SimaticScan: Towards A Specialised Vulnerability Scanner for Industrial Control Systems

Rob Antrobus
Security Lancaster Research Centre
Lancaster University
Lancaster LA1 4WA
UK
security-centre.lancs.ac.uk
r.antrobus1@lancaster.ac.uk

Sylvain Frey
Security Lancaster Research Centre
Lancaster University
Lancaster LA1 4WA
UK
security-centre.lancs.ac.uk
s.frey@lancaster.ac.uk

Benjamin Green
Security Lancaster Research Centre
Lancaster University
Lancaster LA1 4WA
UK
security-centre.lancs.ac.uk
b.green2@lancaster.ac.uk

Awais Rashid
Security Lancaster Research Centre
Lancaster University
Lancaster LA1 4WA
UK
security-centre.lancs.ac.uk
a.rashid@lancaster.ac.uk

Over the years, modern Industrial Control Systems (ICS) have become widely computerised and connected via the Internet and are, therefore, potentially vulnerable to cyber attacks. Currently there is a lack of vulnerability scanners specialised to ICS settings. Systems such as PLCScan and ModScan output pertinent information from a Programmable Logic Controller (PLC). However, they do not offer any information as to how vulnerable a PLC is to an attack. In this paper, we address these limitations and propose SimaticScan, a vulnerability scanner specialised to Siemens SIMATIC PLCs. Through experimentation in a comprehensive water treatment testbed, we demonstrate SimaticScan's effectiveness in determining a range of vulnerabilities across three distinct PLCs, including a previously unknown vulnerability in one of the PLCs. Our experiments also show that SimaticScan outperforms the widely used Nessus vulnerability scanner (with relevant ICS-specific plugins deployed).

industrial control systems, vulnerability scanner, programmable logic controllers

1. INTRODUCTION

Industrial Control Systems (ICS) were once logically and physical isolated from other business functions. Greater levels of interconnectivity have been achieved via the relatively recent adoption of Information Technology (IT) within what are often referred to as Operational Technology (OT) devices (devices specifically designed to provide monitoring, control, or automation functionalities within a physical process). Hybrid IT/OT interconnectivity provides numerous benefits related to safety, performance, regulatory compliance, cost saving, etc. While such benefits contribute to organisational objectives, additional risks are also introduced; risks not considered in earlier deployments of ICS due to their isolated, closed network design.

Programmable Logic Controllers (PLCs) and other connected devices within ICSs are integral to the core design and functionality of many industrial processes. However, the deployment of such devices does not often take into account the means for correct implementation of cybersecurity, as historically ICSs were not designed with security as a core consideration. With cyber attacks on the rise against these systems – the most recent being the targeted power blackout in Ukraine (TrendMicro 2016) – there is an urgent need for tools to establish a better understanding of the vulnerabilities in these systems, across both IT and OT domains.

Existing work, e.g., Project SHINE (Infracritical 2014), has demonstrated that pertinent information can be harvested from many PLCs – in SHINE's case, by using the SHODAN (Shodan 2016) search engine. While such analysis demonstrates that many of these devices are publicly searchable, it does not offer practical insights into how vulnerable the specific devices may be to a cyber attack.

In this paper, we present a specialised vulnerability scanner, SimaticScan, specific to Siemens SIMATIC PLCs. SimaticScan can be used to determine if any PLCs found using the SHODAN search engine are vulnerable to specific attacks by utilising the information retrieved from relevant vulnerability databases. SimaticScan thus addresses the limitations of existing scanners by retrieving fingerprint information of a PLC and comparing it to a database of PLCs that are already known to be vulnerable.

Furthermore, SimaticScan goes beyond simply identifying “potential” vulnerabilities to verifying the existence of these vulnerabilities in the PLC under consideration. Through experimentation in a comprehensive water treatment testbed (Green et al. 2016), we demonstrate SimaticScan’s effectiveness in determining a range of vulnerabilities across three distinct PLCs, including a previously unknown vulnerability in one of the PLCs[1]. Our experiments also show that SimaticScan outperforms the widely used Nessus vulnerability scanner (with relevant ICS-specific plugins deployed).

The rest of this paper is structured as follows. Section 2 discusses existing relevant work in this area. Section 3 presents an overview of SimaticScan and its various stages. Section 4 provides a brief overview of the testbed used for experimentation. Section 5 presents results of the experimentation while Section 6 concludes the paper and identifies directions for future work.

2. RELATED WORK

This section explores existing works which have either sought to extract critical information from PLCs, or have been designed to identify security vulnerabilities in OT devices specifically.

Two lightweight port scanners, PLCScan (Digital Bond 2016) and ModScan (Bristow 2008), attempt to enumerate devices across the MODBUS and S7 protocols. The information retrieved from these tools includes slave ID, device name, firmware version, and CPU number. While the firmware version, for example, can be applied to a custom search of a vulnerability database (Toolswatch 2016) in order to identify known associated vulnerabilities, neither tool provides this level of functionality, requiring manual user input, and knowledge of existing databases and their use.

In the IT domain, a number of vulnerability scanning tools are available. Tools such as Nessus, OpenVAS, and Nexpose are all capable of performing vulnerability assessments, and generating comprehensive reports. However, at the time of writing, only Nessus has a function specifically designed for the assessment of OT systems (described as a Supervisory Control and Data Acquisition SCADA plug-in). Section 5 presents a comparison of SimaticScan against this plugin.

As our tool is designed with Siemens Simatic PLCs as a core focus, the work of Beresford (2011) – discussing exploitation of Siemens SIMATIC S7 PLCs – is of particular note. In particular, three attacks are discussed:

- *Replay Attack* allows an attacker to record the incoming and outgoing traffic of a PLC, and use this as a form of authentication by replaying back a payload of packets with the correct previously recorded authentication packet. This vulnerability is achieved due to the plain-text transmission of data within the S7 protocol.
- *Reversing Memory Protections* which could result in an engineer or operator to be locked out of the PLC.
- *Creation of Crafted Packets* that can read and write data to the PLC, even dump the memory residing in a PLC.

The work of Beresford (2011) offers practical insight into Siemens PLC vulnerabilities. However, in order to assess against these vulnerabilities, an assessor requires the skills to craft their own packets, alongside knowledge of the Metasploit framework. Within SimaticScan, we build on the exploits described here, but raise the levels of abstraction, allowing an assessor to identify such vulnerabilities without the described skills/knowledge.

3. SIMATICSCAN

Figure 1 depicts the logical flow of SimaticScan’s operation. The workflow is linear, with calls to third party libraries, including verification of user input when the scanner requires it. Similar to traditional vulnerability scanners, user input is kept to a minimum, ensuring maximum automated operation. The scanner proceeds through the consecutive stages of the program and outputs information based on a target’s susceptibility to specific vulnerabilities. Some stages can potentially disrupt the target:

[1] We note that this vulnerability has been disclosed to Siemens in line with responsible disclosure practices.

the SNMP scan, DoS and fuzzing stages are therefore made optional and should be enabled only in controlled test environments where failure is acceptable. The following subsections provide high-level descriptions of each of the three phases in Figure 1 and the stages within each phase.

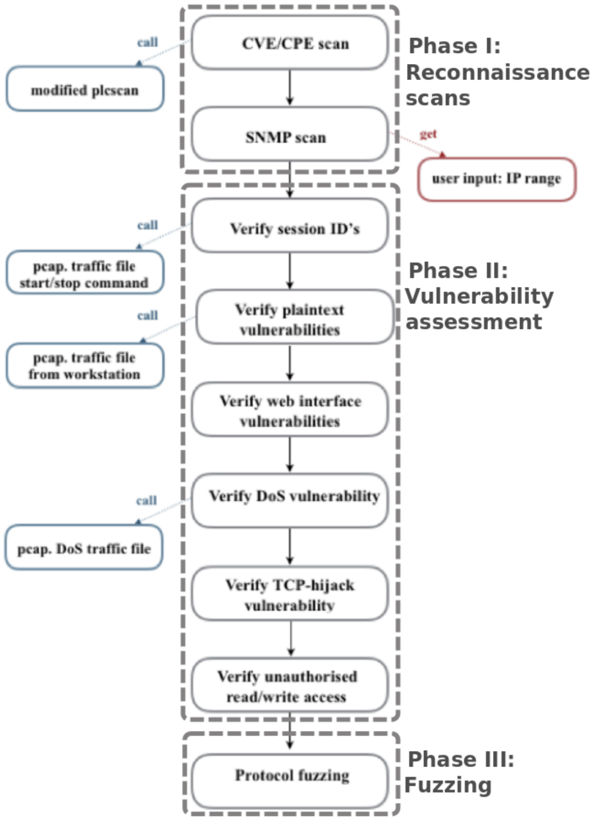

call
modified plcscan
CVE/CPE scan
Phase I: Reconnaissance scans
SNMP scan
get
user input: IP range
call
pcap. traffic file start/stop command
Verify session ID's
Phase II: Vulnerability assessment
Verify plaintext vulnerabilities
call
pcap. traffic file from workstation
Verify web interface vulnerabilities
call
Verify DoS vulnerability
pcap. DoS traffic file
Verify TCP-hijack vulnerability
Verify unauthorised read/write access
Protocol fuzzing
Phase III: Fuzzing

Figure 1: *Overview of SimaticScan*

Phase I: Reconnaissance Scans

CPE/CVE Search. SimaticScan begins by finger-printing the PLC and receiving information about the PLC's internals, such as the module name and type, serial number, and basic hardware/firmware versions. This stage of the scanner relies on a modified version of the aforementioned PLCScan tool. Once retrieved, the firmware version is compared to entries from a list of Common Platform Enumeration (CPE) terms, specified by the vendor, in this case Siemens. If the PLC firmware is less than, or equal to, any of the current CPE terms, it is likely that a vulnerability exists, as Siemens list their CPE entries alongside a CVE entry. *vFeed* (Toolswatch 2016) – a third-party centralised database of vulnerability and mitigation data – is used to search for the CPE and CVE entries associated with a particular PLC's firmware version. Using vFeed's in-built API calls, SimaticScan outputs any CVEs associated with the PLC's CPE entry, allowing the user to obtain more information about a vulnerability, such as severity and risk scores.

SNMP Scan. SimaticScan allows the user to perform a Simple Network Management Protocol (SNMP) scan over a range of IP addresses. The SNMP protocol is used to exchange information between devices in a network, known as agents. An SNMP manager – Scapy (secdev.org 2016) – is used to communicate with all agents on the network, and SimaticScan receives any response from SNMP enabled devices. This information highlights potential pivot points around the network. For example, a scan might receive a response from an engineering workstation, which could present additional attack vectors a threat agent could exploit.

Phase II: Vulnerability Assessment

Verify Session IDs. This stage of SimaticScan is inspired by the work of Beresford (2011), and parses a Wireshark capture for session IDs that may be transmitted in plaintext. Note that a session ID acts as an authentication mechanism between the operational workstation and PLC. Furthermore, it is one that never expires. SimaticScan detects if the session ID is in plaintext, and outputs the vulnerability associated with it, i.e., an attacker could conduct a replay attack against the PLC.

Verify Plaintext Vulnerabilities. After searching for session IDs that may be transmitted in plaintext, SimaticScan looks for additional plaintext vulnerabilities, focusing on the identification of commands sent from a workstation to a PLC. A Wireshark capture is parsed to identify particular commands sent to the PLC in plaintext. If plaintext is found, Simaticscan informs the user and outputs the associated vulnerability.

Verify Web Interface Vulnerabilities. Several PLCs have a built in web interface, allowing for remote monitoring and control. An attacker could seek to exploit this, to gain information about the PLC and its internals, such as time-stamping of commands and operations, and CPU start/stop control functionality. SimaticScan checks if port 80 is open, providing access to a Siemens S7 web interface. As most web interfaces require admin-level privileges to login, SimaticScan searches whether the ones it finds feature hard-coded default credentials – using public records such as StrangeLove (2016) – and warns the user accordingly.

Verify Denial of Service (DoS) Vulnerabilities. DoS is a process whereby an attacker sends a large number of data packets to a target, thus flooding it with too much data, leading to a system crash. SimaticScan can perform this attack by sending a large number of packets to the PLC over the Siemens ISO-TSAP protocol.

Verify TCP Hijack vulnerability. A TCP hijacking attack is used to intercept the communications between two networked hosts. During this particular stage, SimaticScan aims to perform a TCP hijack of a PLC, verifying if a spoof IP address can be introduced. A spoofed IP address can be used by an attacker to perform a man-in-the-middle attack, and also as a basis for redirecting TCP traffic to and from the PLC. Identification of this vulnerability shows that sequence numbers used for authenticating TCP sessions between hosts are not robust enough (not random enough). To this end, SimaticScan attempts to guess the sequence number (via successive increments, while monitoring its own local sequence number) and establish an authenticated TCP session with a spoofed IP address.

Verify unauthorised read/write access. This stage of the scan utilises a third-party Python library – Snap7 (2016) – to determine if values stored in a PLC's memory can be read from, or written to. The focus of this stage is to test if an attacker can read/write values to/from the PLC's datablocks without any authentication, thus affecting a PLC's execution of logic.

Phase III: Fuzzing

Protocol fuzzing is the process of sending random, or semi-random data to a process (Wang et al. (2013), Kitagawa et al. (2010)). In the final stage of the scan, SimaticScan determines if a PLC can be fuzzed, a useful function for determining other vulnerabilities. Fuzzing acts as a vulnerability research tool where the data sent can identify additional vulnerabilities that were not previously discovered. SimaticScan sends random data to the PLC and provides the outputs. Examples of random payloads include:

1. Random strings of ASCII characters.
2. Single-character strings.
3. Blocks of ASCII punctuation characters.
4. HTML-encoded strings.

SimaticScan does not provide feedback regarding the result of the fuzzing phase: for instance, the operator must check whether the target has crashed and possibly restart it manually. This limitation (also affecting the potentially disruptive SNMP scan and DoS phases) is a an opportunity for future improvements of SimaticScan.

4. OVERVIEW OF THE TESTBED

Our ICS testbed (Green et al. 2014, 2016) provides a broad range of ICS devices, physically applied, with no simulation. Virtualisation is used in order to allow for greater flexibility and scaling of workstation and server implementation across the system. Nevertheless, the routing of these systems remains under physical network device responsibility.

This use of physical devices – PLCs in particular – and physical network routing provides a high level of credibility to the experiments conducted and discussed throughout section 5. We note that the network architecture of the testbed has been reviewed by industry experts and deemed valid and representative of real-world large scale distributed systems. It should also be noted that, although it is not the focus of the experiments in this particular paper, the testbed also covers the intersection of IT and OT environments.

5. EXPERIMENTAL RESULTS

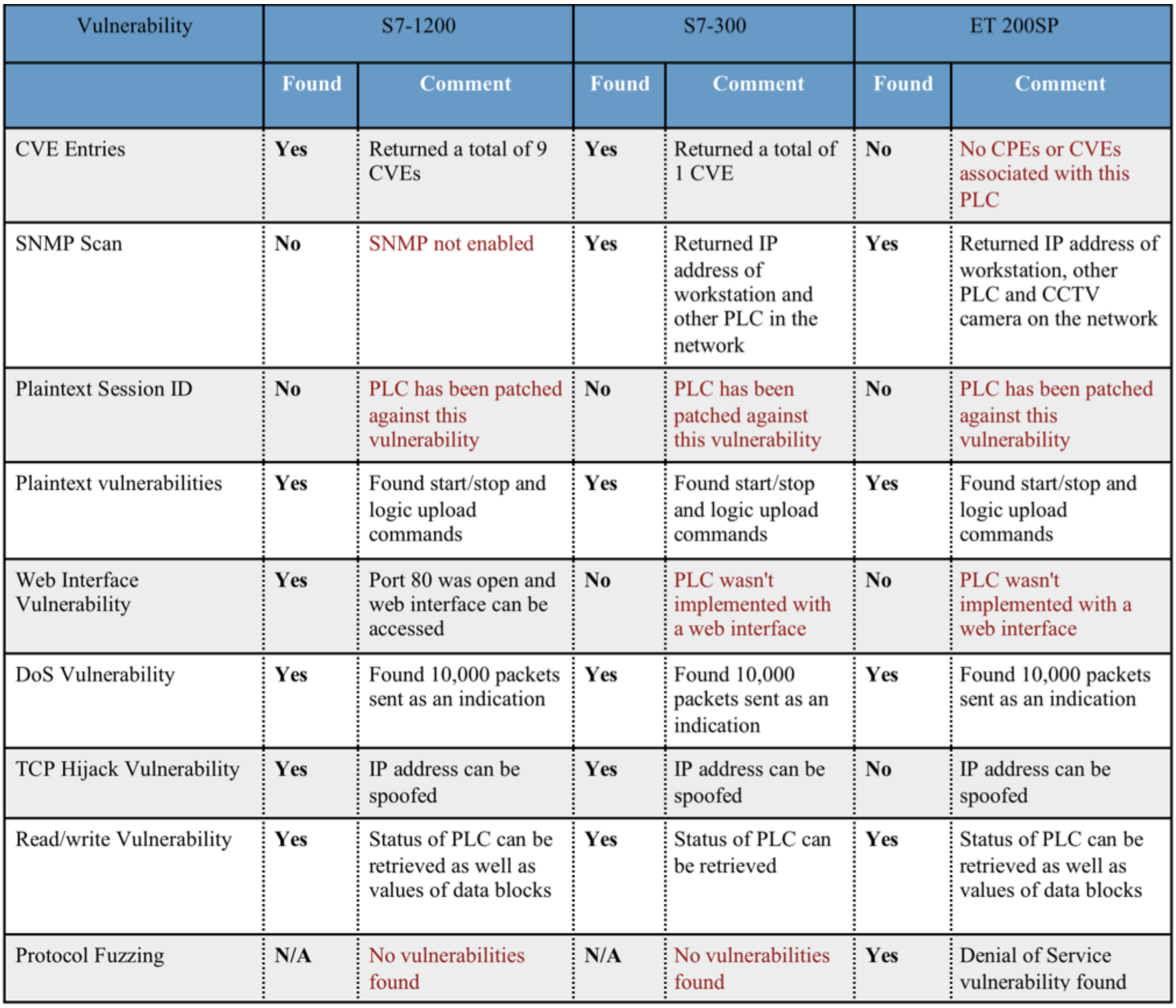
Three Siemens PLCs in the above testbed were used for evaluation: the S7-1200 (Siemens 2016*b*), S7-300 (Siemens 2016*c*), and ET200S (Siemens 2016*a*). The following subsections discuss the vulnerabilities discovered through the use of SimaticScan. In addition to SimaticScan, Nessus and its associated SCADA plug-in were also deployed – cf. Security (2015). This allowed for a comparison of SimaticScan's effectiveness with an existing commercial vulnerability scanner. Finally, SimaticScan was also run against a SCADA honeypot: Conpot (2016).

5.1. SimaticScan deployment over the testbed

The results of scanning the three PLCs, i.e., the S7-1200, the S7-300 and the ET 200SP, with SimaticScan are shown in Figure 2. The comments in the figure summarise any discovered vulnerabilities, or suggestions as to why a vulnerability was not found, e.g., a known patch had been applied.

It can be seen that the *S7-1200* was vulnerable to 8 categories of vulnerabilities. A high number of CVEs were returned, alongside the web interface vulnerability. Protocol fuzzing yielded no additional vulnerabilities. When performing an SNMP scan, the SNMP network was not enabled for the S7-1200. Therefore, any potential pivot points in the network were not returned. The session ID vulnerability has been patched by Siemens in firmware 2.0. As the firmware of this PLC was 3.0.2, no plaintext session IDs could be found.

The *S7-300* was vulnerable to only one CVE. This vulnerability (CVE-2015-2177) is a denial of service vulnerability, and differs from the S7-1200 CVEs, as it affects the entire range of S7-300 PLCs, regardless of the firmware version. Again,



Vulnerability	S7-1200		S7-300		ET 200SP	
	Found	**Comment**	**Found**	**Comment**	**Found**	**Comment**
CVE Entries	**Yes**	Returned a total of 9 CVEs	**Yes**	Returned a total of 1 CVE	**No**	No CPEs or CVEs associated with this PLC
SNMP Scan	**No**	SNMP not enabled	**Yes**	Returned IP address of workstation and other PLC in the network	**Yes**	Returned IP address of workstation, other PLC and CCTV camera on the network
Plaintext Session ID	**No**	PLC has been patched against this vulnerability	**No**	PLC has been patched against this vulnerability	**No**	PLC has been patched against this vulnerability
Plaintext vulnerabilities	**Yes**	Found start/stop and logic upload commands	**Yes**	Found start/stop and logic upload commands	**Yes**	Found start/stop and logic upload commands
Web Interface Vulnerability	**Yes**	Port 80 was open and web interface can be accessed	**No**	PLC wasn't implemented with a web interface	**No**	PLC wasn't implemented with a web interface
DoS Vulnerability	**Yes**	Found 10,000 packets sent as an indication	**Yes**	Found 10,000 packets sent as an indication	**Yes**	Found 10,000 packets sent as an indication
TCP Hijack Vulnerability	**Yes**	IP address can be spoofed	**Yes**	IP address can be spoofed	**No**	IP address can be spoofed
Read/write Vulnerability	**Yes**	Status of PLC can be retrieved as well as values of data blocks	**Yes**	Status of PLC can be retrieved	**Yes**	Status of PLC can be retrieved as well as values of data blocks
Protocol Fuzzing	**N/A**	No vulnerabilities found	**N/A**	No vulnerabilities found	**Yes**	Denial of Service vulnerability found

Figure 2: *Results of scanning the three PLCs with SimaticScan*

the session ID vulnerability was patched, and the web interface was not configured, therefore no web server vulnerabilities were identified. The SNMP scan found potential pivot points including the engineering workstation, and another PLC located on the same subnet. Protocol fuzzing yielded no additional vulnerabilities.

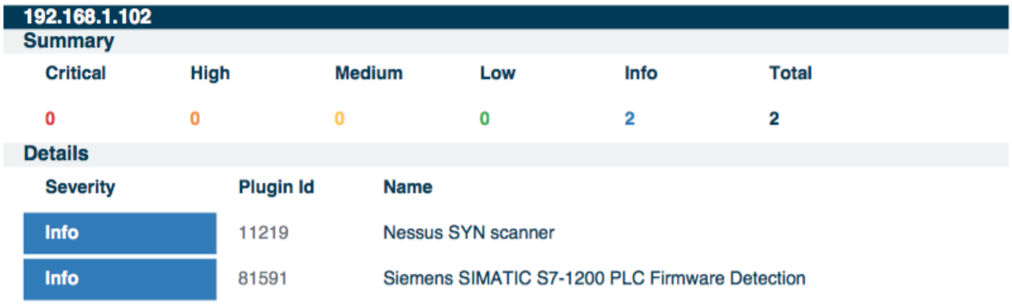
The *ET 200SP* forms part of Siemens' more recent/modern range of PLCs. Because of this, no CPE or CVE information could be found in the vFeed database. Research from other resources, such as Siemens Product Advisory, also found no publicly known vulnerabilities. The SNMP scan returned potential pivot points of the engineering workstation, another PLC, and a CCTV camera, all operating on the subnet. The session ID was not found, and the web interface was not configured, therefore no web server vulnerabilities were identified

Previously unknown vulnerability of ET 200SP.

Phase III of SimaticScan – protocol fuzzing – caused a denial of service on the ET200S, with only a small quantity of packets sent. This DoS vulnerability was not witnessed across the older S7-1200 and S7-300 PLCs. In accordance with the principles of responsible disclosure, we do not provide additional details regarding the exact vulnerability. Contact has been established with Siemens to study the vulnerability in more detail.

5.2. Comparing SimaticScan and Nessus

As discussed earlier, vulnerability scanners such as Nexpose and OpenVAS contain no plugins or procedures for checking the vulnerabilities of a PLC. Of the mainstream vulnerability scanners, only Nessus offers such functionality (Security (2015)).



192.168.1.102

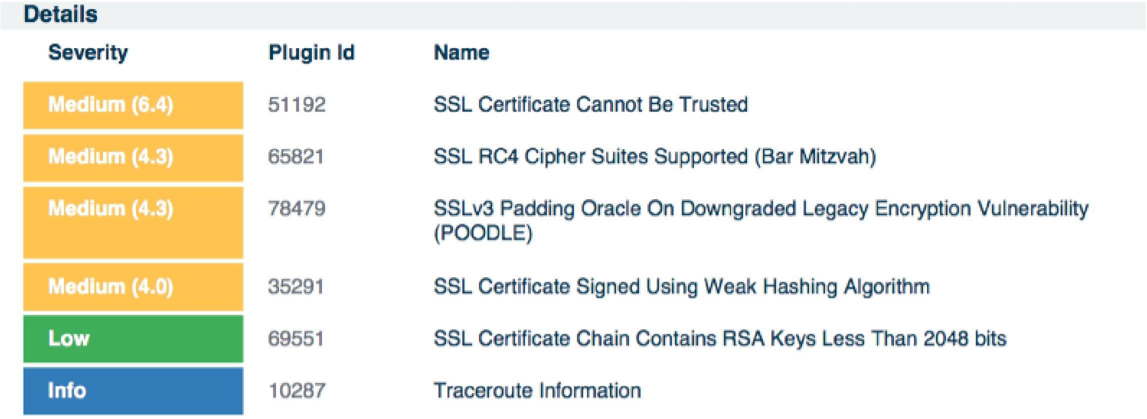
Summary

Critical	High	Medium	Low	Info	Total
0	0	0	0	2	2

Details

Severity	Plugin Id	Name
Info	11219	Nessus SYN scanner
Info	81591	Siemens SIMATIC S7-1200 PLC Firmware Detection

Figure 3: *Results of Nessus scan of S7-1200*



Details

Severity	Plugin Id	Name
Medium (6.4)	51192	SSL Certificate Cannot Be Trusted
Medium (4.3)	65821	SSL RC4 Cipher Suites Supported (Bar Mitzvah)
Medium (4.3)	78479	SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)
Medium (4.0)	35291	SSL Certificate Signed Using Weak Hashing Algorithm
Low	69551	SSL Certificate Chain Contains RSA Keys Less Than 2048 bits
Info	10287	Traceroute Information

Figure 4: *Results of a general Nessus scan*

Therefore, we contrast the output of SimaticScan and Nessus.

There exist a total of 61 SCADA plugins that Nessus uses when scanning for vulnerabilities in an ICS. As SimaticScan is concerned with the Siemens range of PLCs, other SCADA plugins for vendors, such as those from Schneider and Rockwell are disregarded.

Nessus uses the following plugins when performing a vulnerability scan of Siemens PLC devices:

- Siemens SIMATIC S7-1200 PLC Web Server Detection
- Siemens SIMATIC S7-1200 PLC $<$ 4.1 Open Redirection
- Siemens SIMATIC S7-1200 PLC Firmware Detection
- Siemens SIMATIC S7-1200 PLC $<$ 4.1.3 XSRF

These plugins are only associated with the Siemens S7-1200 PLC, hence we had no expectation to find vulnerabilities associated with the S7-300 and the ET 200SP.

Figure 3 shows the vulnerabilities reported by Nessus for the S7-1200. It can be seen that the only information that Nessus was able to retrieve was the PLC Firmware Detection. This plugin of Nessus did not actually return the firmware version number used to check if it is vulnerable according to any CVEs like SimaticScan does. Nor did it return the other vulnerabilities that SimaticScan detected.

Nessus was also unable to detect the web server residing on the S7-1200 PLC. However, when we conducted a Nessus “general” scan, it returned vulnerabilities associated with SSL certificates relating to the web server (cf. Figure 4). So while Nessus does not detect the web server for the PLC through its SCADA plugin, the general scan detects the PLC as a web server and the weak SSL certificates. In contrast, while SimaticScan detects the web server, it does not detect the SSL vulnerabilities.

5.3. SimaticScan vs. SCADA honeypot

Conpot is a configurable ICS/SCADA software honeypot that allows cybersecurity professionals and ICS owners to assess how potential threats would attack a real-world SCADA system (Conpot (2016)). A honeypot is a system that is intentionally misconfigured to be vulnerable. This has the effect in that any potential attackers are more likely to attack this more vulnerable system. We use the honeypot here as a benchmark measuring how SimaticScan performed against a known vulnerable target.

Conpot allows real hardware, such as PLCs to be simulated by creating protocol stacks and templates in software. Using its default template, it simulates a Siemens SIMATIC S7-200 PLC, as well as MODBUS TCP and other protocols, such as, SNMP and HTTP.

We ran SimaticScan against Conpot's default profile for a S7-200 PLC and made the following observations:

- SimaticScan identified two vulnerabilities: DoS attacks and default credentials in web interface.
- The software used to directly communicate with the (simulated) PLC was unavailable on the machine running SimaticScan. Therefore, session ID and plaintext vulnerabilities could not be found as traffic data between an engineering workstation and the PLC could not be established.
- TCP-hijack vulnerabilities could not be verified as the Conpot implementation is a virtual system that binds itself to the loopback (home) interface of the machine running SimaticScan. Therefore, an attempt to TCP-hijack the PLC cannot be made, as SimaticScan tries to hijack the connection originating from the Conpot servers, not the connection to the PLC.
- Read/write vulnerabilities could not be established, as the data blocks on the Conpot PLC had different values and could not be inferred due to time constraints.

The web interface vulnerability does not come as a surprise: it is a design decision to leave default credentials in Conpot's web server to entice attacks. On the other hand, the denial of service vulnerability could threaten the availability of the honeypot.

This experiment can be considered from two complementary angles:

- Evaluating how exhaustive a vulnerability scanner is with respect to established vulnerabilities, as implemented in honeypots.
- Evaluating how exhaustive and realistic a honeypot is when scanned and compared to the actual system it emulates.

As noted in the remarks above, some potential vulnerabilities could not be identified due to limitations on both sides: default configuration and lack of compatible software in Conpot, ability to capture loopback traffic or detect data blocks in SimaticScan. Such an experiment encourages a virtuous cycle, where scanner and honeypot discover each other's limitations and improve based on their mutual evaluations.

6. CONCLUSION AND FUTURE WORK

As the level of connectivity in ICS grows, it is increasingly important to develop defensive solutions that are specialised to such settings. Our comparison of SimaticScan and Nessus demonstrates the value that can be derived from hybrid IT/OT approaches, particularly when utilised with existing general purpose systems such as Nessus.

In order to further establish the importance of such specialised systems, we undertook a SHODAN search of the three PLCs we utilised in our evaluation. We discovered 141 S7-1200, 40 S7-300 and 14 ET 200SP PLCs connected to the global internet. Any unpatched vulnerabilities and/or previously unknown vulnerabilities in these PLCs pose a significant risk. Such risks can be mitigated through identification of vulnerabilities using specialised tools such as SimaticScan.

In future work, the code of SimaticScan will be consolidated and its functionality extended. The variety of equipment available in Lancaster University's tesbed allows the study of equipment from various providers, including, for instance, Siemens, Allen Bradley, and Schneider, for which we plan to build further specialised vulnerability scanners.

Acknowledgements. This work is supported in part by the EPSRC/Chist-Era grant: DYPOSIT: Dynamic Policies for Shared Cyber-Physical Infrastructures under Attack (EP/N021657/1).

REFERENCES

Beresford, D. (2011), 'Exploiting siemens simatic s7 PLCs', https://media.blackhat.com/bh-us-11/Beresford/BH_US11_Beresford_S7_PLCs_WP.pdf. Last accessed 12 April 2016.

Bristow, M. (2008), 'ModScan - a SCADA MODBUS network scanner', https://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-bristow.pdf. Last accessed 12 April 2016.

Conpot (2016), 'Conpot', http://conpot.org. Last accessed 24 February 2016.

Digital Bond (2016), 'PLCScan', http://www.digitalbond.com/tools/plcscan/. Last accessed 12 April 2016.

Green, B., Frey, S., Rashid, A. and Hutchison, D. (2016), Testbed diversity as a fundamental principle for effective ICS security research, *in* 'Proceedings of the First International Workshop on Security and Resilience of Cyber-Physical Infrastructures (SERECIN)', Lancaster University Technical Report SCC-2016-01, pp. 12–15.

Green, B., Paske, B., Hutchison, D. and Prince, D. (2014), Design and construction of an industrial control system testbed, *in* 'PG Net - 15th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting'.

Infracritical (2014), 'Project SHINE findings report', http://www.slideshare.net/BobRadvanovsky/project-shine-findings-report-dated-1oct2014. Last accessed 12 April 2016.

Kitagawa, T., Hanaoka, M. and Kono, K. (2010), Aspfuzz: A state-aware protocol fuzzer based on application-layer protocols, *in* 'Computers and Communications (ISCC), 2010 IEEE Symposium on', pp. 202–208.

secdev.org (2016), 'Scapy', http://www.secdev.org/projects/scapy/. Last accessed 5 May 2016.

Security, T. N. (2015), 'Nessus plugin id 51192', http://www.tenable.com/plugins/index.php?view=single&id=51192. Last accessed 8th May 2016.

Shodan (2016), 'Shodan search engine', https://www.shodan.io. Last accessed 17 February 2016.

Siemens (2016*a*), 'Et200s', http://w3.siemens.com/mcms/programmable-logic-controller/en/distributed-controller/et200sp-based/Pages/default.aspx. Last accessed 24 February 2016.

Siemens (2016*b*), 'S71200', http://w3.siemens.com/mcms/programmable-logic-controller/en/basic-controller/s7-1200/pages/default.aspx. Last accessed 24 February 2016.

Siemens (2016*c*), 'S7300', https://mall.industry.siemens.com/mall/en/ww/catalog/products/5000013?activeTab=order&regionUrl=WW#More%20information. Last accessed 24 February 2016.

Snap7 (2016), 'Snap7', http://snap7.sourceforge.net/. Last accessed 8th May 2016.

StrangeLove, S. (2016), 'Default / hardcoded scada password list', https://github.com/scadastrangelove/SCADAPASS/blob/master/scadapass.csv. Last accessed 8th May 2016.

Toolswatch (2016), 'vfeed', https://github.com/toolswatch/vFeed. Last accessed 5 May 2016.

TrendMicro (2016), 'First malware-driven power outage reported in ukraine', http://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/first-malware-driven-power-outage-reported-in-ukraine. Last accessed 12 January 2016.

Wang, T., Xiong, Q., Gao, H., Peng, Y., Dai, Z. and Yi, S. (2013), Design and implementation of fuzzing technology for opc protocol, *in* 'Intelligent Information Hiding and Multimedia Signal Processing, 2013 Ninth International Conference on', pp. 424–428.