Benchmarking Vulnerability Assessment Tools for Enhanced Cyber-Physical System (CPS) Resiliency

By

Emma McMahon

A Master's Paper Submitted to the Faculty of the

DEPARTMENT OF MANAGEMENT INFORMATION SYSTEMS

ELLER COLLEGE OF MANAGEMENT

In Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2018

STATEMENT BY AUTHOR

This paper has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona.

Brief quotations from this paper are allowable without special permission, provided that an accurate acknowledgement of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part must be obtained from the author.

SIGNED: Emma McMahon

APPROVAL BY MASTERS PAPER ADVISOR

This paper has been approved on the date shown below:

_____ _____

Dr. Mark Patton Date

Lecturer of Management Information Systems

Table of Contents

Table of Figures

Table of Tables

Abstract

Cyber-Physical Systems (CPSs) are engineered systems seamlessly integrating computational intelligence and physical components. CPS advances offer numerous benefits to domains such as health, transportation, smart homes and manufacturing. Despite these advances, the overall cybersecurity posture of CPS devices remains unclear. In this paper, we provide knowledge on how to improve CPS resiliency by evaluating and comparing the accuracy, suitability, and scalability of two popular vulnerability assessment tools, Nessus and OpenVAS. Accuracy and suitability are evaluated with a diverse sample of pre-defined vulnerabilities in Industrial Control Systems (ICS), smart cars, smart home devices, and a smart water system. Scalability is evaluated using a large-scale vulnerability assessment of 1,000 Internet accessible CPS devices found on Shodan, the search engine for the Internet of Things (IoT). Assessment results indicate several CPS devices from major vendors suffer from critical vulnerabilities such as unsupported operating systems, OpenSSH vulnerabilities allowing unauthorized information disclosure, and PHP vulnerabilities susceptible to denial of service attacks.

Introduction

The number of cyber-attacks taking place increases each year. According to an article by CNBC, 918 data breaches occurred in the first six months of 2017 (Graham, 2017). This was a 164% increase from the previous year and led to the compromise of 1.9 billion records. A recent cyber-attack targeted the Wolf Creek Nuclear Operating Corporation (Perlroth, 2017). Attackers were able to penetrate the plant's network by sending emails containing malicious documents which were posed as resumes to senior engineers.

Critical infrastructure (i.e., smart grid, nuclear power plants) has become a prime target for adversaries. Devices used in the energy industry having both physical and computational components are labeled as cyber-physical systems (CPS). According to the National Science Foundation, "cyber-physical systems (CPS) are engineered systems that are built from, and depend upon, the seamless integration of computational algorithms and physical components." Figure 1 illustrates the core functionality of a standard CPS.

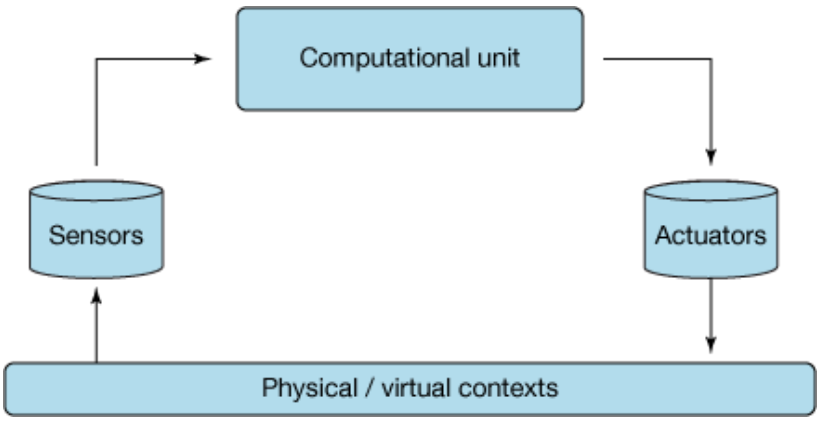

Figure 1. *CPS Functionality (Zanni, 2015)*

This diagram outlines the two critical components of a CPS and how they interact with each other. Data is collected by sensors within the physical then transferred to the computational unit.

From there, the data is processed then sent to the actuators for control. Finally, the information is sent back to the physical unit. This process is repeated in an endless cycle.

According to the National Science Foundation (NSF), CPSs have the potential to "enable capability, adaptability, scalability, resiliency, safety, security, and usability" beyond what is offered by embedded systems (National Science Foundation). At their inception, many of these devices were not intended to have networking capabilities. This presents a major concern as many vulnerabilities they now face were not existent when the systems were designed. Although CPSs provide endless possibilities, security is a serious consideration. As such, it is imperative that professionals be aware of the vulnerabilities they face and associated mitigation strategies.

One exercise used to proactively protect cyber assets is conducting vulnerability assessments. "A vulnerability assessment is the process of identifying and quantifying security vulnerabilities in an environment" (Drew, 2015). Vulnerability assessments are useful as they provide professionals the opportunity to identify their systems' vulnerabilities prior to an adversary exploiting them. Once an organization has found their systems' vulnerabilities, they can issue the appropriate patches/updates.

Several studies have been conducted regarding the security and design of CPSs (Ly and Jin, 2016; DiMase et al., 2015; Fitzgerald et al., 2015; Pasqualetti, 2013; Shafi, 2012). However, no study has analyzed common vulnerabilities faced by CPSs or assessed the performance of vulnerability assessment scanners against CPSs. The purpose of this study is to benchmark the performance of two popular vulnerability assessment scanners (Nessus and OpenVAS) on different types of CPS (e.g., energy, Internet of Things, Industrial Control Systems).

The remainder of this paper is organized as follows. First, we reviewed literature on CPSs, vulnerability assessment, and benchmarking methodologies. Second, we will present our

research design and testbed. Third, key findings of our research are summarized. Finally, a conclusion of our study and several promising future directions are provided.

Literature Review

For this research, three areas of literature were reviewed:

- **CPSs:** identify the various types of devices and associated security concerns
- **Vulnerability assessment:** review vulnerability assessment tools currently available
- **Benchmarking:** understand the steps to take when benchmarking multiple items and determine acceptable measures when benchmarking vulnerability assessment tools

Analyzing these domains will provide a comprehensive understanding of how to conduct a successful benchmark of vulnerability assessment scanners against CPSs.

Cyber-Physical Systems

As previously mentioned, cyber-physical systems are defined as "co-engineered interacting networks of physical and computational components" (Thompson, 2017). The distinguishing factor of CPSs is the interaction of physical components with computational components. Antsaklis outlined the seven defining characteristics of a CPS (2014):

- Cyber capabilities (i.e., networking, computation) for all physical components
- Complex spatial and temporal scales
- Dynamic reorganization and reconfiguration
- Closed control loops at each temporal/spatial scale
- Reliable and certifiable operation
- Close integration of computational and physical processes making attribution of behavioral features difficult
- System purpose is achieved through close interaction of cyber and physical components

Any system containing all seven attributes can be classified as a CPS. Currently, there are five primary industries in which CPS can commonly be found. Table 1 provides descriptions of each industry and example CPSs currently found within the industry (Antsaklis, 2014).

Industry	Description	Example Devices
Medical Care and Health	Delivery of medical products and services to patients	Body sensors, embedded micro-devices, implantable devices, wearable devices
Energy	Generate and provide energy services to consumers	"Smart" buildings, smart grid, nuclear reactor safety systems
Transportation and Mobility	Technology to enhance transportation services	Vehicle-to-vehicle communication (improve safety), autonomous vehicles, next generation military vehicles
Manufacturing	Subsector focused on the production of goods	Printing, casting, process streamlining technology, robotics working simultaneously with people
Materials and Other Sectors	Development of new technologies to accommodate consumers' needs	"Smart" fabrics, wearable technologies (i.e., activity trackers, smart watches)

Table 1. *CPS Industries*

Although there are other industries within which CPSs can be found, the five provided are the most prominent. Therefore, the remainder of this paper will focus only on CPSs found within these industries.

Much of the existing literature on CPSs has focused on system architecture (Axelsson, 2015; Lee et al., 2015; Liu and Jiang, 2016). However, security of such devices has become a major concern in recent years. Humayed et al. (2017) identified the primary areas through which threats against CPSs arise:

- Assumed system isolation
- Heterogenous components
- Increased connectivity

- Software vulnerabilities
- Operating system vulnerabilities

Given the potential societal impact of CPSs, it is imperative that these security issues be addressed.

According to the Department of Homeland Security (DHS), resiliency is "the ability to prepare for and adapt to changing conditions and withstand and recover rapidly from disruptions" (2017). In the context of CPS, this refers to the systems' abilities to conform to changing system requirements and recover from disruptions to their operations. One study proposed solutions for CPS resilience in the context of safe school environments (Rajamaki et al., 2012); the solutions they proposed focused on controls, reporting, and design. Denker et. al (2012) distinguished two core components of resiliency in CPS:

- *Infrastructure resilience:* dependability of the devices and their networks
- *Information resilience:* dependability of the information recorded by the system

One method to enhance resiliency of CPSs is conducting a vulnerability assessment (Wang, 2015).

Vulnerability Assessment

To help identify vulnerabilities of CPS devices, organizations can leverage vulnerability assessment tools. The purpose of conducting a vulnerability assessment is to understand one's weaknesses before an adversary exploits them. There are several tools available for performing a vulnerability assessment. Appendix A provides a summary of some of the most popular web-application and network vulnerability scanners currently available.

Given the variety in vulnerability scanners, it is important to choose the right scanner based upon the assessment's purpose. In our analysis, the most critical attributes of vulnerability

assessment tools were the ability to scan CPSs and perform scans on a large scale. Several studies have been conducted focused on vulnerability assessments. Table 2 provides a summary of recent vulnerability assessment studies.

Based on this literature review, several different device categories have been leveraged for vulnerability assessments: Internet of Things (IoT), Supervisory Control and Data Acquisition (SCADA), and scientific instruments. Additionally, several different tools have been used when conducting these analyses: Shodan, Nessus, Burp Suite, Amazon Web Services (AWS), Open Vulnerability Assessment System (OpenVAS), and several others. Due to their scalability, ability to scan CPSs, and popularity in the INFOSEC community, this study will benchmark the performance of Nessus and OpenVAS. To understand which vulnerability assessment tool is optimal for scanning CPSs, it is important to understand how to perform a proper benchmark.

Year	Author	Focus	Data	Tools	Results
2017	El et al.	Benchmarked Burp Suite and Nessus vulnerability assessment tools on SCADA devices and scientific instruments	20,641 SCADA and 184 scientific devices	Shodan, Nessus, Burp Suite	Burp outperformed Nessus in accuracy and reporting false positives; Nessus was more scalable
2017	Williams et al.	Conducted a large-scale vulnerability assessment of IoT devices found on Shodan using Nessus	156,680 IoT devices	Shodan, Nessus	~13% of IoT devices contained vulnerabilities; of those vulnerabilities, ~10% were deemed "Critical"; several device types not anticipated to connect to the Internet now contain critical vulnerabilities
2016	Torkura et al.	Create a vulnerability assessment scanner focused on cloud security	2 EC2 instances, 3 databases	OpenVAS, AWS	Developed Cloud Aware Vulnerability Assessment System (CAVAS), implementing OpenVAS
2016	Mukhopadhyay et al.	Comparison of vulnerability assessment scanners	N/A	Skipfish, Wapiti, Arachni, Nessus, w3af, Acunetix, Websecurify	Chose to incorporate Nessus into the proposed framework due to its versatility
2016	Samtani et al.	Conducted a large-scale vulnerability assessment of SCADA systems found on Shodan using Nessus	20,461 SCADA systems	Shodan, Nessus	Identified critical vulnerabilities in SCADA devices using both passive and active vulnerability assessment techniques
2015	Casola et al.	Generate a tool capable of automatic configuration according to Security SLA specifications	N/A	OpenVAS, Chef	Introduced a Security SLA monitor to the SPECS project
2014	Chimmanee et al.	Compare three vulnerability scanners regarding search functionality, time, and vulnerability detection	26 devices	NetClarity Audito, Nessus, Retina	NetClarity Auditor has superior search capabilities and vulnerability detection; Nessus took less time to perform scans

Table 2. *Vulnerability Assessment Studies*

Benchmarking

Benchmarking is a common practice when conducting research. By definition, "a benchmark is a standard used to evaluate or measure something" (Bacon and Riddles, 2015). Benchmarking allows users to compare multiple items. It has become a common practice across multiple

industries including applying theoretical models to assess ice density, electrocatalyst activity, and integrated circuits (Bradenburg, 2015; McCrory, 2015; Nikonov et. al, 2013). El et al. (2017) is the only study that has benchmarked multiple vulnerability assessment tools in terms of scalability, accuracy, and false positive reporting. No matter the application, there are common attributes found in each benchmark. Table 3 outlines the key characteristics of a successful benchmark.

Attribute	Description
Relevance	Ensure results are easily understood by users and are not overly complex
Extendibility	Focus on a challenge that has created technological limitations
Repeatable	Easily allow others to repeat
Fairness	Verify that there is no bias in the results that could benefit a particular vendor
Verifiable	Provide reporting, auditing, and other materials showing validity of each measure
Scalable	Supports the analysis being conducted on large data sets
Economically Sustainable	Performance will not consume an extraordinary amount of resources

Table 3. *Key Benchmarking Attributes*

Combining the above characteristics will result in a valuable benchmark. However, there are other critical components of a benchmark. Additional criteria used to assess benchmarks are (Chen, 2014):

- Feature documentation
- Accuracy assessment
- Results verification

To address feature documentation, we will record the features offered by each of the scanners we use in our benchmarking (Chen, 2014). The accuracy of the scanners will be determined by using CPSs configured with vulnerabilities. This will allow us to assess the percentage of accurately identified vulnerabilities and the number of false positives. Finally, results obtained

by the vulnerability assessment tool can be verified by performing scans multiple times (Cornell, 2012). Doing this will ensure the scanners' consistency.

Research Gaps and Questions

After reviewing literature on CPSs, vulnerability assessments and benchmarking, several gaps were identified. First, no study has identified common vulnerabilities across the different categories of CPSs. Second, no study has benchmarked the performance of Nessus and OpenVAS. Finally, no study has benchmarked the performance of vulnerability assessment scanners against CPSs. Based on these gaps, we have posed the following questions:

- What vulnerabilities are commonly seen among the different categories of CPS?
- Between Nessus and OpenVAS, which scanner performs better in terms of scalability, accuracy, and false positive reporting?
- What vulnerability assessment tool should be used when analyzing CPSs?

Research Testbed and Design

Testbed Selection

Our research design (Figure 2) has been separated into two components, each requiring their own testbed. One aspect was used to determine the scanners' scalability whereas the other compared the scanners accuracy and false positive reporting.

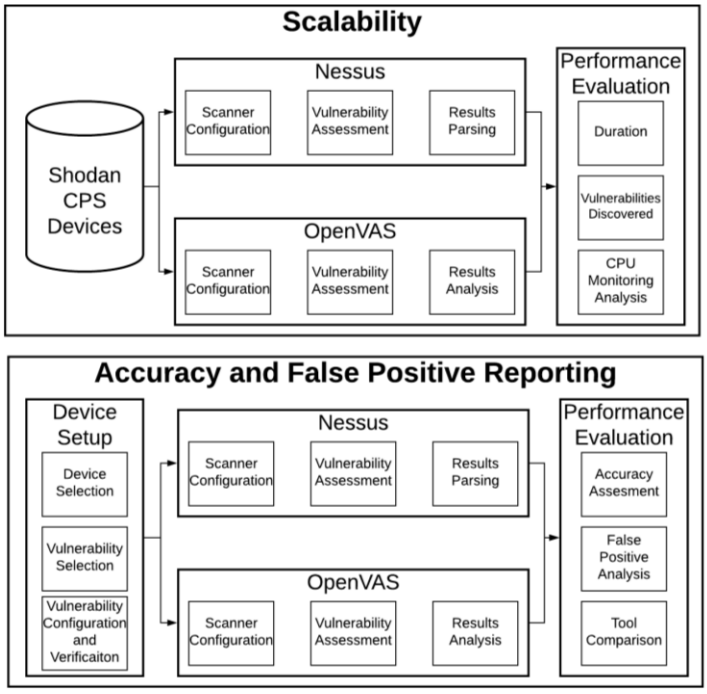

Figure 2. *Research Design*

To gather a large number of CPS devices for the scalability analysis, we used Shodan, a search engine for the IoT. After identifying common communication ports used by devices within the five CPS industries, these ports were then passed through Shodan. Table 4 lists the common communication ports and their respective industry.

Industry	Ports
Medical Care and Health	17729, 17754, 17755, 17756
Energy	502, 20000, 102, 19999, 4800, 4900, 8000
Transportation and Mobility	1, 1024
Manufacturing	7878
Materials and Other Sectors	1, 1024, 17729, 17754, 17755, 17756

Table 4. *CPS Communication Ports*

Using the identified CPS ports, we identified 262,713 unique CPS devices on Shodan. Shodan returns the IP address, open ports, city, country, organization, Internet Service Provider (ISP), date of last update, and Autonomous System Number (ASN) of each device as well as the services running. Figure 3 provides a sample device returned by a Shodan search.

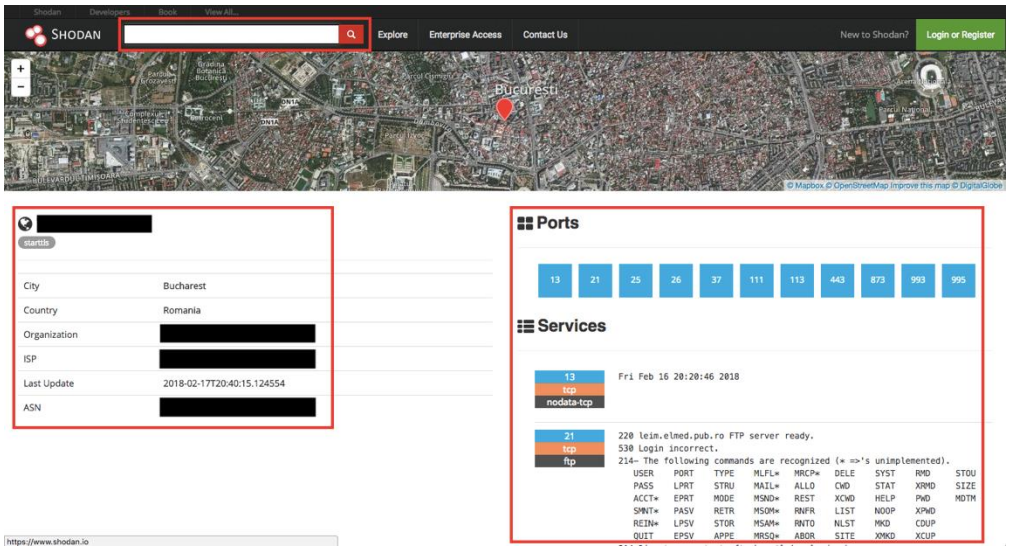

Figure 3. *Sample Shodan Device Search*

Information about each device was then stored in a MySQL database. Using a simple SQL query, the most common open ports and associated protocols can be seen. Results from this query are depicted below (Table 5).

Port	Associated Services	Number of Devices
21	File Transfer Protocol (FTP)	142,210
111	Remote Procedure Call (RPC)	99,372
443	Secure HTTP (HTTPS)	7,705
81	HOSTS2 Name Server	3,044
161	Simple Network Management Protocol (SNMP)	2,087
8081	HTTP	1,429

Table 5. *Common Open Ports in Shodan Identified CPSs*

Devices found on Shodan were used to assess the scalability of Nessus and OpenVAS, but local, vulnerable CPSs were used to analyze the tools' accuracy and false positive reporting. The four CPS devices included in this study were: a smart home, a smart water system, a smart car, and an Industrial Control System (ICS). Common vulnerabilities for each type of device were identified. We then configured each system with the appropriate vulnerability/vulnerabilities. Table 6 summarizes the devices used as well as their configured vulnerabilities.

Device	Vulnerability	Source
Smart Home	Replay attack	Komninos et al., 2014
Smart Water System	Default credentials	Mo et al., 2012
Smart Car	Disabled firewall	Humayed et al., 2017
ICS	Default credentials	Correa, 2017
ICS	Unsupported operating system	Correa, 2017
ICS	Weak SSH Encryption	Correa, 2017
Router	Default credentials	Hendriks, 2017

Table 6. *CPS Devices and Configured Vulnerabilities*

Although it is not classified as a CPS, the router connected to the devices was also tested as this provides a portal to these devices.

Research Design

Scalability

To determine the scalability of Nessus and OpenVAS, we created two Amazon Elastic Compute Cloud (EC2) instances for each scanner. The EC2 instances were configured with the following specifications:

- *Small:* 4 cores, 16 GB RAM, 80 GB SSD
- *Large:* 8 cores, 32 GB RAM, 80 GB SSD

In total, four EC2 instances were built for this study: Nessus Small, Nessus Large, OpenVAS Small, and OpenVAS Large. To assess the scalability of both tools, 1,000 random IP addresses were selected from the 262,713 testbed. Then, the scanners were configured to identify vulnerabilities pertinent to CPSs and disable port scanning. Of the over 80,000 plug-ins offered by Nessus, only those relevant to CPSs were enabled. Examples of plug-in families selected include default Unix accounts, firewalls, Denial of Service (DoS), and mobile devices. OpenVAS offers seven different types of scanners. The Full and Fast scanner is commonly used in industry (Hu et al., 2016). One reason for its popularity is its comprehensive coverage including 54,636 Network Vulnerability Tests (NVTs). Despite the completeness of the scans,

Full and Fast scanners are able to optimize their performance by utilizing previously gathered information. Therefore, we used the Full and Fast scanner for our scalability analysis.

Once the scanners' setup was complete, the 1,000 IP addresses were then passed through each. The scans were continuously monitored until they were finished. Upon completion, Nessus scan results were then imported into a MySQL database for analysis. By default, OpenVAS creates a SQLite database and stores all the information there. After the vulnerability assessments concluded, common vulnerabilities were identified, scan durations were compared, and CPU utilization was gathered.

Accuracy and False Positive Reporting

To assess the accuracy and false positive reporting of the scanners, four CPSs within a local lab environment operated by our colleagues were configured with vulnerabilities. The configurations are outlined in Table 6. Each vulnerability was then verified through exploitation. After the scanners completed scanning each device, reports generated by the scanners were analyzed to determine whether the vulnerabilities were identified. Finally, additional vulnerabilities reported by both scanners with a severity of low or higher were reviewed. To verify the vulnerabilities were false positive, the system configuration was examined and exploit attempts were made. If the system configuration shows no indication of the vulnerability and attacks were unsuccessful, the vulnerability is deemed a false positive.

Results and Discussion

Scalability

Nessus

Using the afore mentioned EC2 instance specifications, Nessus scans were run concurrently on a small machine and large machine. Results of the Nessus large-scale vulnerability assessment are shown in Table 7.

Metric	Nessus Small	Nessus Large
Duration	13 hours 1 minute	26 hours 56 minutes
Vulnerable Devices	973	974
Types of Vulnerabilities	498	560
Total Number Vulnerabilities	21,849	24,222
Devices Containing Non-Informational Vulnerabilities	653	682
Non-Informational Vulnerabilities	4,180	5,159
Average CPU Utilization	30%	13.1%

Table 7. *Nessus Scalability Results*

Reviewing this table, it is clear that the two machines experienced different results. First, we noticed the difference in time required to complete scans on both machines. Although we predicted the large machine with greater resources to complete the scans in less time, the small machine completed its scans much quicker. The scanners were run at separate times to prevent interference. To our knowledge, there were no other known variances that would cause these results. As a result of the longer scan completion time, the CPU utilization of the large machine was much lower than that of the small machine. Furthermore, there was a significant difference in the total number of vulnerabilities reported. Therefore, we chose to compare the results reported by both Nessus machines. This analysis is depicted by Table 8.

Metric	Result
Devices	972
Vulnerability Types	484
Non-Informational Vulnerability Types	273
Informational Results	13,862
Non-Informational Vulnerabilities	682

Table 8. *Nessus Machines Similar Results*

From this, the number devices containing vulnerabilities from both machines was almost identical. However, the vulnerabilities reported were drastically different (note: for results to be the same the IP address and vulnerability must match). As an example, the number of non-informational vulnerabilities from the small and large instances decreased from 4,180 and 5,159 respectively to 682. Based on our analysis, there were 2,606 vulnerabilities reported by the small machine not found by the large machine. Additionally, the large instance reported 3,704 vulnerabilities not identified by the small instance.

Table 9 on the next page provides common vulnerabilities reported by Nessus at the critical, high, and medium levels on both machines. The numbers presented show the number of appearances of each vulnerability per instance, but the vulnerabilities could have existed on different machines. At the critical level, the most common vulnerabilities were Unix unsupported operating systems, PHP vulnerabilities, and unsupported versions of PHP. Common vulnerabilities found at the high level include Apache vulnerabilities, OpenSSL vulnerabilities, and SNMP agent default community names. Finally, at the medium risk level, the vulnerabilities most prevalent were HTTP trace/track methods allowed, SSL untrusted certificate, and SSL self-signed certificates. Potential impacts of exploiting these vulnerabilities include man-in-the-middle attacks, denial of service, remote code execution, authentication bypass, and several others.

Risk	Vulnerability	Appearances (Small)	Appearances (Large)	Appearances (Both)	Potential Impact
Critical	Unix Unsupported Operating System	54	55	53	Vendor no longer issues security patches making vulnerabilities more likely
	PHP Multiple Vulnerabilities	22	44	12	Denial of service, remote code execution
	PHP Unsupported Version	17	23	16	Vendor no longer issues security patches making vulnerabilities more likely
High	Apache Multiple Vulnerabilities	55	75	50	Authentication bypass
	OpenSSL Multiple Vulnerabilities	48	54	36	Denial of service, plaintext recovery attack
	SNMP Agent Default Community Name	25	25	25	Reconnaissance, change host configuration
Medium	HTTP Trace/Track Methods Allowed	264	330	84	Web server information disclosure
	SSL Untrusted Certificate	247	253	99	Man-in-the-middle
	SSL Self-signed Certificate	231	239	94	Man-in-the-middle

Table 9. *Common Nessus Vulnerabilities*

OpenVAS

Identical to the process used for Nessus scans, two Amazon EC2 instances, with the previously discussed specifications, were created to host OpenVAS scans. Table 10 summarizes the results of the large-scale vulnerability assessment using OpenVAS.

Metric	OpenVAS Small	OpenVAS Large
Duration	386 hours 8 minutes	372 hours 37 minutes
Vulnerable Devices	1,000	1,000
Types of Vulnerabilities	3,612	3,711
Total Number Vulnerabilities	227,079	246,000
Devices Containing Non-Informational Vulnerabilities	863	869
Non-Informational Vulnerabilities	38,256	38,075
Average CPU Utilization	0.17%	0.09%

Table 10. *OpenVAS Scalability Results*

Based on the results, both machines using OpenVAS took just over two weeks to scan 1,000 IP addresses. CPU utilization of the large instance was approximately half of the CPU utilized by the small instance. Given the trivial CPU utilization, multiple OpenVAS scanners can be run simultaneously on a single machine. All devices were classified as containing vulnerabilities on both machines, but 863 devices on the small machine and 869 devices on the large machine had vulnerabilities with at least a low severity. Given the discrepancies, we chose to compare the results observed by both OpenVAS machines (Table 11).

Metric	Result
Devices	1,000
Vulnerability Types	3,574
Non-Informational Vulnerability Types	1,027
Informational Results	136,144
Non-Informational Vulnerabilities	25,606

Table 11. *OpenVAS Machines Similar Results*

As both machines found all devices to contain vulnerabilities, the number of devices remained the same. The biggest differences can be seen in the number of non-informational

vulnerabilities reported. Comparing the 38,256 vulnerabilities from the small instance to the 38,075 vulnerabilities on the large instance, only 25,606 vulnerabilities were the same. These inconsistencies demonstrate a strong need to use other vulnerability assessment scanners (e.g., Nessus). Frequent vulnerabilities found at the top three risk levels on OpenVAS are presented in Table 12.

Risk	Vulnerability	Appearances (Small)	Appearances (Large)	Appearances (Both)	Potential Impact
Critical	OpenSSH Multiple Vulnerabilities	406	405	392	Unauthorized information disclosure and modification
	Apache Web Server End of Life	209	206	115	Vulnerabilities likely exist as the vendor is longer issuing security patches
	PHP Type Confusion	180	175	104	Denial of service, remote code execution
High	Multiple CRLF Injection Vulnerabilities	407	406	381	Allow authenticated remote users to bypass shell-command restrictions
	OpenSSH X11 Forwarding Security Bypass	406	405	392	Unauthorized information disclosure and modification
	OpenSSH Denial of Service	406	405	392	Denial of service
Medium	Insufficient Diffie-Hellman Key Exchange	1,142	1,158	1,126	Decrypt SSL/TLS communication offline
	Cryptographic Issues	962	954	937	Plaintext recovery attack
	TCP Timestamps	775	783	765	Calculate uptime of computer

Table 12. *Common OpenVAS Vulnerabilities*

At the critical level, common vulnerabilities were OpenSSH vulnerabilities, Apache server end of life, and PHP type confusion. Next, at the high risk level, there were several appearances of the following vulnerabilities: CRLF injection vulnerabilities, OpenSSH security bypass, and OpenSSH denial of service. Finally, medium risk vulnerabilities that appeared often were insufficient key exchanges, cryptographic vulnerabilities, and TCP timestamps available. Risks

posed by theses vulnerabilities include unauthorized information disclosure and modification, denial of service, remote code execution, and plaintext recovery.

Accuracy and False Positive Reporting

In addition to scalability, it was also important to assess the accuracy and false positive reporting of Nessus and OpenVAS. Table 13 shows the accuracy of the scanners when identifying vulnerabilities pre-configured on the devices.

Device	Vulnerability	Nessus	OpenVAS
Smart Home	Replay Attack	X	X
Smart Water System	Default Credentials	X	X
Smart Car	Disabled Firewall	✓	✓
ICS	Default Credentials	X	✓
ICS	Unsupported Operating System	✓	✓
ICS	Weak SSH Encryption	✓	✓
Router	Default Credentials	X	X

Table 13. *Vulnerability Scanners' Accuracy (✓symbolizes accurate identification; X symbolizes a miss)*

As demonstrated by the results in Table 13, Nessus experienced an accuracy of 42.86% and OpenVAS achieved 57.14% accuracy. The only difference between the performance of the two scanners was OpenVAS' ability to detect default credentials on an ICS (critical vulnerability). Overall, both scanners did not perform as expected. There are several reasons that could explain these results, but the most apparent is misconfiguration of the tool. However, each scan was run multiple times for each vulnerability with both scanners being analyzed each time. In their study, El et al. (2017) also experienced poor accuracy of multiple vulnerability assessment scanners.

The final measure by which the scanners were benchmarked was the number of false positive vulnerabilities they reported. In addition to correctly identifying vulnerabilities on a system, it is important to note vulnerabilities falsely reported by each scanner. False positives create

unnecessary overhead for security experts and can divert their attention from real danger. Using the same testbed of four local CPS devices, we were able to quantify the number of falsely reported vulnerabilities by each scanner. When analyzing vulnerabilities for false positives, we only considered vulnerabilities with at least a low severity that had not been created for the accuracy assessment. According to the reports, Nessus identified two vulnerabilities that were non-existent:

- Enabled IP Forwarding (Medium severity): 2 appearances
- Disabled SMB Signing (Medium severity): 2 appearances

To verify the IP forwarding vulnerability was a false positive, we checked the systems' configurations through the command line as both used Linux operating systems. Results demonstrated no evidence of either appearance. Finally, to verify that SMB signing had been disables, we reviewed online instructions provided by Microsoft showing users how to enable SMB signing on their machines. However, following these instructions, we saw the proper configurations were already in place.

Contrarily, OpenVAS did not report any nonexistent vulnerabilities on our four CPSs with at least a low severity.

Conclusion

This research aims to find common vulnerabilities among CPSs as well as benchmarking state-of-the-art vulnerability assessment tools on such devices. Based on the features they offer, the vulnerability assessment scanners chosen for this study were Nessus and OpenVAS. The performance of these tools was benchmarked in terms of scalability, accuracy, and false positive reporting. A summary of their performance at each level is provided in Table 14.

Benchmarking Dimension	Preferable Scanner
Scalability	Nessus
Accuracy	OpenVAS
False Positive Reporting	OpenVAS

Table 14. *Nessus and OpenVAS Benchmarking Performance Overview*

Given the various strengths of different vulnerability assessment scanners, it is preferable to use multiple tools simultaneously to obtain a comprehensive overview of one's threat landscape.

As cyber security becomes a greater concern, it is imperative that individuals and organizations protect their systems. Leveraging vulnerability assessment scanners tailored to their systems is one way of doing so. Based on our findings, there are several potential future directions of this work:

- OpenVAS and Nessus can be benchmarked against other types of devices found within the INFOSEC community (i.e., scientific instruments, IoT)
- CPSs can be leveraged to assess the performance of other vulnerability assessment scanners (i.e., QualysGuard, Retina, Nexpose)
- Improve vulnerability assessment scalability to provide more real-time data
- Understanding discrepancies in Nessus scanning time and machine resources

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. NSF DUE-1303362 (SFS). Additionally, I would like to thank Pratik Satam, Dr. Sagar Samtani, Dr. Salim Hariri, Dr. Mark Patton, and Dr. Hsinchun Chen for guidance with my research.

References

Angles, Renzo et al. “D1.1.2 Benchmark Principles and Methods.” Print.

Axelsson, Jakob. “Architectural Allocation Alternatives and Associated Concerns in Cyber-Physical Systems.” *ECSAW 2015 - European Conference on Software Architecture Workshops* (2015): 1–6. Web.

Antsaklis, Panos. “Goals and Challenges in Cyber-Physical Systems Research Editorial of the Editor in Chief.” *IEEE Transactions on Automatic Control* 59.12 (2014): 3117–3119. Web.

Bacon, Carl, and Neil E Riddles. “Does Your Benchmark Measure Up?” *CFA institute* 32.1 (2015): 72–80. Print.

Brandenburg, Jan Gerit, Tilo Maas, and Stefan Grimme. “Benchmarking DFT and Semiempirical Methods on Structures and Lattice Energies for Ten Ice Polymorphs.” *Journal of Chemical Physics* 142.12 (2015): n. pag. Web.

Bogetoft, Peter. *Performance Benchmarking: Measuring and Managing Performance.* Springer, New York, 2012.

Casola, Valentina, Alessandra De Benedictis, and Massimiliano Rak. “Security Monitoring in the Cloud: An SLA-Based Approach.” *Proceedings - 10th International Conference on Availability, Reliability and Security, ARES 2015* September 2016 (2015): 749–755. Web.

Chen, S. (2014). The Web Application Vulnerability Scanners Benchmark. Denim Group.

Cornell, D. (2012). Benchmarking Web Application Scanners for YOUR Organization. Denim Group

Correa, Danielle. “Six Key Vulnerabilities Identified within Industrial Control Systems.” *SC Media UK*, 12 Apr. 2017, www.scmagazineuk.com/six-key-vulnerabilities-identified-within-industrial-control-systems/article/649993/.

“Cyber-Physical Systems (CPS).” *National Science Foundation*, www.nsf.gov/funding/pgm_summ.jsp?pims_id=503286.

“Cyber Physical Systems Security.” *Department of Homeland Security*, www.dhs.gov/science-and-technology/csd-cpssec.

“CPS PWG Cyber-Physical Systems (CPS) Framework Release 1.0.” *Cyber-Physical Systems Public Working Group*, NIST, pages.nist.gov/cpspwg/.

Denker, Grit et al. “Resilient Dependable Cyber-Physical Systems: A Middleware Perspective.” *Journal of Internet Services and Applications* 3.1 (2012): 41–49. Web.

DiMase, Daniel et al. “Systems Engineering Framework for Cyber Physical Security and Resilience.” *Environment Systems and Decisions* 35.2 (2015): 291–300. Web.

Drew, Steven. “Vulnerability Assessments Versus Penetration Tests.” *SecureWorks*, 8 Apr. 2015, www.secureworks.com/blog/vulnerability-assessments-versus-penetration-tests.

Fitzgerald, John et al. “Cyber-Physical Systems Design: Formal Foundations, Methods and Integrated Tool Chains.” *Proceedings - 3rd FME Workshop on Formal Methods in Software Engineering, Formalise 2015* (2015): 40–46. Web.

Graham, Luke. “The Number of Devastating Cyberattacks Is Surging — and It's Likely to Get Much Worse.” *CNBC*, CNBC, 20 Sept. 2017, www.cnbc.com/2017/09/20/cyberattacks-are-surging-and-more-data-records-are-stolen.html.

Hendriks, Calvin. “Fixing the Average Internet User ’ S IoT Vulnerabilities.” 1–10. Print.

Hu, Yi et al. “Efficient Distributed Vulnerability Assessment by Utilizing Miniaturized Computers.” 7.4 (2016): 782–790. Print.

Humayed, Abdulmalik et al. “Cyber-Physical Systems Security -- A Survey.” *IEEE Internet of Things Journal* (2017): 1–1. Web.

“Intelligence and Control.” *InfoBeyond*, www.infobeyondtech.com/cn/Info.asp?gCateID=78125387.0.

Komninos, Nikos, Eleni Philippou, and Andreas Pitsillides. “Survey in Smart Grid and Smart Home Security: Issues, Challenges and Countermeasures.” *IEEE Communications Surveys and Tutorials* 16.4 (2014): 1933–1954. Web.

Lee, Jay, Behrad Bagheri, and Hung An Kao. “A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems.” *Manufacturing Letters* 3.December (2015): 18–23. Web.

Liu, Chao, and Pingyu Jiang. “A Cyber-Physical System Architecture in Shop Floor for Intelligent Manufacturing.” *Procedia CIRP* 56 (2016): 372–377. Web.

Ly, Kelvin, and Yier Jin. “Security Challenges in CPS and IoT: From End-Node to the System.” *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI* 2016–September (2016): 63–68. Web.

McCrory, Charles C. L. et al. “Benchmarking HER and OER Electrocatalysts for Solar Water Splitting Devices - SI.” *J. Am. Chem. Soc.* 137.13 (2015): 4347–4357. Web.

Mcquade, Kinnaird. “Open Source Web Vulnerability Scanners : The Cost Effective Choice ?” *2014 Proceedings of the Conference for Information Systems Applied Research* November (2014): 1–13. Web.

Mo, Yilin et al. “Cyber-Physical Security of a Smart Grid Infrastructure.” *Proceedings of the IEEE* 100.1 (2012): 195–209. Web.

Nikonov, Dmitri E., and Ian A. Young. “Overview of beyond-CMOS Devices and a Uniform Methodology for Their Benchmarking.” *Proceedings of the IEEE* 101.12 (2013): 2498–2533. Web.

Pasqualetti, Fabio, F Dorfler, and Francesco Bullo. “Attack Detection and Identification in Cyber-Physical Systems.” *IEEE Transactions on Automatic Control* 58.11 (2013): 2715–2729. Web.

Perlroth, Nicole. “Hackers Are Targeting Nuclear Facilities, Homeland Security Dept. and F.B.I. Say.” *The New York Times*, The New York Times, 6 July 2017, www.nytimes.com/2017/07/06/technology/nuclear-plant-hack-report.html.

Rajamaki, Jyri et al. “Resilience of Cyber-Physical System: A Case Study of Safe School Environment.” *2012 European Intelligence and Security Informatics Conference* August 2012 (2012): 285–285. Web.

Samtani, Sagar et al. “Identifying SCADA Vulnerabilities Using Passive and Active Vulnerability Assessment Techniques.” n. pag. Print.

Shafi, Qaisar. “Cyber Physical Systems Security: A Brief Survey.” *2012 12th International Conference on Computational Science and Its Applications*, 2012, doi:10.1109/iccsa.2012.36.

Thompson, Kristy D. “Cyber-Physical Systems.” *NIST*, 26 Jan. 2017, www.nist.gov/el/cyber-physical-systems.

Torkura, Kennedy A., and Christoph Meinel. “Towards Cloud-Aware Vulnerability Assessments.” *Proceedings - 11th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2015* September (2016): 746–751. Web.

Wang, Xiaotian. “Mission-Aware Vulnerability Assessment for Cyber-Physical System.” (2015). Print.

“What Is Security and Resilience?” *What Is Security and Resilience? | Homeland Security*, 12 July 2017, www.dhs.gov/what-security-and-resilience.

Zanni, Alessandro. “Cyber-Physical Systems and Smart Cities.” *IBM*, 20 Apr. 2015, www.ibm.com/developerworks/library/ba-cyber-physical-systems-and-smart-cities-iot/.

Appendix A: Vulnerability Assessment Scanners

Name	Type	Description	Provider	Number of Hosts	CPS Capable
Nessus	Network Vulnerability Scanner	Identifies vulnerabilities and configuration issues that would allow an attacker to penetrate the network	Tenable Network Security	Default: 30, Licensed: Unlimited	Yes
Burp Suite	Web Application Scanner	Program used to identify multiple types of vulnerabilities in web applications	PortSwigger	Multiple hosts via text file	Internet-enabled devices
QualysGuard	Network Vulnerability Scanner	Security tool used for auditing, compliance, and IT defense purposes	Qualys	Default: 30	Yes
Retina	Web Application Scanner	Web application vulnerability scanner that crawls web pages for OWASP and other web vulnerabilities	BeyondTrust	Community: 256	Internet-enabled devices
OpenVAS	Network Vulnerability Scanner	Vulnerability scanning and management software that uses information gathered from the scanner and transformed into intelligence	OpenVAS	Default: 30	Yes
Nexpose	Network Vulnerability Scanner	Performs vulnerability assessment and reviews policies to provide users with mitigation/remediation strategies	Rapid7	Default: 30, Express & Consultant: 1,024, Enterprise & Ultimate: Unlimited	Yes
Microsoft Baseline Security Analyzer (MBSA)	Network Vulnerability Scanner	Used to analyze Windows systems for outdated versions and misconfigurations	Microsoft	64	OS dependent
Acunetix	Web Application Scanner	Detects more than 3,000 web application vulnerabilities using DeepScan and AcuSensor technologies	Acunetix	Multiple hosts by running multiple instances	Internet-enabled devices
Netsparker	Web Application Scanner	Assessment tool used to analyze the security of a website; has desktop and cloud delivery	Netsparker	Multiple hosts by running multiple instances	Internet-enabled devices