Exploitable hosts used in cloud native cyber attacks

Assaf Morag, Aqua Security

Can an in-depth analysis of elements from cyber attack campaigns teach us something new? The answer is yes. As we've seen all too often, crypto-mining campaigns often initiate a vicious circle that starts by infecting and exploiting the host to seek new targets and infect new victims with the same malware.

Of course, there is nothing unique about this, but a recent malware campaign using this tactic did expose something new. An attacker deployed a container image on one of Aqua's honeypots. It contained a text file with a list of around 6,000 host IPs in one of its image layers. An analysis was performed by the cyber research team, Nautilus, comparing it with similar lists from past attacks. The comparison revealed some interesting information that could shed light on the future direction of cyber attacks against cloud native environments.

A vicious circle

First, let's review how these automated attacks are carried out. Although there are some variants in the images used to attack vulnerable hosts, the core behaviour is very similar. Below we portray how one infected host infects another:

1. After the host is compromised, a malicious image is pulled from Docker Hub and then container entry point commands are run. TOR and SSH services are initiated in order to disguise out-going traffic and open a backdoor to the attacker.
2. A shell script is designed to download further scripts and configuration files from the attacker's command and control (C2) server. The configuration files contain lists of Shodan queries and vulnerable IP addresses.[1]
3. A Shodan search is executed. There are several scripts that support this process. All of these files are designed to allow maximum connection metadata randomness (eg, user agents, cookies, using several different Shodan credentials, etc) to avoid being blocked by Shodan.
4. Each new vulnerable host, which was detected by Shodan, was attacked.

One script is responsible for seizing all competing malicious software, while another is designed to deploy and execute a malicious container image.

Vulnerable hosts

On 12 April 2020, a single attack was launched against a honeypot. The image 'stringscene/thttpd:0.04' was designed

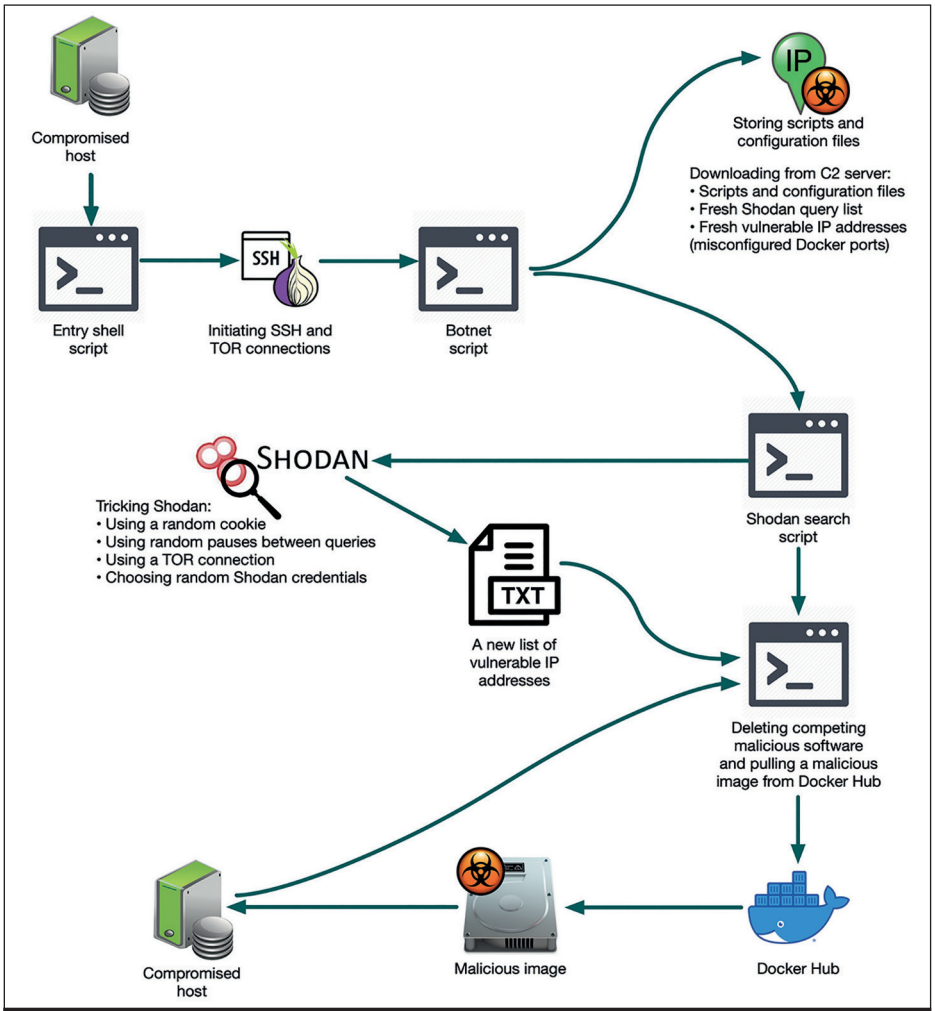


Figure 1: The process by which one host infects another.

to mine crypto-currency. The adversary hid a list of IP addresses within a layer of the container image. Each IP address on the list was set to use port 2375. Traditionally, this port is used as the Docker REST API for unencrypted communication. An examination revealed a list of vulnerable IP addresses, each with a misconfigured Docker API on port 2375.

"Adversaries want to find vulnerable hosts. In order to do so, they need to conduct a mass scan of millions of IP addresses, then determine which ports are open and what services are running on them and find vulnerabilities that can be exploited"

Wanting to learn more from this analysis, we sampled the image from two other past attacks and extracted lists of vulnerable IP addresses. Details are in Table 1.

In total, we analysed the data of 8,558 distinct vulnerable IP addresses. The discrepancy between the sum of IPs that were extracted from these three attacks (8,671) and the number of distinct IP addresses (8,558) suggests that very few IPs appeared in more than one attack – which was indeed the case. Out of 8,558 distinct IP addresses, 97 IPs appeared in two attacks and eight IPs appeared in three attacks. It is unreasonable to assume any organisation would expose such a crucial port for so long (several months), so, it's more reasonable to assume that these IPs are honeypots. Hence, we excluded them from our analysis.

Analysing the Shodan queries

Adversaries want to find vulnerable hosts. In order to do so, they need to conduct a mass scan of millions of IP addresses, then determine which ports are open and what services are running on them and find vulnerabilities that can be exploited.

The adversaries made a smart choice to use Shodan, an online search engine, which stores the metadata of servers. When running a query, the adversary is looking for compromised hosts against a static curated intelligence database. For the end user, Shodan is a passive tool, which means a victim doesn't know that it is being queried. Unlike Shodan, active port scanning tools (eg, Nmap) may leave their imprint on the target's host and tip off the security team when an organisation is being scanned more than usual.

From past attacks, we have collected several configuration files. We retrieved a little over 500 distinct Shodan queries and noticed that the adversaries are:

1. Only looking for vulnerable port 2375. Port 2375 is officially an Internet Assigned Numbers Authority (IANA) used as the Docker REST API for unencrypted traffic. There are several other ports, however, which are also traditionally and officially related to Docker services (for instance 2376, 2377, 4243, 5000, 7946, 9324). Based on the files that we obtained, we haven't seen any references by the adversaries to these ports.

Attack dates	Image	Number of IPs
April 2020	stringscene/thttpd:0.04	5,289
September – October 2019	pocosow/centos:7.6.1810	2,099
June 2019	jzulu/xauto:latest	1,283
Total		8,671

Table 1: List of vulnerable IPs extracted from three attacks on the honeypot.

```
https://www.shodan.io/search?query=port:2375+country:"SG"
https://www.shodan.io/search?query=port:2375+country:"SG"+sh
https://www.shodan.io/search?query=port:2375+country:"JP"+sh
https://www.shodan.io/search?query=port:2375+country:"JP"
https://www.shodan.io/search?query=port:2375+country:"CN"+sh
https://www.shodan.io/search?query=port:2375+country:"CN"
https://www.shodan.io/search?query=port:2375+apache
https://www.shodan.io/search?query=port:2375+xmr
https://www.shodan.io/search?query=port:2375+xmrig+country:"US"
https://www.shodan.io/search?query=port:2375+xmrig+country:"CN"
https://www.shodan.io/search?query=port:2375+xmrig+org:"Aliyun+Computing+Co."
https://www.shodan.io/search?query=port:2375+xmrig
https://www.shodan.io/search?query=port:2375+python
https://www.shodan.io/search?query=port:2375+php
https://www.shodan.io/search?query=port:2375+debian
https://www.shodan.io/search?query=port:2375+nginx
https://www.shodan.io/search?query=port:2375+org:"Hangzhou+Alibaba+Advertising+Co.%2CLtd."
https://www.shodan.io/search?query=port:2375+org:"AWS+Asia+Pacific+%28Seoul%29+Region"
https://www.shodan.io/search?query=port:2375+org:"Amazon+Data+Services+Ireland+Limited"
https://www.shodan.io/search?query=port:2375+org:"Hangzhou+Alibaba+Advertising+Co.%2CLtd."
https://www.shodan.io/search?query=port:2375+version:"1.13.1"
https://www.shodan.io/search?query=port:2375+java
https://www.shodan.io/search?query=port:2375+sh
https://www.shodan.io/search?query=port:2375+mysql
https://www.shodan.io/search?query=port:2375+sql
https://www.shodan.io/search?query=port:2375+country:"KR"
```

Figure 2: Analysing Shodan queries.

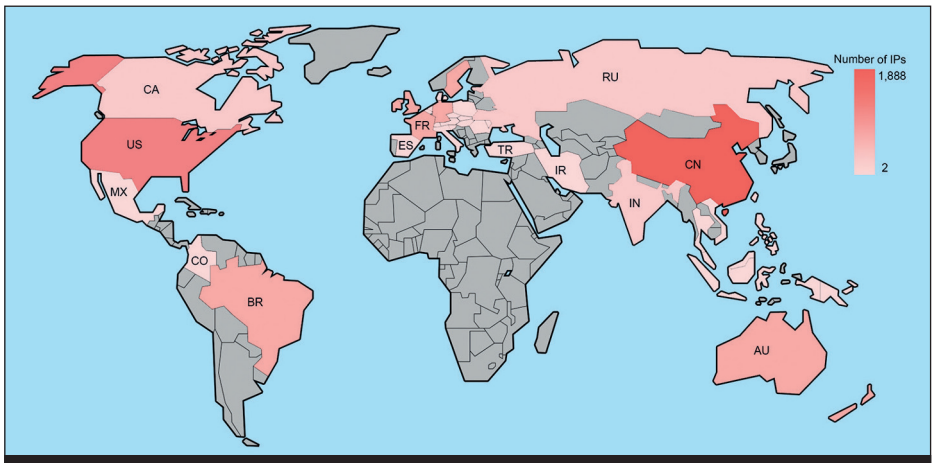


Figure 3: Distribution of vulnerable IPs.

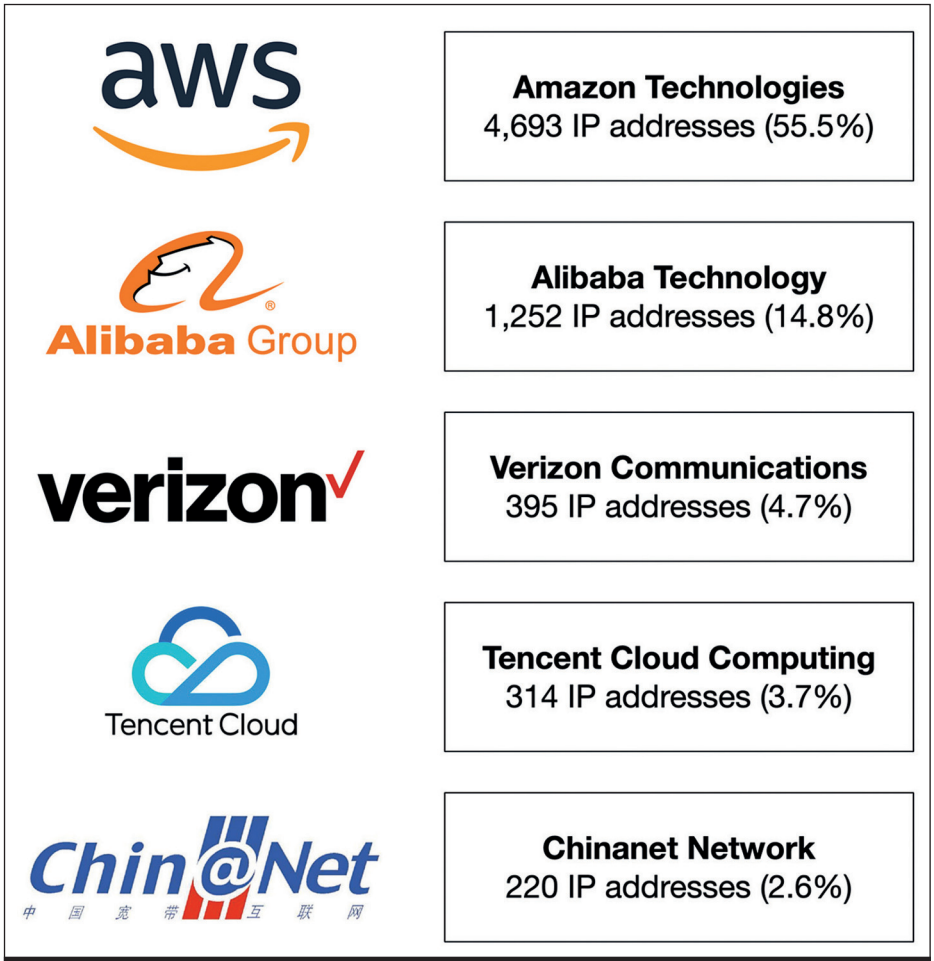


Figure 4: The top five organisations with vulnerable IP addresses.

2. Primarily targeting China, the US, Korea, Singapore, Japan, Brazil, Australia, Russia and India.
3. Using queries to find various services that may run on the Docker port 2375, such as databases, server software, etc.
4. Looking for competing malicious software, such as Kinsing Malware and malicious images (eg, Kannix, avfinder, etc) to block their activity.

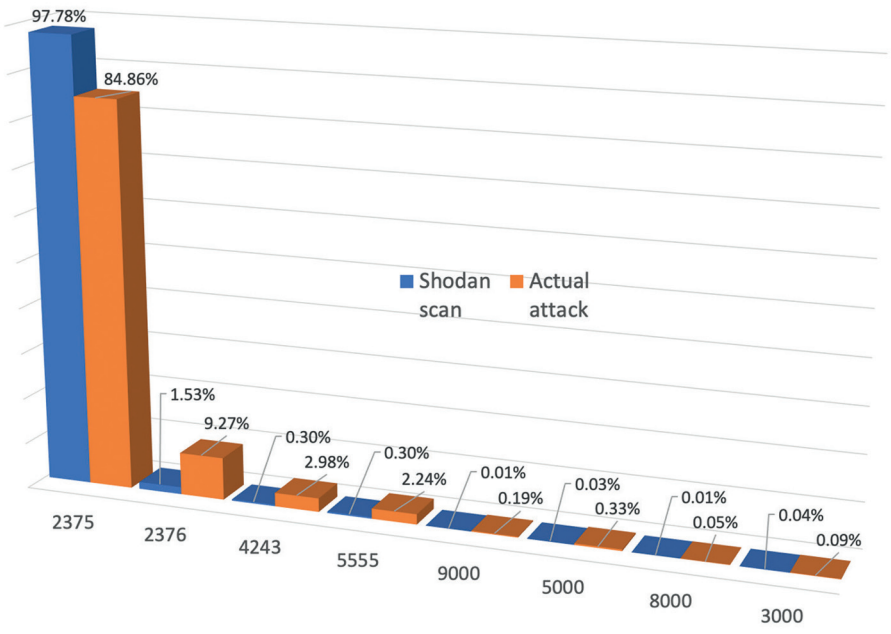


Figure 5: Vulnerable ports found with a Shodan scan and attacks mounted against them.

Analysing vulnerable addresses

Geo-location distribution: Based on the available evidence, China, the US, Japan, Korea and Singapore are the top five most targeted IP addresses, totalling around 60% of the vulnerable IP addresses. This is consistent with our Shodan queries where adversaries are targeting these countries.

Organisation distribution: In Figure 4, you can see the top five organisations with vulnerable IP addresses (based on lists of vulnerable IPs extracted from past attacks). Amazon has the most vulnerable IP addresses. But this is not particularly surprising, since Amazon is ranked as the number one cloud services provider, with an estimated market share of 33%.[2]

Nevertheless, the identity of the rest of the companies in the top five is somewhat surprising. Alibaba, which is ranked fifth, has only 5% market share, but the second most vulnerable IP addresses. Verizon and ChinaNet, which are not even ranked in the top eight cloud services providers, are three and five on the vulnerability ranking, respectively. On the other hand, Microsoft (market share of around 18%) and Google (market share of 8%) have market share estimates putting them in second and third places, but with very few vulnerable IP addresses.

Although some of these findings were a surprise, we should avoid jumping to conclusions. A wrong conclusion might suggest that Amazon and Alibaba may have low security standards, while Microsoft and Google security standards are high. This is not what the data suggests. The reality is much more nuanced, as it could simply be that our data sample is too small or unknowingly biased.

Another problematic aspect is the low dimensionality of details. Many details are missing, such as the identity of the attackers, devices and software targeted, etc. These details could shed more light on these findings and suggest different conclusions.

Vulnerable ports

Unlike what was found in the configuration files, in one of the attacks, a list

Port Number	Role
2375	Docker REST API (plain text) (IANA official)
2376	Docker REST API (ssl) (IANA official).
2377	IANA registered for RPC interface for Docker Swarm.
3000	IANA registered for Cloud9 Integrated Development Environment server. Malware often uses this port as a backdoor
4243	The port is also commonly used by Docker implementations, redistributions and setups (TCP).
5000	Docker Registry server.
5555	Microsoft Dynamics CRM 4.0 (IANA official) There are many reports of malware using this port as a backdoor.
7946	Docker Swarm communication among nodes.
8000	Traditionally used for AWS Local DynamoDB, there are some reports of malware using this port as a backdoor.
9000	ManageEngine AssetExplorer (IT asset management software) uses port 9000 TCP by default. Some online games use this port.
9324	Google Assistant docker containers commonly run a web server listening for HTTP requests on TCP ports 9324 and 5000.

Table 2: A list of ports used in attacks and their official or traditional purposes.

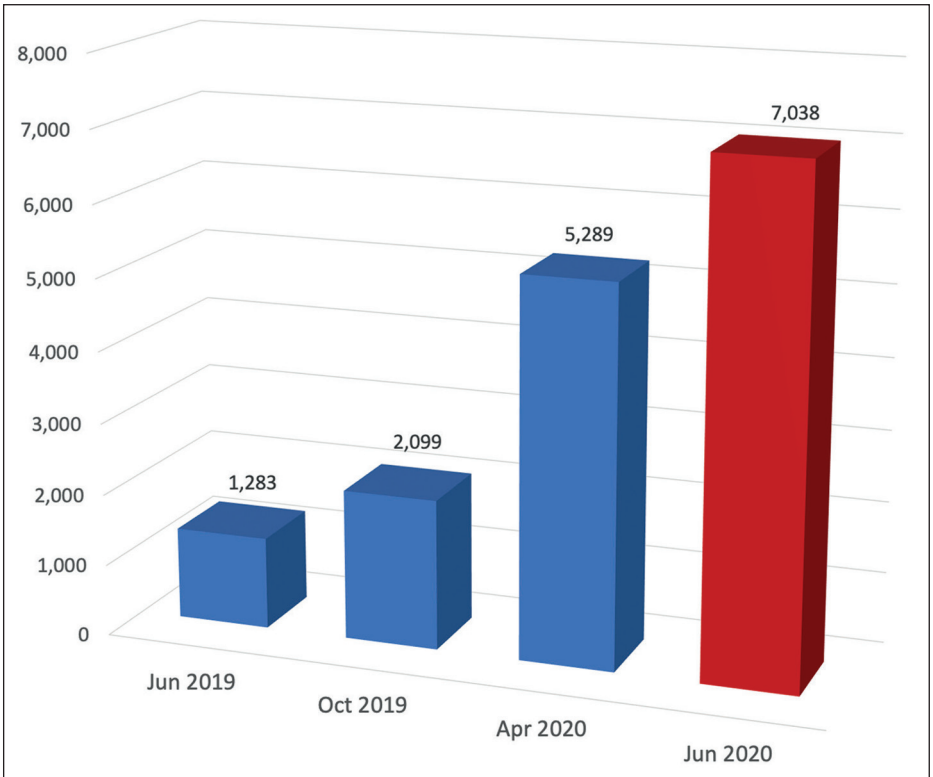


Figure 6: Number of vulnerable IP addresses running Docker services.

of vulnerable IP addresses was detected with various port numbers. This information led to running Shodan queries to detect vulnerable IP addresses with those port numbers. Figure 5 shows a comparison between open ports found with the Shodan scan and actual attacks mounted.

Based on the configuration files retrieved, it appears as though adversaries are mostly targeting port 2375. Nevertheless, there are other vulnerable ports in the wild and adversaries could easily target them – if they haven't done so already. Table 2 shows a list of ports and their official and traditional purposes.[3] Adversaries can also expand their operations to look for more ports that run Docker (or Kubernetes) services.

DevSecOps best practice

Below are some recommendations for DevSecOps. You could implement these as part of your ongoing efforts to mitigate the risks from hidden threats lurking in the cloud:

- Ensure that you are using security and compliance best practices for your public cloud IaaS to mitigate configuration issues across AWS, Azure, Google Cloud, etc. Consider using solutions such as a cloud security posture management tool.
- Scan every image that you use – even from trusted sources. Make sure you are familiar with their use and capabilities. Use a vulnerability scanner such as Trivy (open source).[4]
- Adhere to least privileges access guidelines and avoid root user and privileged modes.
- Dynamically scan images using a dynamic threat analysis tool to uncover hidden suspicious/malicious processes and network communication under simulated runtime conditions using a secure sandbox.

Change over time

As mentioned above, Shodan queries were executed to detect further vulnerable IP addresses running Docker services. Figure 6 shows the results, including the figures that were extracted from the configuration files.

As we suggested above, you shouldn't read too much into any single data point. Nonetheless, in this case, we feel more confident about what the data suggests. It seems like the number of vulnerable hosts running Docker services is increasing over time. This increase appears to be consistent with the following points:

- Using Docker is becoming more robust and easier over time, therefore more people are using these services.
- The variety of people who are using Docker is increasing. This means the

skill level of users is highly variable, which may cause more mistakes and more misconfigured Docker APIs.
- Adversaries are becoming more sophisticated. They are using automated tools to scan and acquire new targets and using more advanced queries to detect vulnerable hosts.

Out of 8,558 IP addresses that were examined, only 105 appeared in more than one list (around 1.2%). This strongly supports our hypothesis that the use of vulnerable IP addresses is increasing.

Summary

This review consisted of three lists of vulnerable IP addresses that were taken from past cyber attacks against Aqua's honeypot. The review included a re-evaluation of the mechanism used to automatically infect the host with crypto-miners and then seek out new vulnerable hosts and infect them as well. Also, there was a review of the analysis regarding the IPs themselves. From this we can draw a number of conclusions:

- The number of vulnerable IP addresses with misconfigured Docker API ports is increasing. This increase is most likely attributable to the increase in Docker usage and adversaries expanding their attack vectors.
- Amazon is the most targeted cloud services provider, and, not surprisingly, has the most vulnerable IPs. Because of its large market share, Amazon may have more end users who are less proficient with cloud native security best practices. This condition often results in environments that are less protected.
- Adversaries are constantly ramping up their game. For instance, they use online search engines to find vulnerable hosts and have automated the infection process.

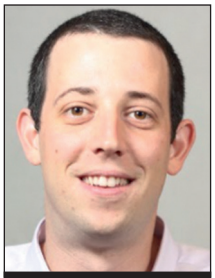
About the author

Assaf Morag is a lead data analyst at Aqua Security. As part of Aqua's research group – Team Nautilus – his work focuses on supporting the diverse data needs of the team.

References

1. Shodan, home page. Accessed Oct 2020. www.shodan.io.
2. Richter, Felix. 'Amazon leads $100 billion cloud market'. Statista, 18 Aug 2020. Accessed Oct 2020. www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/
3. 'Ports database' SpeedGuide. Accessed Oct 2020. www.speedguide.net/ports.php.
4. Trivy, GitHub page. Accessed Oct 2020. https://github.com/aquasecurity/trivy.

How threat actors abuse ICS-specific file types


Nadav Erez

Nadav Erez, Claroty

Project files are integral to industrial control system (ICS) solutions, providing all the necessary data and instructions each machine on the operational technology (OT) network needs to operate. While engineers will use them to ensure the smooth running of operations, security teams can use them to gather an accurate picture of what machines are running on the system along with other critical data, such as where they are and what they are supposed to be doing.

However, extracting information from ICS engineering project files is not always straightforward. While some ICS software vendors offer simple import-export functionality supporting standardised file types such as CSV, others use binary, proprietary formats that can only be interpreted using vendor-specific software.

A lack of full visibility into what is running on the network and how it normally functions presents a significant security risk, because threat actors could infiltrate the network and the security team would be none the wiser. Further, due to their inherent vulnerabilities, ICS project files present an opportunity for threat actors to change how machines operate to cause significant damage, which can be achieved by luring engineers into phishing scams.

ICS project files

An ICS project file is made up of several different files containing a whole range of data that is necessary to carry out the saved project.

What information should we expect to see in these project files? At the top level it would be the network layout, which holds information about what assets are on the network. This might be a PROFIBUS, a standardised, open, digital communications system used in manufacturing automation, along with any stations connected to it.

Additionally, the project file needs to contain details about each individual asset on the network. This will include the devices' IP addresses and serial numbers, as well as data about the slots that each device has and what they are being used for, including module details and order numbers.

The logic necessary for these devices is also saved on the project file, which includes function block or ladder dia-