World Wide ICS Honeypots: A Study into the Deployment of Conpot Honeypots

Sam Maesschalck
s.maesschalck@lancaster.ac.uk
Lancaster University
United Kingdom

Vasileios Giotsas
v.giotsas@lancaster.ac.uk
Lancaster University
United Kingdom

Nicholas Race
n.race@lancaster.ac.uk
Lancaster University
United Kingdom

ABSTRACT

Honeypots are a well-known concept used for threat intelligence and are becoming more ordinary within ICS environments. A well-known ICS honeypot, Conpot, is popular and has been deployed on a large scale. These deployments are not always correctly configured and have odd characteristics compared to a real industrial control system. This paper explores several common Conpot signatures and deployments found through internet search engines such as Shodan. We identify that the default deployment of Conpot is not enough when deploying a honeypot. Afterwards, we explore the behaviour of a real PLC when conducting the same reconnaissance operations. To verify these red flags, we deploy three honeypots with a different configuration, have them scanned by Shodan and evaluate the traffic they get. Our experiments indicate that Shodan leverages CIP for ICS classification. We conclude that proper deployment of a low-interaction honeypot, such as Conpot, requires time and resources to entirely obfuscate the device and fool the attacker to a limited level. However, small changes to the default configuration does increase the performance of Conpot and results in more returning traffic.

CCS CONCEPTS

• **Security and privacy → Systems security**; **Network security**;
• **Networks → Cyber-physical networks**.

KEYWORDS

Honeypots, Conpot, Industrial Control Systems, ICS, Security, Critical Infrastructure

ACM Reference Format:
Sam Maesschalck, Vasileios Giotsas, and Nicholas Race. 2021. World Wide ICS Honeypots: A Study into the Deployment of Conpot Honeypots. In *Seventh Annual Industrial Control System Security (ICSS) Workshop (ICSS 2021), December 7, 2021, Austin, TX*. ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Honeypots are an interesting security concept; instead of keeping attackers out, you want to invite them in [17]. The recent move towards Internet-connected Industrial Control Systems (ICS) [1] brings honeypots to those specialised networks as well. There have been a large number of attacks on critical infrastructure that have had vast implications and have evolved over time to a point where nation-states are one of the main threats in current days. Unlike traditional IT infrastructure, these devices have a specific function and are deployed in unique environments. These and other ICS specific characteristics affect the required security measures as well. Their highly specialised function requires an attacker to have knowledge about the operation of these devices to exploit them successfully. This can be seen as in recent times ICS and critical environments are being targeted by nation-states [9]. On another side, these devices are not part of the typical organisational infrastructure but are instead deployed by specific sectors and in distinct environments, which a honeypot must successfully replicate [12].

They are commonly classified into two, and sometimes three, types: Low-, (medium-) and high-interaction. The difference between these types lies in the deployment, maintenance and, most important, data capturing capabilities. Generally speaking, the higher the level of interaction, the more data the honeypot can capture[16]. This is because a high-interaction honeypot is, in essence, the same system as would be running in the operational environment but with the addition of monitoring systems such as host and network intrusion detection systems. Therefore, high-interaction variants are less likely to be identified as honeypots [6]. Within the ICS environment, these differences are even more important. As attackers are generally more knowledgeable, they could easily spot anomalies within the honeypot. Automated attacks would also be interrupted as they would not receive the appropriate response [13], which is easily built-in a script. Therefore, a honeypot needs to resemble a real system as closely as possible.

Further, due to their critical functions, any attack can have significant effects. However, from a security perspective, legitimate requests should not be blocked as that can have an equally dangerous effect. Therefore we need to find a balance. Honeypots can aid in this as they can try to redirect malicious requests to those systems instead of the operational infrastructure and provide us with threat intelligence to be used in other security measures.

This paper investigates the deployment of Conpot honeypots and discusses the strengths and weaknesses of these deployments. Further, we use well-known Internet scanners to scan the Internet for ICS infrastructure and honeypots. We utilise this data to provide an overview of the differences between real devices and Conpot honeypots. These weaknesses result in red flags for attackers targeting ICS systems and should divert them from these honeypots. Scanners such as Shodan, and its Honeyscore, should be able to pick up on these discrepancies [13].

The core contributions of this paper are:

- Evaluating the default Conpot in relation to a real PLC

ICSS 2021, December 7, 2021, Austin, TX

- Improving upon the default Conpot configuration
- Providing background into Shodan discovery of ICS devices
- Presenting an overview of misconfigured Conpot deployments on the Internet

The remainder of this paper is structured as followed. Section 2 of this paper explores the concept of ICS honeypots and introduces Conpot, a well-known honeypot, alongside other examples. The next section investigates the default configuration of Conpot and compares this to the interaction a real PLC provides for attackers. In section 4 we aim to improve upon the default Conpot configuration by deploying a default Conpot and two additional Conpots with a changed configuration within our Cyber Threat Lab. We verify these changes by evaluating the traffic to these deployments and their discovery by Shodan. Afterwards, section 5.2 delves into real Conpot honeypots deployed on the Internet, which we found via Shodan, Censys and ZoomEye, and the inconsistencies with real systems. Section 6 concludes the paper and section 7 proposes future research possibilities.

2 BACKGROUND

Honeypots have been used successfully for years within traditional IT environments; however, they are still not common within real ICS environments. The benefits of honeypots have been proven extensively. Sochor and Zuzack [15] have deployed several Dionaea honeypots, which are designed to capture malware and one Kippo honeypot, which emulates SSH. They have received over 10,000 connections to their Kippo deployment originating from India, China, Russia and other countries. The Dionaea deployments received more than 1,000,000 connections, of which more than 200 000 distributed malware. Zhuge et al. [21] used high-interactive honeypots to collect autonomous spreading malware, which collected thousands of binaries. Honeypots can indeed provide a wealth of data, although there are also risks attached to their deployment. Especially high interaction honeypots can be compromised [2] and used as an attack vector within the network. The key to obtaining valuable data lies in the proper configuration and deployment of the honeypot.

Due to the nature of low-interaction honeypots, they tend to be easier to identify. A study analysing the fingerprintability of GasPot concluded there are generalisable fingerprinting schemes [20]. There are several issues that are more linked to the nature of low-interaction honeypots, but others can be easily avoided. An Automatic Tank Gauging devices running on an IP address owned by Digital Ocean is strange. Not changing default configurations leaves traces that knowledgeable hackers will identify. These are two examples of clear misconfigurations that should not happen. Further, discrepancies such as increased delay can also have an effect on the believability of the honeypot [18]. One simple spelling error can lead to the attacker identifying the honeypot [10].

One of the most popular low-interaction ICS honeypots is Conpot [11], which is capable of mimicking a range of devices. Further, Conpot can be reconfigured to fit the purpose of the honeypot better. Through a search for Conpot deployments, we can see that there has been a wide range of studies that deployed Conpot in different environments. There have been other low-interaction honeypots such as SCyPH [5] and Mimepot [3]. We can see that SCyPH is promising but lacks to emulate all functions of ICS devices properly. Mimepot does use SDN to redirect malicious traffic to the honeypot and provides the attackers with responses to their requests; this reduces the chances of the honeypot being discovered. However, the study did not include a thorough evaluation.

3 DEFAULT CONFIGURATION

3.1 Setup

To have a baseline configuration to assess Conpot, we deploy a default Conpot configuration within a lab environment. This is done on an Ubuntu 20.04 machine. Further, we deploy a real Siemens S7-300 PLC within the Lancaster ICS lab [7] in its basic configuration. To verify the behaviour of both devices we utilise Snap7, which is a library for interfacing with Siemens S7 PLC.

3.2 Methodology

To evaluate the obfuscation capabilities of Conpot, we investigate how we can differentiate the default Conpot configuration from the default configuration of a real PLC. We will be looking for common signatures of Conpot and discrepancies in services running on the device. To assess both systems from a reconnaissance phase, we will be running several NMAP scans. The first NMAP scan is an adapted slow comprehensive scan (nmap -sS -sU -T4 -A -v -PE -PP -PS80,443 -PA3389 -PU40125 -PY -g 53 –script "default or (discovery and safe)" IP-ADDR), a normal intense scan (nmap -T4 -A -v IP-ADDR) and a scan targeted on port 102 (nmap -sS -sU -p 102 -T4 -A -v -PE -PP IP-ADDR). Afterwards, we will run a Snap7 script, attempting to establish a connection, upload a data block and retrieve the CPU info.

3.3 Conpot

When looking at the services running on a Conpot deployment, we can see FTP running on port 21, S7 on port 102, Modbus on port 502, TFTP on port 69, BACnet on port 47808 and in the default configuration, a web page running on port 80. The OS detection scan guesses three Linux distributions with 96% probability and five other Linux distributions with either 94% or 95% probability. Port 21 shows a default FTP server running. The webpage (Figure 1) running on port 80 contains many references to Conpot, such as 'Technodrome' and 'Last-Modified: Tue, 19 May 1993 09:00:00 GMT'. S7 is one of the most important services for this Conpot deployment. However, the built-in s7-info script fails execution, which might be due to the emulator not supporting all functions. SNMP, running on port 161, is identified as pysnmp and lists Siemens, SIMATIC, S7-200 as system description. We can already see multiple failures within the default Conpot configuration that have to be resolved to properly obfuscate the honeypot.

Attempting to connect to the Conpot, we can successfully establish a TCP connection and a further COTP connection. Both of these messages get an acknowledgement. However, when we request data from the device, there is no response. Querying the device for CPU info results in a TCP: Connection Timed Out exception, as does sending a request to upload a data block. This leads us to think that there is no real environment connected to the device and would be a red flag to attackers.

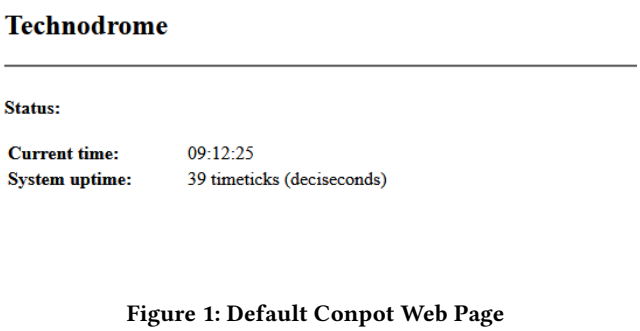

Figure 1: Default Conpot Web Page

3.4 PLC

When interacting with real PLC systems, we see a similar result within the NMAP scans. Unlike with the Conpot honeypots, of which NMAP consistently guesses Linux as one of the possible operating systems, real PLC systems provide a more mixed result. With one of the real PLCs, NMAP guesses Thomson TG712 DSL Router, Microsoft Xbox game console, HP PSC 2400-series printer and Wyse ThinOS 6 as possible operating systems with 90% or higher possibility. From this result, it is clear that NMAP does not have extensive experience with PLC systems and their specialised operating systems.

A Comprehensive NMAP scan further affirms that NMAP cannot determine the exact OS. A more in-depth port-specific scan reveals the iso-tsap service with version Siemens S7 PLC running on port 102. Further specific information gives a module serial number, version, system name and copyright saying "Original Siemens Equipment". Retrieving this data on a known Conpot system resulted in an error. There are no indications that this system might be a honeypot. Further, another real PLC gives us a dashboard (Figure 2) when interacting with it, which is very different from the Conpot page. This allows us to gain more information about the network and device; this is not available on a Conpot system.

If we send the same pattern of messages as we did with the Conpot deployment: a TCP connection followed by a COTP connection and a data request, we get more information. The first two messages get the same response as with the honeypot, but if we request data, then we do get a response from the PLC. We also receive a response when asking the device for information on its CPU. When we try to upload a data block, an exception is thrown stating the function is not authorised for the current protection level. This shows that the PLC is managing another device, or there is at least simulated data on the system. However, based on all the other characteristics, there is no clear indication of a simulated environment.

4 IMPROVING CONPOT CONFIGURATION

From looking at the Conpot configuration files, we can immediately see that the S7Comm implementation is limited. Nine of the eleven requests listed result in a 'request_not_implemented' exception. Within those nine requests, we can find read, write, download and upload. These are basic requests for any PLC; not implementing these results in a severe lack of features.

To improve upon the Conpot configuration it is important to determine the purpose of the honeypot. This includes reducing the protocols that are emulated, such as (T)FTP, and aiming to enable

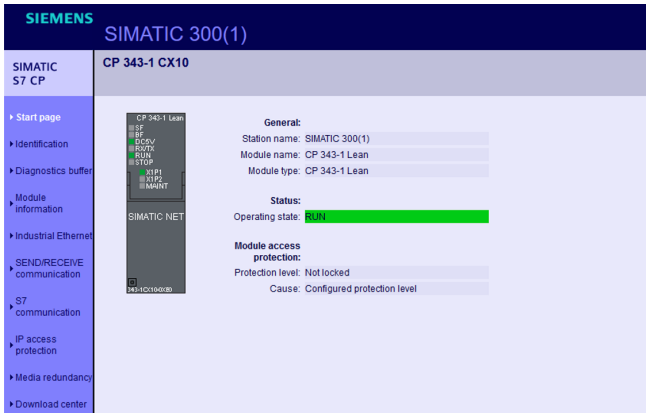

Figure 2: Siemens Simatic 300 Dashboard

the protocols generally seen with a real PLC deployment. Although every protocol can improve upon the data collection of Conpot, there can also flaws in the emulation as they do not respond as expected. Another fault we see on several deployments is the SSH service that is sometimes running on the honeypot, which both NMAP and Shodan identify as the Ubuntu version of SSH as it stems from the operating system. In regards to the HTTP service, it is vital to change the appropriate configuration files before deploying the honeypot. This configuration file includes all of the common signatures we have listed beforehand.

4.1 Experiment

We verify these changes by deploying three Conpot honeypots (Table 1) within our threat lab and aim for them to get scanned by Shodan. The first of these honeypots is the default Conpot template with SSH disabled. For the second deployment, we disable SSH, (T)FTP and change all the common signatures from the HTTP service, which will help us determine the importance of the common signatures for Shodan to identify a Conpot honeypot. This way the honeypot does not bear any signs of signatures such as 'Technodrome', or the default last modified date. The last honeypot has only the S7Comm emulation is activated to replicate a Siemens PLC running only S7Comm.

	Conpot 1	Conpot 2	Conpot 3
(T)FTP	✓		
HTTP	✓	✓	
BACnet	✓	✓	
MODBUS	✓	✓	
CIP	✓	✓	
S7Comm	✓	✓	✓

Table 1: Overview of Conpot Deployments

We ran all three honeypots twice within our threat lab, once over a 54 day period and once over a 47 day period. During the first experiment we received a combined total of 627 connections over all deployments. Of those connections, there were 308 (49.1%) distinct IP addresses. Overall, Conpot 1 received 257 connections

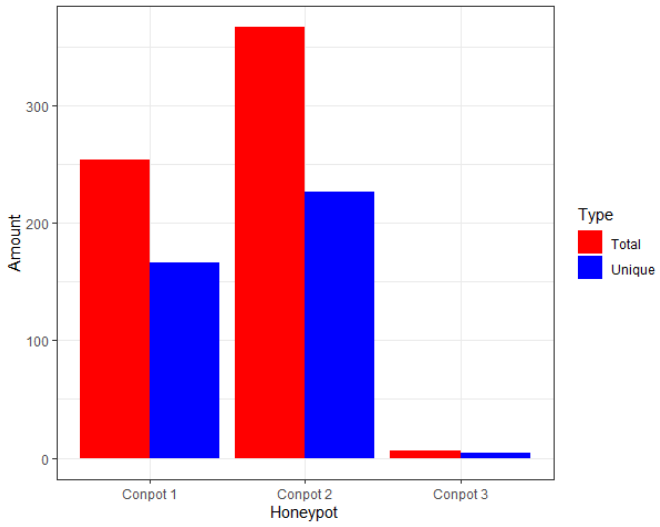

Figure 3: Total Connections for each Honeypot - First Period

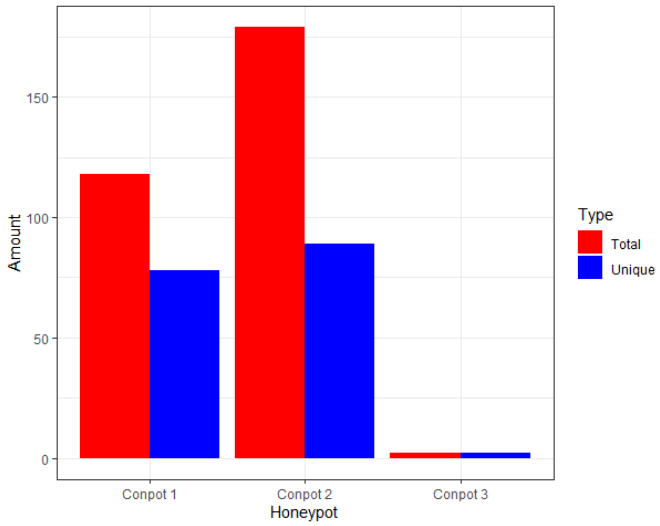

Figure 4: IP Addresses only seen on one Honeypot - First Period

(166 distinct), Conpot 2 received 367 connections (227 distinct), and Conpot 3 received 6 connections (4 distinct) (Figure 3). From this overview, we can see the deployment with only S7 received the least connections by far, and the deployment where we obfuscated the default Conpot deployment received the most. The Conpot 2 deployment received more than one third more connections than Conpot 1, making it the most popular honeypot by a significant margin. Looking at the unique IP addresses that only scan one honeypot in Figure 4 we can see a slight difference between Conpot 1 and Conpot 2. However, Conpot 2 did receive a significant amount of returning IPs.

The second period of data collection saw a total of 1151 connections over all deployed honeypots. Of those there were 560 (48.7%)

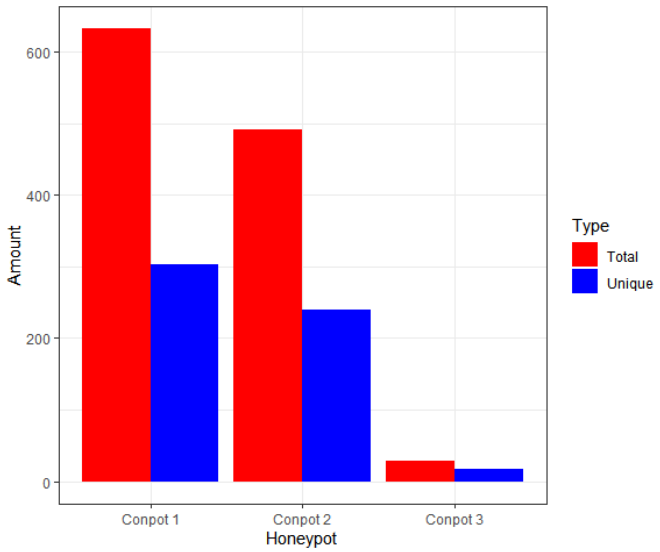

Figure 5: Total Connections for each Honeypot - Second Period

distinct IP addresses. Across all deployments, Conpot 1 received the most connections with 632 total (303 distinct), Conpot 2 received the second most connections with 491 total (240 distinct) and Conpot 3 came in with 28 total connections (17 distinct) (Figure 5). Similar to the first period we ran these honeypots, Conpot 3 received the least connections by far. However, it received significantly more connections compared to the Conpot 3 from the first experiment. Unlike previously, Conpot 2 received less activity than Conpot 1. The main reason for this is the 163 (T)FTP connections seen by Conpot 1. When we only look at the shared protocols between Conpot 1 and Conpot 2, Conpot 1 has 469 connections which puts it slightly behind Conpot 2. Overall, Conpot 1 and Conpot 2 received a similar amount of connections on the shared protocols with only CIP and MODBUS showing a slight difference in favour of Conpot 2. Looking at the IP addresses that are only seen in one honeypot as shown in Figure 6, Conpot 2 (164) received slightly more than Conpot 1 (139) but again has a significantly higher amount of those that are returning (369 vs. 208). Conpot 3 received 1 IP that is not seen by another honeypot, that IP was not seen multiple times.

Throughout both experiment, two honeypots were discovered by Shodan, Conpot 1 and Conpot 2. The time when both honeypots is diffent within both periods. Within the first period the time of discovery differs between both, with Conpot 1 being discovered after 35 days and Conpot 2 being discovered after 45 days. Additionally, Conpot 1 was discovered without any further flags, whereas Conpot 2 was discovered and flagged as an Industrial Control System. Looking at the average daily connections between all three deployments (Figure 7 & Figure 8), we can see Conpot 1 saw a significant increase in activity after discovery, whereas Conpot 2 saw a decline of 35%. The time of and way of Shodan discovery over both deployments differed. Conpot 1 was scanned by Shodan over FTP and HTTP protocols, whereas Conpot 2 saw scans on HTTP

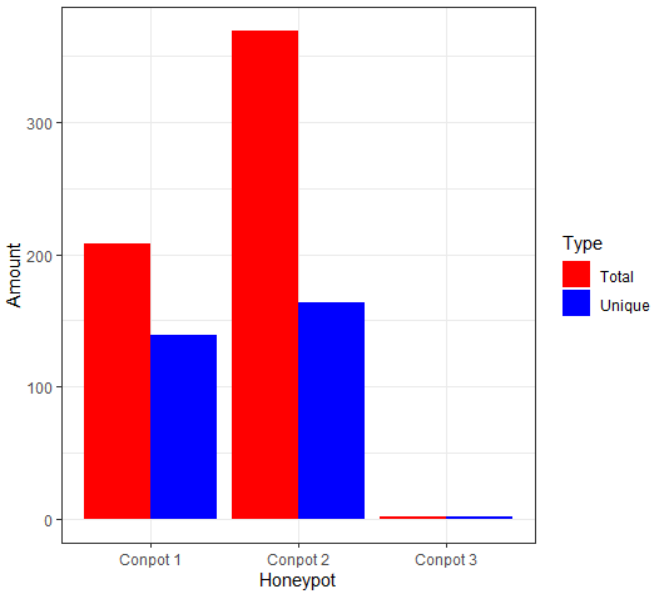

Figure 6: IP Addresses only seen on one Honeypot - Second Period

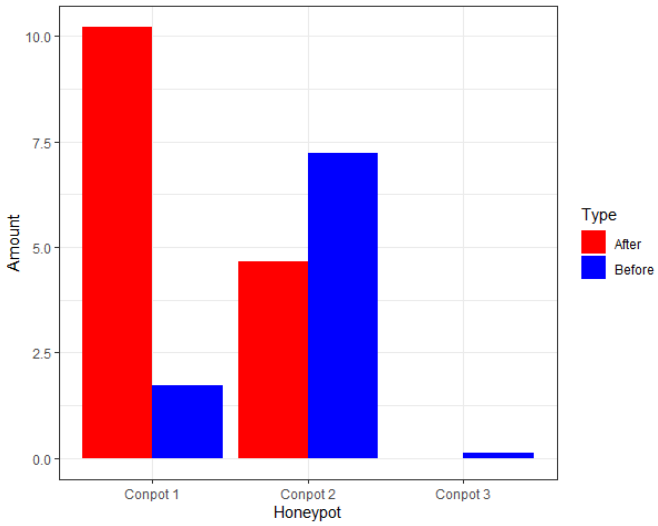

Figure 7: Average Daily Connections before and After Shodan Discovery - First Period

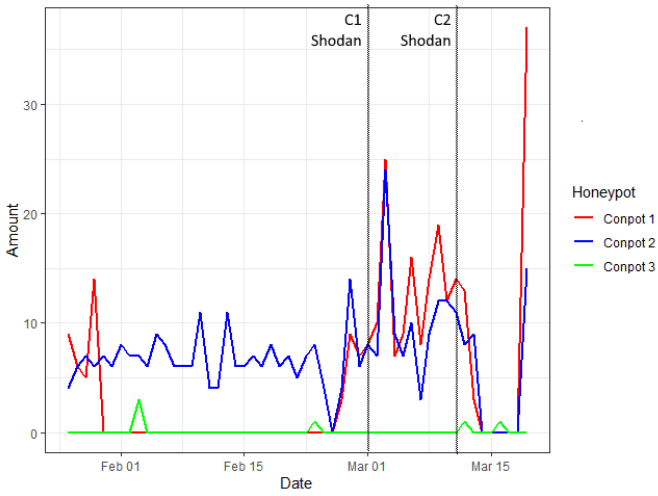

Figure 8: Daily Connections for each Honeypot - First Period

and CIP (Common Industrial Protocol) protocols. Conpot 3 saw no scans from Shodan.

During the second period, both Conpot 1 and Conpot 2 were discovered quickly. With Conpot 1 being scanned on the CIP protocol after 1 day and Conpot 2 being scanned on the CIP protocol after 2 days. This was not the only difference between both experiments, as this time Conpot 1 was also flagged as an ICS. Due to the quick discovery the comparison between the average daily connections before and after discovery is not as interesting. The daily connections itself however is. As we can see in Figure 9, the connections do not drop to zero anymore on Conpot 1 but are more stable. Compared to the first period, Conpot 1 was also scanned over the CIP, BACnet and TFTP. The only protocol that saw no Shodan scans on Conpot 1 and Conpot 2 in this second period was S7comm. Conpot 3 did not receive any Shodan scans.

Overall, we can see a wide variety in connections on each protocol (Figure 10 & Figure 11). During the first period we can see BACnet (33%) and CIP (31%) being the most popular and HTTP (19%) coming in third. Looking a the second period, BACnet (24%) and CIP (36%) are still the most popular, with HTTP (18%) again coming in third. However, Conpot 1 saw the most connections over (T)FTP during the first period, but saw the most activity on CIP during the second. For Conpot 2, CIP is the most popular during the second period with BACnet coming second with a significant difference, whereas the first period saw BACnet being the most popular with CIP being slightly behind. Conpot 3 received the most S7 connections of all three deployments during both experiments. When looking at the IP addresses scanning each protocol, we observed during both experiments that each IP only scans one protocol. This can indicate we mainly got internet scanners that scan one protocol with each address. During the first period, 74 (24%) of the unique IP-addresses that targeted our systems belong to Censys, this seems to support that statement further. Another 26% of unique IPs belonged to cloud provider Digital Ocean, which makes it difficult to assess who is behind these requests. For the second period, 81 (21.2%) of the 382 unique IP-addresses belong to Censys and 78 (20.4%) belong to Digital Ocean.

4.2 Results

Looking back at the activity we have seen on all three Conpot deployments over both periods, there are several conclusions we can make. The main goal of the experiments was to improve upon the default Conpot configuration. Given that Conpot 2 received significantly more traffic than the default configuration in the first period and of the IP-addresses that only targeted one honeypot it had a considerable more returning connections during both periods,

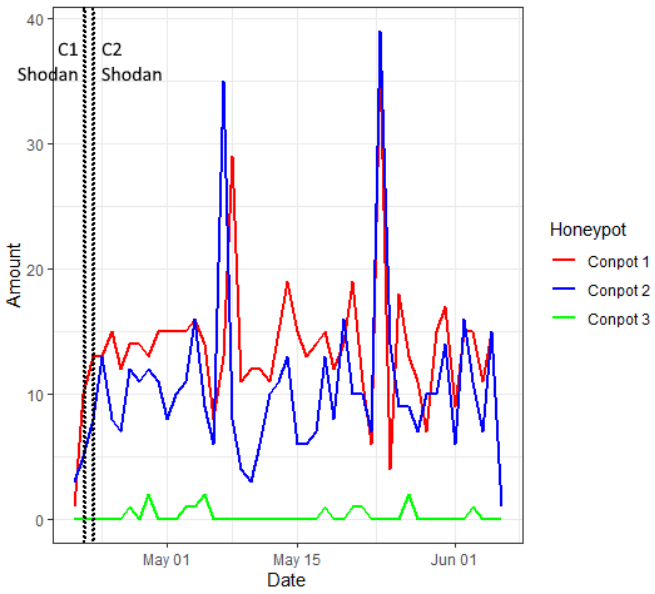

Figure 9: Daily Connections for each Honeypot - Second Period

we have achieved this goal. Disabling (T)FTP and changing the default signatures on HTTP resulted in more activity on protocols linked to ICSs (BACnet, CIP). With regards to S7comm the difference was very small at both experiments. Therefore, we are reluctant to draw a conclusion on S7comm connections. All these factors combined make a strong case that our Conpot 2 deployment is a significant improvement upon the default configuration. Sadly, we saw not a lot of activity on the Conpot 3 deployment in order to draw a conclusion on its effectiveness. However, during both periods it received the most connections on the S7comm protocol, with a relatively big difference during the second experiment. The lack of traffic can be the result of several factors, including not a lot of activity looking for S7comm protocols in general, as we did not see a lot of S7comm connections on other deployments. Further, we acknowledge that the deployment on a university-owned IP-address range can have an impact on attracting real ICS adversaries.

Another goal of this experiment was to have our deployments scanned and indexed by Shodan. As two of our honeypots were indexed twice, we can claim to have achieved this goal. More interestingly, Conpot 2 was discovered and flagged as an ICS both times, whereas Conpot 1 was only flagged as an ICS during the second period. This seems to be because Conpot 1 was not scanned on the CIP during the first experiment. During the second period both Conpot 1 and Conpot 2 were also scanned on BACnet, which they were not during the first period. Neither the first or second period saw Shodan scans on MODBUS. Because the first scan before the honeypots being classified as ICS happened over the CIP protocol, and the first period did not see any BACnet scans, they had to have been identified over CIP. This further believes us to think CIP is used by shodan to classify ICS devices, but that does not conclude it is the only protocol Shodan uses. As none of our systems were classified as honeypots, more can be done to improve upon honeypot detection, especially the limited behaviour of the protocols can be utilised for this. Looking at the connections Shodan made to both honeypots, we can see Conpot 1 received both FTP and HTTP connections during the first period and (T)FTP, HTTP, BACnet and CIP during the second. Conpot 2 received scans on HTTP and CIP during the first period, and CIP, BACnet and HTTP during the second. We set out that discovery by Shodan would result in an increase in traffic to the honeypot. Although this is the case for Conpot 1, which saw a substantial increase in connections, Conpot 2 saw a decline in average traffic per day after discovery during the first experiment. One caveat we have to make with this is that Conpot 1 was discovered by Shodan 10 days before Conpot 2, which could skew the data slightly. Due to the quick Shodan discovery in the second period we cannot infer any trend from that dataset. In research done by Bodenheim [4] activity did not increase after indexing in Shodan. In our data we observe both an increase and slight decrease during the first period. However, given Conpot 1 did receive zero activity over a period of time until indexed by Shodan during the first test but did receive constant activity during the second test we can make a preliminary conclusion it does increase activity.

Overall, we can see that even slight improvements upon the default Conpot configuration can lead to an increase in traffic to the honeypot its ICS related protocols. Getting attackers to return and keep interacting with the honeypot is important to obtain good threat intelligence.

5 CONPOT DEPLOYMENTS ON THE INTERNET

Looking at our research into Conpot deployments we can establish that low-interaction honeypots such as Conpot provide a limited amount of data to investigate and are generally incapable of presenting themselves convincingly as a real system. There is a clear improvement when low-interaction honeypots are deployed in a more realistic or as part of a realistic environment. Further, we can see that high-interaction honeypots gather significantly more useful data [8, 14]. However, across interaction levels, we can see that the deployment of the honeypot plays a vital role in the success of the honeypot.

5.1 Methodology

To better understand Conpot deployments connected to the Internet, we have used popular Internet scanners to find ICS devices. We will investigate several of the systems returned by Shodan, Cencys and ZoomEye. Unlike Cencys and ZoomEye, Shodan does identify systems as honeypots within the results. This is one of the methods to determine if a system is a honeypot easily. Aside from this, we will be investigating common signatures of Conpot and discrepancies from a real PLC device.

5.2 Results

When investigating the PLC devices that can be found through Shodan, we can immediately see some honeypots that are wrongly configured. Hundreds of honeypots are found with one or more

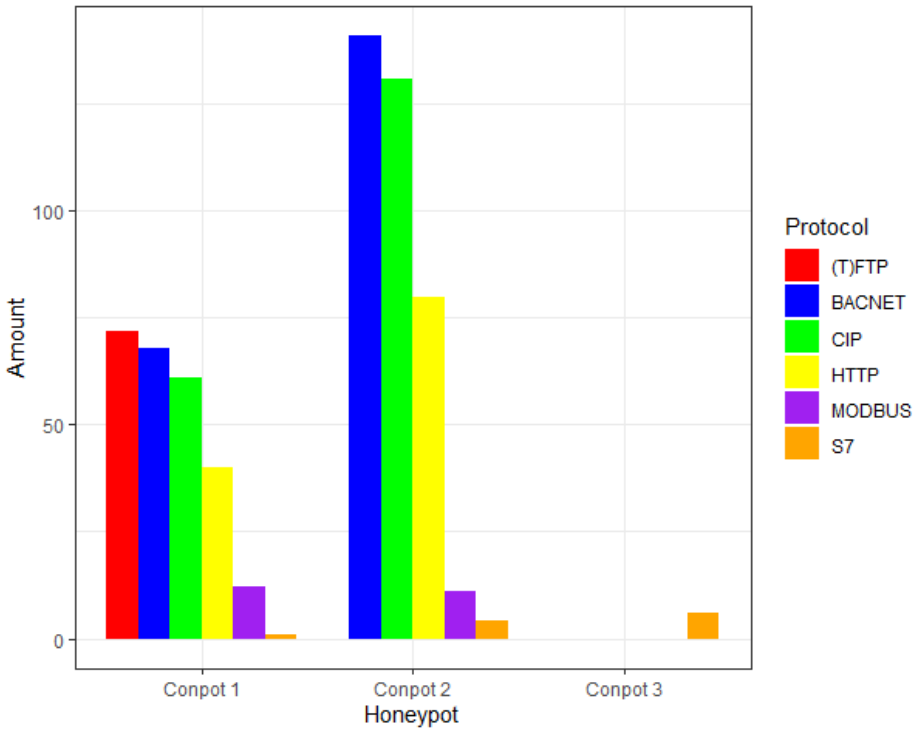

Figure 10: Protocol Connections per Deployment - First Period

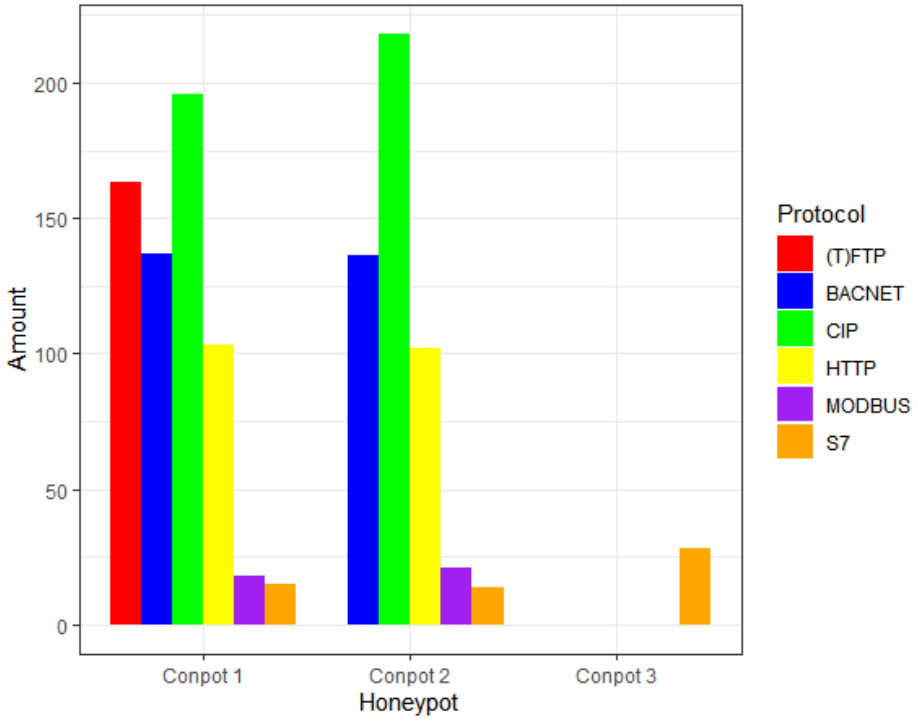

Figure 11: Protocol Connections per Deployment - Second Period

PLC name: Technodrome	319
Plant identification: Mouser Factory	319
Serial number of module: 88111222	336
Last-Modified: Tue, 19 May 1993 09:00:00 GMT (HTTP)	187

Table 2: Common Conpot Signatures on Shodan

Organisation	Amount
University of Maryland	241
NTT	34
Choopa, LLC	10
Amazon.com	6
Digital Ocean	3

Table 3: Top Organisations for "PLC Name: Technodrome" on Shodan

Conpot Signature	Results
Technodrome	385
Mouser Factory	327
88111222 & port:102	348
Tue, 19 May 1993 09:00:00 GMT & port:102	244

Table 4: Common Conpot Signatures on Censys

default Conpot configurations. These include the name 'Technodrome', 'Mouser Factory' as plant information and a serial number of '88111222' (Table 2). Some of the PLCs are also deployed on Digital Ocean, a well-known cloud provider, several IP addresses related to the University of Maryland, Nippon Telegraph and Telephone (NTT), and Amazon AWS IP addresses (Table 3). Aside from NTT, none of these organisations would not generally deploy a PLC on their network; particularly cloud deployment is unseen. For an experienced attacker, any of these signatures should be a red flag. In the case of The University of Maryland, we get 119 results when searching for Modbus devices within one of their network ranges (129.2.27.0/24).

Some organisations do change some parts of the default configuration but still leave obvious red flags on the system. One of these examples can be seen in Figure 12, which is a Conpot honeypot deployed in Poland hosted by OVH ISP. We can see they changed the PLC name on the webpage but failed to change the information linked to the FTP service running on port 21. The FTP server still shows 'Technodrome' and 'Mouser Factory', which is a clear sign of a Conpot deployment. We can also see some deployments from the Australian Academic and Research Network, where the TCP server also still contains clear indications of a Conpot honeypot.

In Table 4, we can see that Censys produces a similar result when searching for these common signatures. However, when using ZoomEye (Table 5), these results are significantly increased. From this, we expect that ZoomEye uses another mechanism than

Conpot Signature	Results
Technodrome	4442
Mouser Factory	2951
Serial Number: 88111222 & port:102	1015
Last-Modified: Tue, 19 May 1993 09:00:00 GMT	3442

Table 5: Common Conpot Signatures on ZoomEye

both Censys and Shodan. As it focuses on the discovery of ICS devices [19], it might find more than others, but further investigation has to be done to find the reason for this discrepancy.

It is clear that these Conpot deployments we have found are not representing real PLC devices. They are wrongly configured and easy to spot. We, therefore, are sceptical about their usefulness. There are quite a few common Conpot signatures, and these are scattered across the honeypot. We can see in Figure 12 that even though there was an attempt to obfuscate the honeypot, there were still common signatures found in other parts of the configuration, such as the FTP server. Going to a non-existing web page also gives an error referring to Linux Apache. The slightest hint to one of these signatures should be a red flag for any attacker. However, these deployments might still capture automated scans and attacks. However, we believe that even these attacks could quickly check the deployment for honeypot signatures or check if Shodan has it categorised as a honeypot. To mitigate these common signatures, the default template should not be used or should be reconfigured to mitigate these shortcomings.

6 CONCLUSION

We can see that there are many inconsistencies between a default Conpot deployment and an actual PLC device. The honeypot we have investigated should have raised many questions to real attackers. For one, there are a lot of common Conpot signatures, which should have been obfuscated before deployment. There are many references to the actual operating system of the device, Linux, which will put off any knowledgeable attacker. Deployments such as these would not provide valuable threat intelligence to the organisation deploying them. If an attacker were to attack a similar deployment, they would notice that no data is returned when requested, although a connection can be set up.

For a Conpot deployment or any low-interaction ICS honeypot, there are many obstacles to overcome. Due to the simulated environment, there is a general lack of interaction. This does extend to the operating system, which is a major red flag for any attacker. ICSs are hosted with specific hardware and are situated within environments that the general population does not generally interact with. Because of this, an attacker that would typically target these environments would be knowledgeable and able to spot these inconsistencies. The rise of search engines as Shodan only increases the importance of obfuscation of the honeypot, as once it is classified as a honeypot, the system would see less valuable interactions. Our own experiments showed that our slightly obfuscated Conpot deployment received more connections on ICS-related protocols and more returning traffic. In contrast, the default Conpot received many connections over (T)FTP, with (T)FTP being the most and

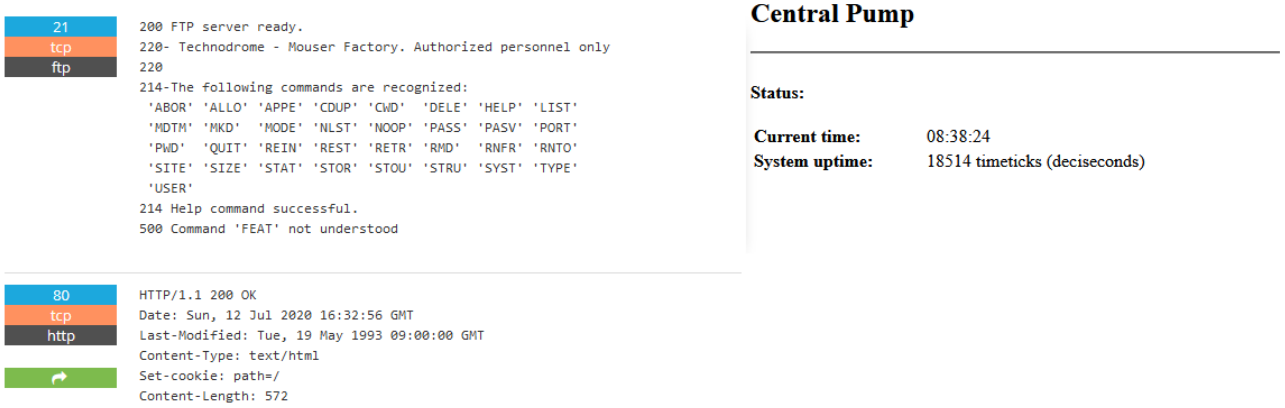

Figure 12: Example of Obfuscation Attempt

second most popular protocol in the first and second experiment, respectively. We also established Shodan leverages CIP as one of the protocols to identify a system as an ICS, but does not perform an in-depth scan that identified any of our discovered systems as a honeypot.

With the inclusion of Shodan within NMAP, it becomes even more important for honeypots to be configured appropriately. It is easy for attackers to detect Conpot within their NMAP scan, a default step in the reconnaissance phase. Further integration with and improvement of Shodan should provide even more information and makes the discovery of many honeypots even easier. We believe that for proper obfuscation of a low-interaction ICS honeypot, a lot of time and resources would have to be spent to simulate an actual device accurately. Nevertheless, fooling a real attacker that is targeting ICSs, remains unlikely through a low-interaction ICS honeypot. We would recommend a high-interactive variant to gain the most valuable data.

7 FUTURE WORK

When we compare results for common Conpot signatures on Shodan, Censys and ZoomEye, we noticed that the latter one returned a lot more results. Further investigation into the reason for this should be done. Although we have identified some inconsistencies between several Conpot deployments and a real PLC, we have not investigated appropriate adjustments to mitigate these. The creation of proper guidelines for the deployment of ICS honeypots would be beneficial for any future honeypot deployments.

ACKNOWLEDGMENTS

This research is funded by EPSRC and BT Prosperity Partnership through the NG-CDI project (Award Number EP/R004935/1), a collaborative partnership with the Universities of Lancaster, Bristol, Cambridge and Surrey and BT.

REFERENCES

[1] Irfan Ahmed, Sebastian Obermeier, Sneha Sudhakaran, and Vassil Roussev. 2017. Programmable Logic Controller Forensics. *IEEE Security and Privacy* 15, 6 (2017), 18–24.

[2] Abdulrazaq Almutairi, David Parish, and Raphael Phan. 2012. Survey of High Interaction Honeypot Tools : Merits and Shortcomings. In *Proceedings of the 13th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting*.

[3] Giuseppe Bernieri, Mauro Conti, and Federica Pascucci. 2019. MimePot: A model-based honeypot for industrial control networks. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics* 2019-Octob (2019), 433–438. https://doi.org/10.1109/SMC.2019.8913891

[4] Roland C Bodenheim. 2014. *Impact of the Shodan computer search engine on internet-facing industrial control system devices.* Technical Report. Air Force Institute of Technology Wright-Patterson Air Force Base.

[5] D. I. Buza, F. Juhász, G. Miru, M. Félegyházi, and T. Holczer. 2014. CryPLH: Protecting Smart Energy Systems from Targeted Attacks with a PLC Honeypot. In *International Workshop on Smart Grid Security*. 181–192. https://doi.org/10.1007/978-3-319-10329-7

[6] Saurabh Chamotra, J. S. Bhatia, Raj Kamal, and A. K. Ramani. 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* (2011), 130–135.

[7] Benjamin Green, Richard Derbyshire, William Knowles, James Boorman, Pierre Ciholas, Daniel Prince, and David Hutchison. 2020. {ICS} Testbed Tetris: Practical Building Blocks Towards a Cyber Security Resource. In *13th {USENIX} Workshop on Cyber Security Experimentation and Test ({CSET} 20)*.

[8] Stephen Hilt, Federico Maggi, Charles Perine, Lord Remorin, Martin Rösler, and Rainer Vosseler. [n.d.]. Caught in the Act : Running a Realistic Factory Honeypot to Capture Real Threats.

[9] Thomas Miller, Alexander Staves, Sam Maesschalck, Miriam Sturdee, and Benjamin Green. 2021. Looking back to look forward: Lessons learnt from cyber-attacks on Industrial Control Systems. *International Journal of Critical Infrastructure Protection* 35 (2021), 100464.

[10] Iyatiti Mokube and Michele Adams. 2007. Honeypots: Concepts, approaches, and challenges. *Proceedings of the Annual Southeast Conference* 2007 (2007), 321–326.

[11] MushMush Foundation. [n.d.]. Installation on host using Virtualenv - Conpot 0.6.0 documentation. https://conpot.readthedocs.io/en/latest/installation/install.html

[12] Neil C. Rowe. 2006. Measuring the effectiveness of honeypot counter-counterdeception. *Proceedings of the Annual Hawaii International Conference on System Sciences* 6, C (2006), 1–10. https://doi.org/10.1109/HICSS.2006.269

[13] Neil C. Rowe, Thuy D. Nguyen, Marian M. Kendrick, Zaki A. Rucker, Dahae Hyun, and Justin C. Brown. 2020. Creating Effective Industrial-Control-System Honeypots. *American Journal of Management* 20, 2 (2020), 112–123.

[14] P. Simões, T Cruz, J. Proença, and E. Monteiro. 2015. Specialized Honeypots for SCADA Systems. In *Intelligent Systems, Control and Automation: Science and Engineering*. 251–269. https://doi.org/10.1007/978-3-319-18302-2

[15] Tomas Sochor and Matej Zuzcak. 2014. Study of internet threats and attack methods using honeypots and honeynets. In *International Conference on Computer Networks*. Springer, 118–127.

[16] Lance Spitzner. 2002. *Honeypots: Tracking Hackers.* Addison Wesley, Reading, MA.

[17] The Honeynet Project. 2001. *Know Your Enemy: Revealing the Security Tools, Tactics, and Motives of the Blackhat Community.* Addison-Wesley.

[18] Michail Tsikerdekis, Sherali Zeadally, Amy Schlesener, and Nicolas Sklavos. 2018. Approaches for preventing honeypot detection and compromise. In *2018 Global*

Information Infrastructure and Networking Symposium (GIIS). IEEE, 1–6.

[19] Wei Xu, Yaodong Tao, and Xin Guan. 2018. The landscape of Industrial Control Systems (ICS) devices on the internet. *2018 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, CyberSA 2018* (2018). https://doi.org/10.1109/CyberSA.2018.8551422

[20] Mohammad Reza Zamiri-Gourabi, Ali Razmjoo Qalaei, and Babak Amin Azad. 2019. Gas what? I can see your gaspots. Studying the fingerprintability of ICS honeypots in the wild. *ACM International Conference Proceeding Series* (2019), 30–37. https://doi.org/10.1145/3372318.3372322

[21] Jianwei Zhuge, Thorsten Holz, Xinhui Han, Chengyu Song, Wei Zou, and Ò Ý Ù Þóùû. 2007. Collecting Autonomous Spreading Malware Using High-Interaction Honeypots. In *International Conference on Information and Communications Security*. 438–451.