepared if the adversaries might target them as well. Due to the nature of ICSs, and the required knowledge, attacks might be re-purposed within other attacks. These external honeypots would fit under CAF objective C2 and the NIST detect function.

6.1.2. High-interactive honeynet

The high-interactive honeynet is located in a separate network, and its purpose is to trap attackers and capturing data, this will provide better information as to who is attacking the organisation. Although these honeypots would be deployed within an organisation they would more align with the goals of research honeypots. This network consists of several high-interaction honeypots to provide the organisation with a comprehensive data set. Looking back at the previously discussed implementations, we can see that a higher level of interaction allows the honeypot to capture more data. It also enables the deployment of all ICS protocols supported by the PLC within this environment. To increase the potential of these honeypots further, there would be pre-programmed activity in the network to simulate not only the devices but also the network activity of an operational network. The aim is to make this environment as highly interactive as possible for which we recommend the deployment of a range of honeypots (both ICS and non-ICS) such as Windows honeypots to encompass different levels from the Purdue model. A benefit of some of these honeypots, such as Windows servers and clients, is that multiple honeypots can be deployed on a virtual machine (Alata et al., 2006), without losing the high-interaction aspect. To further emulate a real network environment, several servers (such as FTP or websites) should be deployed as a high-interaction honeypot. By providing both clients and servers, an attacker has more opportunities to transpose between devices which will provide more valuable data. These high-interactive honeypots could be improved by having them engage with attackers and actively interact with phishing attempts (Li and Schmitz, 2009) as phishing is still one of the most popular ways to gain a foothold in a network. We are calling them bi-directional honeypots, due to the engagement of both parties.

Although high-interaction honeypots pose a more significant risk to the network, this is partially mitigated by the segregation between the honeypots and the operational network. All traffic within the honeypot network has to be monitored by an IDS, and hosts must have a host-based intrusion detection system (HIDS) to observe the host. Examples of these HIDSs are AT&T's AlienVault Unified Security Management (AT&T, 2021) and FireEye's Endpoint Security (FireEye, 2021). Specific intrusion detection or prevention systems for ICS environments include Check Point IPS (Check Point, 2020), FortiGate (Fortinet, 2021), Dragos Industrial Cybersecurity Platform (Dragos, 2021) and Claroty Platform (Claroty, 2021). All data captured has to be monitored by the security department to enable them to take appropriate actions. Generally, this part of the framework is flexible, and various amounts of honeypots can be deployed in this location. We agree that this network will require high-maintenance, but it can also provide valuable threat intelligence to be used to lower overall risk and increase security. In terms of legislation, this honeynet would help with Objectives B2, B4, B5, C1, C2 and D2 of the CAF and NIST functions detect and respond.

6.1.3. Internal network honeypots

The second location within the organisation consists of low-interaction honeypots and is deployed in an effort to capture threats within the network. Because of this, these honeypots are better aligned with the purpose of production honeypots. In a similar attempt to Simões et al. (2015), we believe that a good low- to medium-interaction honeypot in this location would be less detectable due to being surrounded by real systems. Additionally, small changes it the default configuration of readily available low-interaction ICS honeypots can lead to more valuable activity (Maesschalck et al., 2021). Because these honeypots are low-interaction, they do not require many resources. Therefore we advise organisations operating critical national infrastructure to implement PLC honeypots. Further, we propose the use of operational PLCs to provide the low-interaction PLC honeypot with data. This data could be obtained, and stored in a database, by sending requests to the operational PLCs at startup. We suggest sending the top 20 requests typically requested to a PLC and storing the response to later use when asked by a malicious user. This data will make the honeypot resemble a real PLC more accurately and reduce the chances of discovery. To further increase the chances of discovery, we propose a periodic connection of legitimate internal systems (such as engineering laptops) to the low-interaction honeypots, which can lure attackers to those systems. As with the separate honeypot network, both intrusion and host-based intrusion detection systems should be put in place to monitor the network and systems that host the low-interaction honeypots. The security department should closely monitor the data captured by these honeypots, as any form of malware captured at this stage has potentially already infected other systems inside the network.

Another aim of these honeypots is to provide forensic investigators with more information about the attack that occurred inside the organisation as a honeypot is capable of storing binaries and other attack-related information for further forensic analysis. Incident response and recovery is imporant and these honeypots could fit well within, for example, the mid-incident and post-incident phases decribed in the framework proposed by Staves et al. (2020). Once the internal network is infected chances are high multiple systems are infected or will be breached soon. These honeypots can give system administrators the opportunity to detect propagating malware or attacks quicker and decrease the chances systems are infected for several months or years. Mapping these low-interactive honeypots to the CAF, they would fall under Objective B2, C and D2. Within the NIST Framework they would fall under the detect and respond functions.

6.2. Could HoneyPlant have made a difference

We have discussed ICS attacks, ICS legislation, honeypots and given an overview of ICS honeypot implementations. With that information, we have proposed a new framework. Now we explore how HoneyPlant could theoretically be used to improve the security of an organisation. We do this by giving an example of how honeypots are used to capture malware and link the capabilities of HoneyPlant back to the ICS attacks we previously discussed.

6.2.1. Honeypots and malware discovery

As previously stated, honeypots can be used to capture binaries and provide a wealth of threat intelligence. Several honeypots have already been explicitly designed to capture malware. These allow for further investigation of the binaries used and even the detection of previously unseen forms of malware.

The New Zealand Honeynet Project implemented and evaluated Nepenthes (Riden, 2006). They explored how the low-interaction honeypot can be used to alert administrators when there is a network compromise and its effective malware detection rate. The developers of the honeypot claim a detection rate of 73 to 84 per cent for new forms of malware (Baecher et al., 2006). Riden used the Norman Sandbox included with the honeypot environment to perform run-time analysis. The honeypot was listening on 255 IP addresses for five days and collected 74 unique samples of which 48 were identified by the anti-virus software used in the test.

When Dionaea captures malware, it calculated the MD5 hash and uses this as the file name; this way, it does not store the same binary twice (Skrzewski, 2012). Other honeypots can hold the same form of malware multiple times which is a waste of resources. Skrzewski ran Dionaea for over nine months, recorded more than 169 000 attempted connections and captured 537 unique malware samples. Out of 1189 attempted connections, 181 were recorded more than once with the top five being recorded over 150 times.

6.2.2. HoneyPlant and discussed ICS attacks

At the beginning of this paper, Stuxnet, BlackEnergy and Wolf Creek were discussed. In this section, we will theoretically discuss if the implementation of our proposed system could have made a difference within these attacks.

Stuxnet entered the Iranian nuclear facility via a USB drive and then spread through the network. As the malware was not yet circulated over the Internet chances are low, it would have been detected by the honeypots. Therefore it would not have been instantly blocked within the network. When the malware started distributing itself across the network, it will have attempted to infect the internal honeypot, which then will have detected the malicious nature, and enable the security team to block further infections and respond to already infected systems. In the case of Stuxnet, the malware would have been spotted within the network. This would have allowed administrators to react before any damage was caused.

BlackEnergy initially infected systems via Word documents within emails and was active on the Internet before it infected the Ukrainian facilities. Because of this, honeypots would have detected it in the wild; accordingly, this data would have been used to update blacklists and other security tools. This piece of malware could have been discovered by built-in anti-malware software within mail filtering services like Microsoft Exchange Online Protection before entering the network. It would also have been identified by network monitoring systems or an internal honeypot when it spread within the network. NIST has also recognised this possibility in their Special Publication 800-160 (Riley et al., 2017). Further, emails received by the honeypot could be investigated or automatically executed within a sandbox.

Although the Wolf Creek attack was not successful in infecting the nuclear part of the facility, it still presented a significant risk. If more systems were in place to limit infection and increase the detection of malware, the risk would have been significantly lower. Similar to BlackEnergy, the initial infection occurred through email, which could possibly have been prevented through a mail filtering service if it was aware of the malware. The malware would have been caught by honeypots situated on the Internet and spreading over the internal network would have been prevented. In this case, if an infected device were to be connected to the nuclear network, network monitoring systems would then have restricted the infection of the ICSs.

Aside from capturing attacks launched on the honeypots by malicious users, system administrators could pretend to be a non-security conscious user and actively deploy malware within the honeypot environment. This could enable further research into the working of the malware, which can then be used to secure the system against novel malware. This can be done through email, active penetration testing, or USB sticks. These opportunities show that a high-interactive honeynet can be used in several ways and does not solely need to rely on real threats to provide useful intelligence.

6.3. Benefits of HoneyPlant

As seen in the previous subsection, the deployment of honeypots in line with HoneyPlant could have potentially stopped some of the adversarial actions during the Stuxnet, BlackEnergy and Wolf Creek attacks. Additionally, we have established clear links between the deployment of the modules within the framework and where these fit with the UK Cyber Assessment Framework and the NIST Cybersecurity Framework. Although this is a high-level framework, with several more in-depth suggestions within each module, we believe it could be of great benefit to operational technology engineers when deciding to deploy honeypots within their environment. We have previously discussed the importance of regulation and legislation within ICS, and especially critical infrastructure, environments in which honeypots are rarely included. The establishment of links between honeypots as a security mechanism and as a mechanism to adhere to these regulations provides a clear merit for their deployment.

Previous work (Antonioli et al., 2016; Cifranic et al., 2020; Litchfield et al., 2016; Provos, 2003; Simões et al., 2013) has focused on very in-depth ways to deploy honeypots within ICS environments, where the location and their impact on the regulation has not been considered. Within this framework we address the matter of guidance and regulation in relation to the deployment of honeypots, and highlight the importance of considering the deployment environment and then tailoring honeypots accordingly to suit that environment context. By following our general guidance specific to each module of HoneyPlant, and leveraging other resources related to the deployment of ICS honeypots such as the Conpot documentation, that tailor-made deployment can be achieved.

7. Conclusion

This paper has presented a comprehensive survey of literature related to the deployment of ICS honeypots. Our survey began with an overview of industrial control systems and honeypots, afterwards we explored past attacks targeting ICS environments and the standards and guidance that are important within these environments. This all fed into a discussion on how honeypots can aid in adhering to these standards and guidance which gave the background for the survey of ICS honeypot deployments.

We analysed three ICS attacks, Stuxnet, BlackEnergy and Wolf Creek. These gave an overview of the attacks that these systems

are facing. These attacks are only a small subset of ICS attacks, and many more have been and are being conducted. It is clear from these three attacks that the security of these systems is of great importance to our safety. Current ICS security lacks the ability to respond to new forms of malware quickly, and the implementation of new patches is slower than necessary to mitigate vulnerabilities rapidly. Due to the nature of these systems implementing patches requires a vetting process to make sure the system will not be affected, and time slot has to be selected in which the system can be safely shut down and updated.

Due to the heavy regulations within the several sectors in which these devices are used, new forms of ICS security have to stem from legislation and guidance. For this reason, we explored several pieces of legislation and guidance from governments and international bodies. These included, and focused, on the UK Cyber Assessment Framework and the US NIST Framework for Improving Critical Infrastructure Cybersecurity. After introducing these, we explored them further in a honeypot context. This showed that honeypots could actively support several parts of many of these legislations and guidance.

To further explore the capabilities of different low-, medium- and high-interactive honeypots, we conducted an extensive survey of published ICS honeypot implementations. The outcome of this survey was an extensive overview of honeypot capabilities and their effectiveness in capturing threat intelligence. However, this also uncovered a need for a better classification system as the current three levels of interaction are limited in capturing the actual abilities of the honeypot. Further, we could see that many authors call their honeypots high-interactive because they realistically implement one or some protocols. We would argue that the threshold for a honeypot to be high-interactive also lays providing an attacker with and environment similar to a device within an operational environment. This includes fully implemented protocols but also the possibility to interact with other systems such as the operating system. It is clear that a high-interaction honeypot allows for more interaction than its low-interaction counterpart, but this classification does not require a high-interactive environment around the honeypot. Therefore, we introduced an additional set of classification that focuses on the environment within the honeypot is deployed. Deploying one device as a honeypot does not result in an environment similar to an operational system. Therefore, we would like to call such honeypot networks a low-interactive honeypot network or honeynet, ones that provide an attacker with an interactive environment would be called a high-interactive honeynet. A vital aspect of this implementation lays in providing a better understanding of the implementation of the honeypot as a whole, when discussing a high-interaction honeypot within a high-interactive honeynet we immediately understand the completeness of the implementation.

Finally, we proposed a honeypot framework that can be used within an organisation to support the deployment of honeypots. This framework is divided into two sections, the organisational network and external organisations. Both these parts can provide a wealth of data, but if an organisation would only focus on one, we would encourage this to be the organisational network. Honeypots deployed by external organisations lack in capturing threats that actively target the organisation, but can capture general trends and attacks targeting similar organisations. The organisational network honeypots are deployed in two separate environments, within the operational network and honeynet that is separated from the operational network. Within the operational network, low-interaction honeypots should be deployed. The separate honeynet should mainly rely on several high-interaction honeypots but can also include low-interaction honeypots; this network should closely represent a typical field site within the organisation. We concluded that this framework fits well within the legislation and guidance previously discussed, and therefore presents both benefits from a security perspective and to support required adherence to legislation.

8. Future work

This research has shown us that honeypots in an industrial control system environment have the opportunity to provide a wealth of data related to the threats facing the organisation, when deployed correctly. Aside from the practical value honeypots can also aid with legislative requirements.

The survey has also identified areas of research that would benefit from additional exploration from an ICS honeypot prospective. This includes an analysis of data captured by tools within the ICS honeypot environment, in order to improve their forensic value for investigators. This is particularly important for low-interactive honeypots, which have relatively limited capabilities. For this goal, it should be investigated how interactive an ICS honeypot has to be in order to capture an adequate amount of data and if low-interaction honeypots can achieve this. Further work can also look more in-depth at mappings between honeypot systems and specific levels within the Purdue model to provide complete coverage, and therefore improving the realism of the honeypot. This could also result in further improvement to Conpot and other readily available low-interaction honeypots, or the development of a new system. On a general note, research can be done to improve the deceptiveness of honeypots, and how honeypots can be used as a preventive measure aside from their current implementations. This should definitely include the ability for honeypots to interact with phishing attacks, as this is a popular way for attackers to get into the network. As well as how honeypots can be deployed in a more defensive manner, similar to traditional security systems. Particularly interesting within this area, due to the specific nature of ICSs, would be how anomaly detection could be used with data captured from ICS honeypots and how it could feed into the security of the network.

Finally, further research in this area could include examining the applicability of honeypots within the legal environment, to give organisations both the confidence and motivation to deploy them. A thorough analysis of legal requirements could provide further evidence that such an investment benefits an organisation from both a legal and security perspective. In addition, there is a need for research that establishes clear legal guidance for the implementation of honeypots from a legal/ethical perspective. One of the main problems that arise from a honeypot deployment is the usage of those honeypots by the attacker for other goals such as botnets or dissemination of illegal material. Another problem that can arise when using honeypots is so-called entrapment, which could result in a legal procedure.

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.

CRediT authorship contribution statement

Sam Maesschalck: Conceptualization, Methodology, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Vasileios Giotsas:** Conceptualization, Methodology, Investigation, Writing – original draft, Visualization, Writing – review & editing, Supervision. **Benjamin Green:** Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing, Visualization, Supervision. **Nicholas**

Race: Conceptualization, Methodology, Writing – review & editing, Supervision, Funding acquisition.

Acknowledgement

The authors gratefully acknowledge the support of the Next Generation Converged Digital Infrastructure (NG-CDI) Prosperity Partnership funded by UK's EPSRC and British Telecom plc (award number EP/R004935/1).

References

Abe, S., Tanaka, Y., Uchida, Y., Horata, S., 2018. Developing deception network system with traceback honeypot in ICS network. SICE Journal of Control, Measurement, and System Integration 11 (4), 372–379.
Agence nationale de la sécurité des systèmes d'information, 2012. Managing cybersecurity for industrial control systems.
Ahmed, I., Obermeier, S., Sudhakaran, S., Roussev, V., 2017. Programmable logic controller forensics. IEEE Secur. Privacy 15 (6), 18–24.
Alata, E., Nicomette, V., Kaâniche, M., Dacier, M., Herrb, M., 2006. Lessons learned from the deployment of a high-interaction honeypot. Proceedings - Sixth European Dependable Computing Conference, EDCC 2006 22, 39–44.
Ani, U.D., Daniel, N., Oladipo, F., Adewumi, S.E., 2018. Securing industrial control system environments: the missing piece. Journal of Cyber Security Technology 2 (3–4), 131–163.
Zhao, C., Qin, S., 2018. A research for high interactive honepot based on industrial service. 2017 3rd IEEE International Conference on Computer and Communications, ICCC 2017 2018-Janua (June), 2935–2939.
Antonioli, D., Agrawal, A., Tippenhauer, N.O., 2016. Towards high-interaction virtual ICS honeypots-in-a-box. CPS-SPC 2016 - Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and PrivaCy, co-located with CCS 2016 13–22.
AT&T, 2021. AlienVault Unified Security Management. AT&T https://cybersecurity.att.com/resource-center/videos/alienvault-unified-security-management-usm-overview.
Baecher, P., Koetter, M., Holz, T., Dornseif, M., 2006. The nepenthes platform: an efficient approach to collect malware. International Workshop on Recent Advances in Intrusion Detection 165–184.
Barak, I., 2020. Cybereason'S newest honeypot shows how multistage ransomware attacks should have critical infrastructure providers on high alert. Cybereason. https://www.cybereason.com/blog/cybereason-honeypot-multistage-ransomware
Barrett, M.P., 2018. Framework for improving critical infrastructure cybersecurity. National Institute of Standards and Technology.
Berman, D., Butts, J., 2012. Towards characterization of cyber attacks on industrial control systems: emulating field devices using gumstix technology. Proceedings - 2012 5th International Symposium on Resilient Control Systems, ISRCS 2012 63–68.
Bernieri, G., Conti, M., Pascucci, F., 2019. Mimepot: a model-based honeypot for industrial control networks. Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics 2019-Octob, 433–438.
Bhamare, D., Zolanvari, M., Erbad, A., Jain, R., Khan, K., Meskin, N., 2020. Cybersecurity for industrial control systems: asurvey. computers & security 89, 101677.
Bodenheim, R., Butts, J., Dunlap, S., Mullins, B., 2014. Evaluation of the ability of the shodan search engine to identify internet-facing industrial control devices. Int. J. Crit. Infrastruct. Prot. 7 (2), 114–123. doi:10.1016/j.ijcip.2014.03.001.
Bilge, L., Dumitras, T., 2012. Before we knew it: an empirical study of zero-day attacks in the real world. In: Proceedings of the 2012 ACM conference on Computer and communications security. ACM, Raleigh, North Carolina, USA, pp. 833–844.
BMI, 2013. Austrian Cyber Security Strategy. BMI.
Bodenheim, R. C., 2014. Impact of the Shodan Computer Search Engine on Internet-facing Industrial Control System Devices. Air Force Institute of Technology https://apps.dtic.mil/sti/pdfs/ADA601219.pdf.
Bou-Harb, E., Lucia, W., Forti, N., Weerakkody, S., Ghani, N., Sinopoli, B., 2017. Cyber meets control: a novel federated approach for resilient cps leveraging real cyber threat intelligence. IEEE Commun. Mag. 55 (5), 198–204.
Bundesamt für Sicherheit in der Informationstechnik, 2013. ICS Security compendium.
Buza, D.I., Juhász, F., Miru, G., Félegyházi, M., Holczer, T., 2014. CryPLH: Protecting Smart Energy Systems from Targeted Attacks with a PLC Honeypot. In: International Workshop on Smart Grid Security. Springer Cham, pp. 181–192.
Caravelli, J., 2019. Cyber Terrorism and Covert Action. In: Cyber Security: Threats and Responses for Government and Business, pp. 1–22.
Cárdenas, A.A., Amin, S., Sastry, S., 2008. Research challenges for the security of control systems. HotSec 5, 15.
Cavusoglu, H., Cavusoglu, H., Jun, Z., 2008. Security patch management: share the burden or share the damage? Manage Sci 54 (4), 657–670.
Chamotra, S., Bhatia, J.S., Kamal, R., Ramani, A.K., 2011. Deployment of a low interaction honeypot in an organizational private network. Proceedings of 2011 International Conference on Emerging Trends in Networks and Computer Communications, ETNCC2011 130–135.
Cherepanov, A., Lipovsky, R., 2016. Blackenergy-What We Really Know About the Notorious Cyber Attacks (October), 1–8.
Check Point, 2020. Intrusion Prevention System - IPS. Check Point https://www.checkpoint.com/products/intrusion-prevention-system-ips/.
Cifranic, N., Romero-Mariona, J., Souza, B., Hallman, R.A., 2020. Decepti-scada: A framework for actively defending networked critical infrastructures. IoTBDS, pp. 69–77.
Claroty, 2021. Claroty Platform. Claroty https://www.claroty.com/platform.
Collins, S., McCombie, S., 2012. Stuxnet: the emergence of a new cyber weapon and its implications. Journal of Policing, Intelligence and Counter Terrorism 7 (1), 80–91.
Crain, J.A., Bratus, S., 2015. Bolt-On security extensions for industrial control system protocols: A Case study of DNP3 SAv5. IEEE Security & Privacy 13 (3), 74–79.
Cremers, C., Dehnel-Wild, M., Milner, K., 2019. Secure authentication in the grid: a formal analysis of DNP3 SAv5. J. Comput. Secur. 27 (2), 203–232.
Department for Business Energy & Industrial Strategy, 2017. Civil Nuclear Cyber Security Strategy. UK Government.
Department of Homeland Security, 2011. Chemical facility anti-terrorism standards (cfats) personnel surety program. DHS.
Department of Homeland Security, 2016. DHS Works with Critical Infrastructure Owners and Operators to Raise Awareness of Cyber Threats. DHS.
Derbyshire, R., Green, B., Hutchison, D., 2021. "Talking a different language": anticipating adversary attack cost for cyber risk assessment. Computers & Security.
Derbyshire, R., Green, B., Prince, D., Mauthe, A., Hutchison, D., 2018. An analysis of cyber security attack taxonomies. Proceedings - 3rd IEEE European Symposium on Security and Privacy Workshops, EURO S and PW 2018 153–161.
Dey, D., Lahiri, A., Zhang, G., 2015. Optimal policies for security patch management. INFORMS J Comput 27 (3), 462–477.
Didier, P., Macias, F., Harstad, J., Antholine, R., Johnston, S. A., Piyevsky, S., Schillace, M., Wilcox, G., Zaniewski, D., Zuponcic, S., 2011. Converged PlantwideEthernet (CPwE) Designand ImplementationGuide. Rockwell Automation https://literature.rockwellautomation.com/idc/groups/literature/documents/td/enet-td001_-en-p.pdf.
Disso, J.P., Jones, K., Bailey, S., 2013. A plausible solution to scada security honeypot systems. In: 2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications. IEEE, pp. 443–448.
Dodson, M., 2020. Using Global Honeypot Networks to Detect Targeted ICS Attacks. In: 12th International Conference on Cyber Conflict, pp. 275–291.
Dragos, 2021. The Dragos Industrial Cybersecurity Platform. Dragos Inc https://www.dragos.com/platform/.
Dutta, N., Jadav, N., Dutiya, N., Joshi, D., 2020. Using honeypots for ICS threats evaluation. Recent Developments on Industrial Control Systems Resilience. Studies in Systems, Decision and Control 255, 175–196.
Fabro, M., Gorski, E., Spiers, N., Diedrich, J., Kuipers, D., 2016. Recommended practice: improving industrial control system cybersecurity with defense-in-depth strategies. DHS Industrial Control Systems Cyber Emergency Response Team.
Farwell, J.P., Rohozinski, R., 2011. Stuxnet and the future of cyber war. Survival 53 (1), 23–40.
Ferretti, P., Pogliani, M., Zanero, S., 2019. Characterizing background noise in ICS traffic through a set of low interaction honeypots. Proceedings of the ACM Conference on Computer and Communications Security 51–61.
Fortinet, 2021. FortiGate Intrusion Prevention System (IPS). https://www.fortinet.com/products/ips. Fortinet.
FireEye, 2021. Endpoint Security Software and Solutions. FireEye https://www.fireeye.com/solutions/hx-endpoint-security-products.html.
Green, B., Derbyshire, R., Knowles, W., Boorman, J., Ciholas, P., Prince, D., Hutchison, D., 2020. {ICS} testbed tetris: Practical building blocks towards a cyber security resource. 13th {USENIX} Workshop on Cyber Security Experimentation and Test ({CSET} 20).
Green, B., Derbyshire, R., Krotofil, M., Knowles, W., Prince, D., Suri, N., 2021. PCaaD: Towards automated determination and exploitation of industrial systems. Computers & Security.
Green, B., Hutchison, D., Frey, S.A.F., Rashid, A., 2016. Testbed diversity as a fundamental principle for effective ICS security research. Serecin.
Green, B., Krotofil, M., Abbasi, A., 2017. On the significance of process comprehension for conducting targeted ICS attacks. CPS-SPC 2017 - Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy, co-located with CCS 2017 57–68.
Haney, M., Papa, M., 2014. A framework for the design and deployment of a SCADA honeynet. ACM International Conference Proceeding Series 121–124.
Harrop, W., Matteson, A., 2015. Cyber Resilience: A Review of Critical National Infrastructure and Cyber-security Protection Measures Applied in the Uk and Usa. In: Current and Emerging Trends in Cyber Operations. Springer, pp. 149–166.
Holczer, T., Félegyházi, M., Buttyán, L., 2015. The design and implementation of a plc honeypot for detecting cyber attacks against industrial control systems. In: Proceedings of International Conference on Computer Security in a Nuclear World: Expert Discussion and Exchange.
Hong, J., 2012. The state of phishing attacks. Commun ACM 55 (1), 74–81.
Hunter, P., 2006. Regular patching cycles. Computer Fraud and Security 2006 (3), 6–7.
Riden, J., 2006. Using Nepenthes Honeypots to Detect Common Malware. https://www.symantec.com/connect/articles/using-nepenthes-honeypots-detect-common-malware. Symantec.
Hilt, S., Maggi, F., Perine, C., Remorin, L., Rösler, M., Vosseler, R., 2020. Caught in the Act : Running a Realistic Factory Honeypot to Capture Real Threats.

Frye, S., 2013. Infographic: The life cycle of a server. https://www.techrepublic.com/blog/data-center/infographic-the-life-cycle-of-a-server/. Tech Republic.
Jaromin, R. M., 2013. Emulation of Industrial Control Field Device Protocols, 1–185.
INCIBE, 2017. Protocols and network security in ICS infrastructures. INCIBE https://www.incibe-cert.es/en/publications/guides/ics-network-security.
INCIBE, 2019. Industrial honeypot implementation guide. INCIBE https://www.incibe-cert.es/en/blog/industrial-honeypot-implementation-guide.
ISO, IEC, 2019. 2019 BSI Standards Publication Risk management - Risk assessment ISO/IEC, 31010.
Jenney, P.H., 2013. ICS Software Protection. In: Laing, C., Badii, A., Vickers, P. (Eds.), Securing Critical Infrastructures and Critical Control Systems: Approaches for Threat Protection. IGI Global, Hershey, PA, pp. 217–239.
Jicha, A., Patton, M., Chen, H., 2016. SCADA Honeypots: an in-depth analysis of conpot. IEEE International Conference on Intelligence and Security Informatics: Cybersecurity and Big Data, ISI 2016 (May 2013) 196–198.
Jie, P., Li, L., 2011. Industrial control system security. Proceedings - 2011 3rd International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2011 2, 156–158.
Joint Task Force Transformation Initiative, 2013. Security and privacy controls for federal information systems and organizations. NIST Special Publication 800 (53), 8–13.
Kaspersky, E., 2011. The man who found stuxnet–sergey ulasen in the spotlight. Nota Bene.
Kaspersky Lab Industrial Cyber Security Center, 2019. Establishing Zones. Kaspersky Lab.
Kendrick, Marian M., Rucker, Zaki A., 2019. ENERGY-GRID THREAT ANALYSIS USINGHONEYPOTS. Naval Postgraduate School.
Khan, R., Maynard, P., McLaughlin, K., Laverty, D., Sezer, S., 2016. Threat Analysis of BlackEnergy Malware for Synchrophasor based Real-time Control and Monitoring in Smart Grid. *SCADA Cyber Security Research*.
Knapp, E.D., Langill, J.T., 2014. Industrial network security: Securing critical infrastructure networks for smart grid, SCADA, and other industrial control systems. Syngress.
Knowles, W., Prince, D., Hutchison, D., Disso, J.F.P., Jones, K., 2015. A survey of cyber security management in industrial control systems. Int. J. Crit. Infrastruct. Prot. 9, 52–80.
Krawetz, N., 2004. Anti-Honeypot technology. IEEE Security & Privacy 2 (1), 76–79.
Kuman, S., Gros, S., Mikuc, M., 2017. An experiment in using IMUNES and conpot to emulate honeypot control networks. 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2017 - Proceedings 1262–1268.
Kushner, D., 2013. The real story of stuxnet. IEEE Spectr 50 (3), 48–53.
Langner, R., 2011. Stuxnet: dissecting a cyberwarfare weapon. IEEE Secur. Privacy 9 (3), 49–51.
Larkin, R., Lopez, J., Butts, J., 2012. Evaluation of traditional security solutions in the SCADA environment. Proceedings of the 7th International Conference on Information Warfare and Security: ICIW 399–403.
Lasi, H., Fettke, P., Kemper, H.G., Feld, T., Hoffmann, M., 2014. Industry 4.0. Business and Information Systems Engineering 6 (4), 239–242.
Lau, S., Klick, J., Arndt, S., Roth, V., 2016. POSTER: Towards highly interactive honeypots for industrial control systems. Proceedings of the ACM Conference on Computer and Communications Security 24-28-Octo, 1823–1825.
Li, S., Schmitz, R., 2009. A novel anti-phishing framework based on honeypots. IEEE.
Litchfield, S., Formby, D., Rogers, J., Meliopoulos, S., Beyah, R., 2016. Rethinking the honeypot for cyber-physical systems. IEEE Internet Comput 20 (5), 9–17.
Maesschalck, S., Vasileios, G., Race, N., 2021. World wide ICS honeypots: A study into the deployment of conpot honeypots. Seventh Annual Industrial Control System Security (ICSS) Workshop. ACM.
Maglaras, L.A., Kim, K.H., Janicke, H., Ferrag, M.A., Rallis, S., Fragkou, P., Maglaras, A., Cruz, T.J., 2018. Cyber security of critical infrastructures. ICT Express 4 (1), 42–45. doi:10.1016/j.icte.2018.02.001.
Marnerides, A.K., Giotsas, V., Mursch, T., 2019. Identifying infected energy systems in the wild. e-Energy 2019 - Proceedings of the 10th ACM International Conference on Future Energy Systems 263–267.
Mashima, D., Chen, B., Gunathilaka, P., Tjiong, E.L., 2017. Towards a grid-wide, high-fidelity electrical substation honeynet. 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm) (October) 89–95.
Mashima, D., Li, Y., Chen, B., 2019. Who's scanning our smart grid? empirical study on honeypot data. In: 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, pp. 1–6.
McLaughlin, S., Konstantinou, C., Wang, X., Davi, L., Sadeghi, A.R., Maniatakos, M., Karri, R., 2016. The cybersecurity landscape in industrial control systems. Proc. IEEE 104 (5), 1039–1057.
Miller, T., Staves, A., Maesschalck, S., Sturdee, M., Green, B., 2021. Looking back to look forward: lessons learnt from cyber-attacks on industrial control systems. Int. J. Crit. Infrastruct. Prot..
Mirian, A., Ma, Z., Adrian, D., Tischer, M., Chuenchujit, T., Yardley, T., Berthier, R., Mason, J., Durumeric, Z., Halderman, J.A., Bailey, M., 2016. An internet-wide view of ICS devices. 2016 14th Annual Conference on Privacy, Security and Trust, PST 2016 96–103.
Mokube, I., Adams, M., 2007. Honeypots: concepts, approaches, and challenges. Proceedings of the Annual Southeast Conference 2007, 321–326.
Mukkamala, S., Sung, A., Abraham, A., 2005. Cyber Security Challenges: Designing Efficient Intrusion Detection Systems and Antivirus Tools. In: Vemuri, V.R. (Ed.), Enhancing Computer Security with Smart Technology. Auerbach, pp. 125–163.
Nakashima, E., Warrick, J., 2012. Stuxnet was work of U.S. and Israeli experts, officials say. The Washington Post.
Navarro, O., Balbastre, S.A.J., Beyer, S., 2018. Gathering intelligence through realistic industrial control system honeypots. In: International Conference on Critical Information Infrastructures Security. Springer, pp. 143–153.
NCSC, 2018. Risk Management Guidance. NCSC.
NCSC, 2019. Cyber Assessment Framework. NCSC.
Neitzel, L., Huba, B., 2014. Top ten differences between ICS and IT cybersecurity. InTech Magazine (June). https://www.isa.org/standards-and-publications/isa-publications/intech-magazine/2014/may-jun/features/cover-story-top-ten-differences-between-ics-and-it-cybersecurity/
NIST, 2012. NIST Special publication 800-30 revision 1 - Guide for conducting risk assessments. NIST Special Publication (September).
Obregon, L., 2015. Secure architecture for industrial control systems. SANS Institute Information Reading Room.
ODVA, 2020. CIP Safety. https://www.odva.org/technology-standards/distinct-cip-services/cip-safety/.
Office for Nuclear Regulation, 2018. Preparation for and Response to Cyber Security Events.
Perlroth, N., 2017. Hackers Are Targeting Nuclear Facilities, Homeland Security Dept. and F.B.I. Say. The New York Times https://www.nytimes.com/2017/07/06/technology/nuclear-plant-hack-report.html.
Piggin, R., Buffey, I., 2016. Active defence using an operational technology honeypot. In: The 11th International Conference on System Safety. London
Pliatsios, D., Sarigiannidis, P., Liatifis, T., Rompolos, K., Siniosoglou, I., 2019. A novel and interactive industrial control system honeypot for critical smart grid infrastructure. IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD 2019-Septe, 1–6.
Ponomarev, S., Atkison, T., 2016. Detection by telemetry analysis. IEEE Trans Dependable Secure Comput 13 (2), 252–260.
Portuguese National Cybersecurity Centre, 2020. National cybersecurity framework.
Provos, N., 2003. Honeyd: A Virtual honeypot daemon. Proceedings of the 10th DFNCERT Workshop 1–7.
Bundesamt für Sicherheit in der Informationstechnik, 2021. Industrial control system security. https://www.bsi.bund.de/EN/Topics/Industry_CI/ICS/Download/download_node.html.
Paralax, 2019. Awesome Honeypots. https://github.com/paralax/awesome-honeypots. GitHub.
ONR, 2021. Security Assessment Principles (SyAPs). Office of Nuclear Regulation. http://www.onr.org.uk/syaps/.
Public Safety Canada, 2012. Industrial Control System (ICS) Cyber Security: Recommended Best Practices. Public Safety Canada https://www.publicsafety.gc.ca/cnt/rsrcs/cybr-ctr/2012/tr12-002-en.aspx.
Ranger, S. https://www.zdnet.com/article/security-four-zero-day-attacks-spotted-in-attacks-against-honeypot-systems/. ZDNet.
Rashid, A., Gardiner, J., Green, B., Craggs, B., 2019. Everything is awesome! or is it? cyber security risks in critical infrastructure. In: International Conference on Critical Information Infrastructures Security. Springer, pp. 3–17.
Redwood, O., Lawrence, J., Burmester, M., 2015. A Symbolic Honeynet Framework for SCADA System Threat Intelligence. In: ICCIP 2015: Critical Infrastructure Protection IX, Vol. 466. Springer, Cham, pp. 103–118.
Richmond, J., 2011. Evolving battlefields: does stuxnet demonstrate a need for modifications to the law of armed conflict. Fordham Int'l LJ 35, 842.
Riley, M., Dlouhy, J., Gruley, B., 2017. Russians are suspects in nuclear site hackings, sources say. Bloomberg, July.
Ross, R., Pillitteri, V., Graubart, R., Bodeau, D., McQuaid, R., 2019. Developing Cyber Resilient Systems: A Systems Security Engineering Approach. Technical Report. National Institute of Standards and Technology.
Rowe, N.C., 2006. Measuring the effectiveness of honeypot counter-counterdeception. Proceedings of the Annual Hawaii International Conference on System Sciences 6 (C), 1–10.
Rosborough, C., Gordon, C., Waldron, B., 2019. All About Eve: Comparing DNP3 Secure Authentication With Standard Security Technologies for SCADA Communications. 13th Australasian Information Security Conference.
Shodan, 2020. Honeypot or not?https://honeyscore.shodan.io/.
Serbanescu, A.V., Obermeier, S., Yu, D.Y., 2015. A flexible architecture for industrial control system honeypots. SECRYPT 2015 - 12th International Conference on Security and Cryptography, Proceedings; Part of 12th International Joint Conference on e-Business and Telecommunications, ICETE 2015 04, 16–26.
Shiva, S., Roy, S., Dasgupta, D., 2010. Game theory for cyber security. Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence.
Simões, P., Cruz, T., Gomes, J., Monteiro, E., 2013. On the use of honeypots for detecting cyber attacks on industrial control networks. In: Proc. 12th Eur. Conf. Inform. Warfare Secur. ECIW. Vol. 2013
Simões, P., Cruz, T., Proença, J., Monteiro, E., 2015. Specialized Honeypots for Scada Systems. In: Cyber Security: Analytics, Technology and Automation. Springer, pp. 251–269.
Skrzewski, M., 2012. Network Malware Activity - a Viewfrom Honeypot Systems. In: Kwiecień, A., Gaj, P., Stera, P. (Eds.), Computer Networks. Springer, Berlin, pp. 198–206.
Spitzner, L., 2002. Honeypots: Tracking hackers. Addison Wesley, Reading, MA.
Spitzner, L., 2003. The honeynet project: trapping the hackers. IEEE Secur. Privacy 1 (2), 15–23.
Staves, Alexander, Balderstone, Harry, Green, Benjamin, Gouglidis, Antonios, Hutchison, David, 2020. A Framework to Support ICS CyberIncident Response and Re-

covery. The 17th InternationalConference on Information Systems for Crisis Response and Management.
The European Commission, 2016. Directive (EU) 2016/1148 of the european parliament and of the council of 6 july 2016 concerning measures for a high common level of security of network and information systems across the union. OJ L 194 (19.7), 2016.
, 2019. The European Parliament and The Council of The European Union. The Directive on security of network and information systems (NIS Directive). The European Parliament and The Council of The European Union https://ec.europa.eu/digital-single-market/en/network-and-information-security-nis-directive.
The Honeynet Project, 2001. Know your enemy: Revealing the security tools, tactics, and motives of the blackhat community. Addison-Wesley.
ThreatStop, 2016. Black energy. Security Report. https://threatstop.com/sites/default/files/ThreatSTOP_BlackEnergy.pdf.
US-CERT, 2016. ICS Alert (ICS-ALERT-14-281-01E) Ongoing Sophisticated Malware Campaign Compromising ICS (Update E). https://www.us-cert.gov/ics/alerts/ICS-ALERT-14-281-01B.
US Nuclear Regulatory Commission, 2010. Cyber security programs for nuclear facilities. US Nuclear Regulatory Commission, Office of Nuclear Regulatory Research.
Vasilomanolakis, E., Srinivasa, S., Cordero, C.G., Mühlhäuser, M., 2016. Multi-stage attack detection and signature generation with ics honeypots. In: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium. IEEE, pp. 1227–1232.
Vasilomanolakis, E., Srinivasa, S., Muhlhauser, M., 2015. Did you really hack a nuclear power plant? an industrial control mobile honeypot. 2015 IEEE Conference on Communications and NetworkSecurity, CNS 2015 729–730.
Vetterl, A., Clayton, R., 2018. Bitter harvest: systematically fingerprinting low-and medium-interaction honeypots at internet scale. 12th USENIX Workshop on Offensive Technologies (WOOT 18). https://github.com/desaster/kippo
Wade, S., 2011. SCADA Honeynets: the attractiveness of honeypots as critical infrastructure security tools for the detection and analysis of advanced threats. Iowa State University 67. http://lib.dr.iastate.edu/etd/12138/
Wang, Y., Wei, J., Vangury, K., 2014. Bring your own device security issues and challenges. 2014 IEEE 11th Consumer Communications and Networking Conference, CCNC 2014 80–85.
Wang, Z., Zhang, J., Li, G., Liu, Q., Chi, Y., Yang, T., Zhou, W., 2019. Honeynet construction based on intrusion detection. ACM International Conference Proceeding Series.
Wells, L.J., Camelio, J.A., Williams, C.B., White, J., 2014. Cyber-physical security challenges in manufacturing systems. Manufacturing Letters 2 (2), 74–77. doi:10.1016/j.mfglet.2014.01.005.
Wilhoit, K., Hilt, S., 2015. The GasPot Experiment: Unexamined Perils in Using Gas-Tank-Monitoring Systems. Trend Micro. https://documents.trendmicro.com/assets/wp/wp_the_gaspot_experiment.pdf.
Yee, K.-P., 2004. Aligning security and usability. IEEE Se 2 (5), 48–55.
You, J., Lv, S., Hao, Y., Feng, X., Zhou, M., Sun, L., 2019. Characterizing Internet-Scale ICS Automated Attacks through Long-Term Honeypot Data. In: Zhou, J., Luo, X., Shen, Q., Xu, Z. (Eds.), Information and Communications Security, Vol. 11999. Springer, Cham, pp. 71–88.
Zamiri-Gourabi, M.R., Qalaei, A.R., Azad, B.A., 2019. Gas what? i can see your gaspots. studying the fingerprintability of ICS honeypots in the wild. ACM International Conference Proceeding Series 30–37.
Zhang, F., Zhou, S., Qin, Z., Liu, J., 2003. Honeypot: A Supplemented active defense system for network security. Parallel and Distributed Computing, Applications and Technologies, PDCAT Proceedings 231–235.

Sam Maesschalck is a PhD student within the Lancaster University School of Computing and Communications and the Security Lancaster Institute. His research involves both technical (systems and networks) and non-technical (cyberspace and international relations) aspects of cyber security, mainly focused within a critical infrastructure environment.

Vasileios Giotsas received the Ph.D. degree from University College London (UCL). He is currently a Lecturer with Lancaster University, where he leads the Networks Area Research of the Security Institute. His research interests include network measurements and the analysis of the routing systems.

Benjamin Green is an academic fellow in the School of Computing and Communications at Lancaster University, UK. His research in- volves both offensive and defensive elements of Industrial Control System security. He is involved in several related research projects, including Operational Technology Management after Cyber Incident (OT-MCI).

Nicholas Race is Professor of Networked Systems within the School of Computing & Communications at Lancaster University. He is also the Director of the Cyber Security Research Centre, and the Associate Dean for Research in the Faculty of Science Technology at Lancaster. His research focuses on developing future networking services built upon Software Defined Networks and Network Functions Virtualisation. This includes new techniques to enhance the Quality of Experience (QoE) of media streaming and support for the detection and remediation of network anomalies. He has a particular interest in the use of these concepts for monitoring and security, building upon his previous work in developing lightweight intrusion detection mechanisms and security monitoring for Wireless Mesh Networks. He is Principal Investigator of NG-CDI, a £5M EPSRC Prosperity Partnership research programme with BT that aims to develop a future network that is “autonomic”, with the capability to react and reconfigure infrastructure accordingly with minimal human intervention.