IIoT Cybersecurity Risk Modeling for SCADA Systems

Gregory Falco, Carlos Caldera, and Howard Shrobe

***Abstract*—Urban critical infrastructure such as electric grids, water networks, and transportation systems are prime targets for cyberattacks. These systems are composed of connected devices which we call the Industrial Internet of Things (IIoT). An attack on urban critical infrastructure IIoT would cause considerable disruption to society. Supervisory control and data acquisition (SCADA) systems are typically used to control IIoT for urban critical infrastructure. Despite the clear need to understand the cyber risk to urban critical infrastructure, there is no data-driven model for evaluating SCADA software risk for IIoT devices. In this paper, we compare non-SCADA and SCADA systems and establish, using cosine similarity tests, that SCADA as a software subclass holds unique risk attributes for IIoT. We then disprove the commonly accepted notion that the common vulnerability scoring system risk metrics of exploitability and impact are not correlated with attack for the SCADA subclass of software. A series of statistical models are developed to identify SCADA risk metrics that can be used to evaluate the risk that a SCADA-related vulnerability is exploited. Based on our findings, we build a customizable SCADA risk prioritization schema that can be used by the security community to better understand SCADA-specific risk. Considering the distinct properties of SCADA systems, a data-driven prioritization schema will help researchers identify security gaps specific to this software subclass that is essential to our society's operations.**

***Index Terms*—Critical infrastructure, cybersecurity, industrial control systems (ICSs), Industrial IoT (IIoT), Internet of Things (IoT) security, risk, supervisory control and data acquisition (SCADA).**

I. Introduction

A. Problem Statement

CYBERATTACKS can easily disable Industrial Internet of Things (IIoT) devices responsible for urban critical infrastructure. Urban critical infrastructure includes smart grids, water networks, and transportation systems. In 2015, multiple power substations in Ukraine were compromised resulting in rolling power outages affecting 225 000 people [1]. Ukraine's supervisory control and data acquisition (SCADA) system that is responsible for controlling the smart grid's IIoT devices is vast and complicated such that it will be impossible to patch all vulnerabilities throughout the networks. While there are vulnerability taxonomies and cybersecurity frameworks that may help to mitigate risk, these tools do not provide data-driven guidance about SCADA security research priorities or a dynamic model to evaluate risk based on various operating parameters. This paper provides a risk analysis of critical infrastructure SCADA vulnerabilities and exploits using statistical methods. Further, the study offers technical SCADA IIoT design recommendations to help mitigate future system exploit risk.

Evaluating IIoT exploit risk is challenging. The problem is accentuated by findings of various security researchers that the common vulnerability scoring system (CVSS) risk metrics created by First.org and used by the Department of Homeland Security (DHS) and the National Institute of Standards and Technology (NIST) are not effective at predicting exploits [2], [3]. Further, NIST's cybersecurity framework that intends to help organizations evaluate cyber risk for industrial control systems (ICSs) faces adoption challenges and does not directly address exploit probability. Despite being labeled as best-in-class, reasons for slow adoption include the considerable time and expense required to implement the framework [4]. SCADA and critical infrastructure vulnerability taxonomies exist that could help to identify cyber risk [5]–[7]. While these taxonomies could be useful, the findings are not grounded in data-driven, empirical analysis which raises questions about their applicability to cyber risk in the field.

B. SCADA IIoT Overview

SCADA systems provide a supervisory control software layer across multiple programmable logic controllers (PLCs), which are a type of IIoT. SCADA systems are designed for use over long distances such as water or electric distribution. Because of these longer distances, there tends to be less control over the networks that use them. The 80% of U.S. utilities run on SCADA systems [8]. SCADA operates using telephony communication or other third party networks, which reduces the speed, frequency, and quality of communications [9]. For this reason, SCADA tends to be event driven meaning that data is only communicated from the devices to the software when there is a change in value [9]. Controlling other IIoT devices, SCADA systems require an operator console or human–machine interface (HMI) from which an engineer can view, command, and control the devices connected to the system [10]. This HMI is also vulnerable to attack where an attacker could intercept the PLCs data and alter it on the

Manuscript received January 23, 2018; revised March 11, 2018; accepted March 27, 2018. Date of publication April 6, 2018; date of current version January 16, 2019. This work was supported in part by Lockheed Martin and in part by CyberSecurity@CSAIL *(Corresponding author: Gregory Falco.)*

The authors are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139-4307 USA (e-mail: gfalco@mit.edu).

Digital Object Identifier 10.1109/JIOT.2018.2822842

HMI [11]. SCADA systems typically runs on a commercial off-the-shelf Windows PC which can expose the software to an array of operating system, Windows-based attacks [12]. A growing challenge is that there is an increased interest in connecting SCADA-based IIoT systems to IT networks. This can allow for hackers to access potentially vulnerable SCADA systems through backdoors using TCP/IP-based attacks.

C. National Policy and Regulatory Landscape

In 2013, Executive Order (EO) 13636: Improving Critical Infrastructure Cybersecurity was published. The EO encourages the adoption of cybersecurity best practices and mandated that the NIST develop new ways of assessing cybersecurity risk [13]. The EO falls short, however, because it is entirely voluntary and contains no incentive structures. Also, it puts the burden for taking action only on the shoulders of critical infrastructure operators [14]. While NIST created a strong cybersecurity framework—which is hailed in industry as best-in-class—the financial burden of implementing NIST's framework is a serious barrier to adoption [4]. A less time-intensive, expensive and streamlined alternative to NIST's recommendations is needed for the SCADA community.

Industry organizations like the North American Electric Reliability Corporation (NERC), have tried to step in [15]. For example, in 2008, NERC proposed critical infrastructure protection reliability standards to the Federal Energy Regulatory Commission (FERC) to improve security for the electric grid [16]. FERC has adopted these recommendations, mandating U.S. electric companies comply with all voluntary cybersecurity regulation. Extensive survey results from NERC revealed that there are loopholes in the regulation. This enabled 75% of companies to opt-out of cybersecurity regulation while those companies that could not opt-out preferred to pay fines rather than update their system security [17].

D. Vulnerability Identification and Classification

Vulnerability frameworks are useful tools that draw attention to specific categories of threats. Several frameworks for vulnerabilities exist today. The MITRE Corporation, developed and maintains a database of common vulnerability and exposures (CVEs) to keep track of known software vulnerabilities. Each CVE has an associated risk score created by First.org called the CVSS. The CVSS base score is calculated using a complex formula that is primarily a function of an exploitability score and impact score. NIST's National Vulnerability Database (NVD) cites each score (CVSS, impact, and exploitability) alongside each CVE. Findings by Allodi and Massacci [2] and Nayak *et al.* [3] indicated that existing security research metrics such as CVSS, exploitability, and impact scores for vulnerability are not an indication of exploit for software. Previous studies focused on software vulnerabilities without considering if there are certain subclasses of software where vulnerability risk metrics actually are effective at indicating exploitability. SCADA as a subclass of software should be investigated to understand the vulnerability metrics' relationship with exploits.

Along with their database of CVEs, MITRE created a database of common weakness enumeration (CWEs) [18]. CWEs classify CVEs by type of vulnerability resulting in a standardized and comprehensive list of cyber weakness classes. While CWEs provide a common language for how to define a vulnerability, it does not provide guidance for which CWEs are most relevant for certain classes of software like SCADA systems which would be relevant to urban critical infrastructure. From 2009 to 2011, MITRE and the SANS Institute created a prioritized list of CWEs called the CWE/SANS top 25 most dangerous software errors. The list aimed to identify the greatest software vulnerability types; however, it was nonspecific to a given class of software. The top 25 list used the common weakness scoring system (CWSS) which evaluates vulnerabilities by assessing three metric groups: "base finding metric group (captures the inherent risk of the weakness, confidence in the accuracy of the finding, and strength of controls), attack surface metric group (assesses the barriers that an attacker must overcome in order to exploit the weakness), and the environmental metric group (evaluates the characteristics of the weakness that are specific to a particular environment or operational context)" [19]. The principal weakness of the CWE/SANS prioritized list is that it fails to consider empirical evidence of exploits. A statistical prioritization would be more effective than a scoring prioritization such as CWE/SANS top 25 because a data-driven study can account for the prevalence of exploits found in the wild.

Typologies and taxonomies of critical infrastructure attack and vulnerability exist [6]. Two previous studies on critical infrastructure vulnerabilities focus on different domains: 1) Pak [6] focused on software attacks and 2) Grubesic and Matisziw [5] focused on nonsoftware vulnerabilities. These typologies are very useful to understand the broad critical infrastructure landscape, but fall short as insightful resources for security professionals and researchers because neither are specific enough to provide actionable insight to managers, administrators or policy makers. Also, neither specifically analyze SCADA system security which is essential to city-sustaining systems.

Pak [6] listed types of general attacks he believes are most relevant to CI such as distributed denial of service attacks, worms, and Trojan horses. Pak [6] also made high-level organizational recommendations including strengthening information sharing practices among vulnerable CI sectors, publicly announcing vulnerabilities to ensure patching, and encouraging public/private collaboration to enhance security posture through training and education programs [6]. Further, he encourages continuous monitoring for open ports susceptible to attacks [6]. Pak's [6] recommendations lack specificity due to the breadth of cyber systems included in the standard critical infrastructure definition that includes industries as diverse as the financial and energy sectors. Therefore, security professionals are unable to leverage this research to further fortify their infrastructure.

Grubesic and Matisziw [5] addressed critical infrastructure vulnerability but do not discuss software vulnerabilities. They proposed the following variables are essential to understanding

CI vulnerability: condition and decay, capacity and use, obsolescence, interdependencies, location and network topology, disruptive threats, policy and political environment, and safeguards [5]. While their vulnerability typology is applicable for CI SCADA systems, their omission of software vulnerabilities deprives OT security engineers of concrete and actionable recommendations.

A cyberattack taxonomy was developed by Zhu *et al.* [7] for SCADA systems. Zhu *et al.*'s [7] provided recommendations for control engineers such as: beware of false data injection, man-in-the-middle, and denial of service attacks. In addition to describing types of attacks control engineers should be cognizant of, Zhu *et al.* [7] provided specific guidance in terms of hardware and software vulnerabilities for SCADA systems. The vulnerabilities they determined to be most critical for SCADA include: lack of privilege separation in embedded operating systems, buffer overflow, and SQL injection [7]. While these are concrete vulnerabilities that control engineers can seek out to secure across SCADA systems, it is unclear from Zhu *et al.* [7] analysis how they determined these attacks and vulnerabilities were most important for SCADA. The vulnerability list is supported by some examples of SCADA systems that have these vulnerabilities but there is no data-driven evidence that these are the predominant risks for this class of ICS.

Based on existing literature, there is a need to understand the similarities and differences between SCADA and non-SCADA vulnerabilities and exploits. Also, the relationship between First.org's vulnerability risk metrics and the prevalence of exploits for the software subclass of SCADA systems should be investigated. Further, a data-driven vulnerability prioritization schema for SCADA that is customizable based on an organization's business parameters is needed to complement NIST's complex ICS cybersecurity framework.

II. Our Contribution

In this paper, we reaffirm other scholarly findings that the CVSS risk metrics are not correlated with exploits for all software vulnerabilities; however, unlike our research colleagues we discover that CVSS risk metrics associated with the software subclass of SCADA systems are strongly correlated with exploit. We demonstrate that certain risk metrics are stronger indicators than others in evaluating the likelihood of exploits for SCADA systems. These metrics are used to generate a customizable prioritization schema for SCADA vulnerabilities. A schema can provide a focal point for security researchers to develop SCADA-specific solutions for the most critical vulnerabilities that extends beyond patching. Patching is not always feasible in the SCADA/IIoT environment because these systems must be running at all times and there is little guidance from SCADA vendors on the effect a patch might have on a SCADA system [20], [21]. The vulnerability prioritization schema can also complement NIST's cybersecurity framework for understanding ICS risk. Finally, by determining the prioritized exploit risk, we can make targeted SCADA IIoT software development recommendations for mitigating the associated vulnerabilities.

A. Experimental Findings

To evaluate the landscape of vulnerabilities, a database was collated from the DHS' ICS Computer Emergency Response Team (ICS-CERT) and the MITRE Corporation's CVE systems. The 828 SCADA-relevant CVEs were found across the databases after accounting for duplicates and entries with insufficient information. These CVEs were then classified by their categorical vulnerability type called CWE which is published by MITRE. This categorization enabled the calculation of a SCADA CWE density which provides insight into the distribution of SCADA vulnerabilities across various CWEs. Risk metrics from NIST's NVD were collected for each CVE based on First.org's rating methodology. The average risk score across all CVEs in a given CWE were then calculated, which provided average risk metrics for each vulnerability type. Exploits were then Web-scraped from ExploitDB [22], CVEDetails [23], and the Metasploit [24] code database yielding 52 exploits across 44 SCADA-related CVEs. These exploits were then categorized by their associated CWE, which allowed for the calculation of an exploit density per vulnerability type (CWE).

A cosine similarity test was run on SCADA versus non-SCADA data to understand if there are differences in the distribution of vulnerabilities and exploits across the systems. The distribution of CWEs for SCADA and non-SCADA were found to be the same. However, the distribution of types of vulnerabilities exploited were shown to be different despite having similar vulnerability profiles. This indicates the importance of the exploit density metric for SCADA CWEs.

Multivariate regression models were then run to evaluate the relationship between various SCADA risk metrics and exploit density. An R2 value of 0.924, which is indicative of a strong correlation was found. The independent variables regressed against the dependent variable, exploit density included: CVE density (number of CVE's per CWE), average impact score per CWE, and average exploitability score per CWE.

These variables were then used to develop the SCADA prioritization schema. The top CWEs by vulnerability density, exploit density, exploitability score, and impact score were assessed and combined to generate the prioritization schema.

In summary, we make the following contributions in this paper.

1) SCADA is a unique software subclass with unique attack targets. We statistically validate that exploits for SCADA systems focus on penetrating a specific set of vulnerabilities as compared to non-SCADA systems.
2) First.org's CVSS risk metrics can be used to determine the risk of exploit for the software subclass of SCADA systems. Previously, studies concluded in blanket statements that First.org's exploitability and impact scores were not indicative of exploit risk. This finding provides grounds for substantial further work to evaluate the correlation of exploit and CVSS scores for other software subclasses.
3) SCADA vulnerabilities can be prioritized by data-driven risk metrics in a customizable schema. This has two benefits. First, security researchers could use this schema

TABLE I
ICS-CERT VERSUS MITRE SCADA VULNERABILITIES

	ICS-CERT	MITRE
Number of SCADA CVEs	293	854
Number of Missing SCADA CVEs	592	31
Total SCADA CVEs	885	885

to understand the greatest SCADA vulnerability risk and orient their research to addressing these vulnerabilities. Second, a customizable schema provides flexibility to organizations and IIoT operators to adjust the vulnerability prioritization based on business parameters. Additional variables can be incorporated to the schema or weights can be applied to tailor the prioritization to a given organization.

4) SCADA IIoT system developers can use the prioritization schema to easily identify the principal vulnerabilities based on exploit risk from this paper and take measures to design systems without these vulnerabilities in the future. We offer technical design recommendations for SCADA IIoT system software developers to mitigate the primary exploit risks we identify. Inherently accounting for these vulnerabilities during SCADA system design will dramatically reduce the potential attack surface for IIoT urban critical infrastructure operations.

III. METHODOLOGY

A. Data Collection

Data was first captured on vulnerabilities specific to SCADA systems. Data was collected from publicly available sources including ICS-CERT, MITRE's CVE and CWE database, and NIST's NVD. The intention was not only to collate the specific vulnerabilities for SCADA, but also metadata about these vulnerabilities. The types of information collected included: CVE name and number, associated CWE for each CVE, the CVSS base score for each CVE, the impact score for each CVE, and the exploitability score of each CVE. SCADA vulnerabilities were determined based on keywords in the description of each vulnerability across the databases. Keywords used included "SCADA" and "Supervisory Control and Data Acquisition." Other variations of these keywords were also used to capture potential misspellings.

There was an interesting discrepancy between ICS-CERT's SCADA vulnerabilities cited and MITRE's SCADA-related CVEs. As represented in Table I, ICS-CERT was missing 592 SCADA CVEs that were present in MITRE's database where MITRE was missing 31 SCADA CVEs that were listed in ICS-CERT. This discrepancy could represent a lag between updating the two databases considering vulnerabilities are found more quickly than the database can be updated [25]. However, it could also represent the lack of integration between the two databases as they are independently curated. For purposes of this paper, a master list of SCADA CVEs was created by combining the two databases and removing overlapping SCADA CVEs.

Throughout the course of data collection, other data irregularities were also discovered. Some of the CVEs for SCADA in the MITRE database failed to have CWEs associated with them. This could be due to the CVE being a nonclassified vulnerability type. As recently as CWE version 2.8 (as of May 2016 version 2.9 was released), man-in-the-middle vulnerabilities were not a classified CWE, yet 2.9 has been updated to include this CWE. The CWE list is an ongoing project and the absence of some CWEs are likely a function of this. For consistency of the dataset, all CVEs that lacked a CWE were not included in the analysis. While this could skew the results of the research and guide operators toward a specific CWE without accounting for non-CWE-classified vulnerabilities, there is an underlying assumption made that if a CWE does not exist for a class of CVEs, it is not a popular vulnerability. This assumption was further supported by only 57 out of the 885 SCADA vulnerabilities did not have associated CWEs. Further manual analysis of the CVEs without CWEs confirmed that the CVEs were not all typologically related thereby dismissing the possibility that a major type of future CWE is missing.

After cleaning the data set and reconciling the discrepancies across the ICS-CERT and MITRE vulnerability databases, the master list contained 828 SCADA-related vulnerabilities.

After collecting all SCADA vulnerability data available, a similar process was conducted on non-SCADA vulnerabilities. The intention of collecting non-SCADA data is to evaluate the differences and similarities between SCADA prioritization schema and non-SCADA prioritization schema. Considering the thousands of documented non-SCADA vulnerabilities, a random sample was selected from the MITRE CVE database (excluding all SCADA-CVEs). The random sample contained an equal number of vulnerabilities to those in the SCADA master vulnerability list. Similar to the SCADA list, CVEs with missing metadata were removed from the dataset to preserve consistency.

Once the master list of vulnerabilities was created, a similar list of exploits for the vulnerabilities was developed. A Web-scraper was developed to capture relevant exploits associated with each vulnerability. The Web-scraper pulled data from ExploitDB, CVEDetails and the Metasploit code database. The intent of the collection was to search for all publicly available exploits that corresponded to the relevant CVEs on the master list (both SCADA and non-SCADA). While some CVEs did not have any publicly available exploits associated with them, others had multiple. In total, for the master CVE list, 44 SCADA CVEs were discovered to have 52 associated exploits (some CVEs had more than one exploit) and 103 total non-SCADA CVEs were found to have exploits.

It is important to note that an inherent limitation of the research is the availability of publicly available information on both vulnerabilities and exploits. Similarly to how MITRE contained vulnerabilities that ICS-CERT did not and vice versa, there are likely other sources of vulnerabilities for SCADA systems that were not captured. The same is true of exploits, the Web-scraper only pulled from a finite source of exploits. Exploits that appear on forums or on Github were not captured as part of this data collection process. Future work should include expanding the search for available exploits relevant to SCADA CVEs.

TABLE II
TOP SCADA CWES BY DENSITY

Rank	CWE	Density
1	119: Buffer Overflow	0.244
2	200: Information Exposure	0.105
3	20: Improper Input Validation	0.100
4	79: Cross-Site Scripting	0.063
5	2: Path Traversal	0.062

TABLE III
TOP SCADA CWES BY EXPLOIT DENSITY

Rank	CWE	Density
1	119: Buffer Overflow	0.615
2	200: Path Traversal	0.115
3	20: Improper Input Validation	0.058
4	79: Permissions, Privileges, and Access Controls	0.039
5	22: Code Injection	0.039

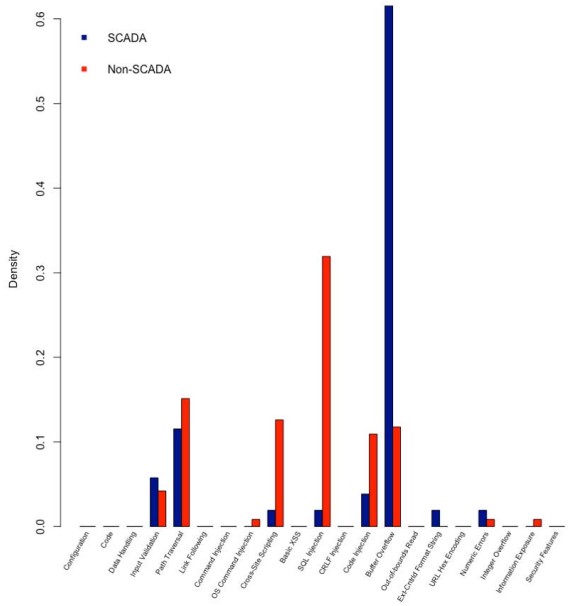


Fig. 1. SCADA versus non-SCADA vulnerability density.

B. Analysis

For purposes of this paper, vulnerability analysis was rolled up to the CWE level. First, the vulnerability density of each CWE was calculated. This was done by dividing the total number of CVEs per CWE by the total number of vulnerabilities. For example, there were 202 CVEs in the CWE "buffer overflow." This was divided by the total number of SCADA vulnerabilities, 828, to determine the CWE density of 24.40%. The density of SCADA CWEs are an indicator of how often these vulnerability types will be found in SCADA critical infrastructure and is important to establishing a prioritization schema. The top five CWEs by density are listed in Table II.

While one class of CWE may have the highest density across a system type, it does not necessarily mean that there are exploits associated with these CWEs. Because of this, CWE density may not be what matters most to SCADA operators and security personnel. The density of CWE exploits could provide a better assessment of operational risk considering the exploits are readily available for use by attackers. The same formula was applied to the exploits per CWE. For example, there were 32 exploits associated with CVEs in the CWE "out-of-bounds read." This was divided by the total number of SCADA exploits, 52, to arrive at the exploit density for buffer overflow to be 61.54%. The top five CWEs for exploit density are listed in Table III.

An important observation is that CWE-200: information exposure is not listed under the top five CWEs for exploit density. This is likely because of the nature of the CWE. Information exposure is the act of an operator providing credentials to an unauthorized actor. It is a managerial exploit rather than a technical one that can be found in a public database, hence the reason it is not covered under top CWEs for exploit density. Because of this, CWE-200 should still be considered a main concern for SCADA systems.

To provide insight for security professionals into SCADA-specific risks, a comparison was made to non-SCADA vulnerability types and their associated exploits. The intention is not to prove that SCADA is entirely different from IT system security, but to inform operators of nuances of SCADA systems.

Based on a side-by-side analysis of the density of CWEs, it is clear that SCADA security professionals should be looking for Buffer Overflow vulnerabilities, compared with non-SCADA which is dominated by Cross Site Scripting. Fig. 1 illustrates these vulnerability density's comparing SCADA and non-SCADA.

A comparison of SCADA versus non-SCADA CWE exploit density reveals that SCADA operators should be most concerned with Buffer Overflow vulnerabilities (as they have the greatest risk of having exploits associated with them). This can be compared to non-SCADA systems where it seems that the predominant CWE to have an exploit associated with it is SQL Injection.

The significance of these SCADA versus non-SCADA differences were evaluated by applying a cosine similarity test on the Web-scraped data. Cosine similarity measures how similar two nonzero vectors are to each other. The closer the cosine similarity value is to 1 indicates a 0° separation between the two vectors (meaning the data sets are very similar). If the cosine similarity is closer to 0, it indicates that there is a 90° separation between the two vectors indicating the data sets are polarized. For purposes of this paper, we will set a threshold of a cosine similarity of greater than 0.5 (indicating a vector angle of 45° or less) is considered to be "similar" data sets and less than 0.5 as dissimilar data sets.

The cosine similarity of the vulnerability density of SCADA compared with non-SCADA was 0.860. This indicates that the overall distribution of the vulnerability types of SCADA versus non-SCADA are very similar and differences are not significant. However, the cosine similarity of the exploit density per CWE of SCADA compared with non-SCADA was 0.408. Considering the threshold set, we can affirm that the exploit landscape is different for SCADA versus non-SCADA

TABLE IV
NON-SCADA EXPLOITS VERSUS CWE FREQUENCY, CVSS, IMPACT SCORE, AND EXPLOITABILITY SCORE

	Estimate	Std. Error	t-value	$\Pr(>\|t\|)$
Intercept	-138.387	144.248	-0.959	0.365
Frequency	0.122	0.085	1.455	0.184
CVSS Score	-50.304	76.508	-0.658	0.529
Impact Score	37.497	54.048	0.694	0.507
Exploitability Score	28.329	36.668	0.773	0.462
Adj. R^2	0.098		p-value	0.340

TABLE V
SCADA EXPLOITS VERSUS CWE FREQUENCY, IMPACT SCORE, AND EXPLOITABILITY SCORE

	Estimate	Std. Error	t-value	$\Pr(>\|t\|)$
Intercept	-22.490	7.225	-3.113	0.014
Frequency	0.167	0.015	11.424	3.12E-6
Impact Score	0.642	0.565	1.137	0.288
Exploitability Score	1.717	1.033	1.661	0.135
Adj. R^2	0.924		p-value	2.27E-5

in a significant way. This significance magnifies the importance of the CWE exploit density's role in SCADA-specific prioritization. This shows that despite consistent vulnerability distributions across SCADA and non-SCADA systems, attackers choose to create exploits for distinctly different vulnerabilities for SCADA systems compared to the exploits they create for non-SCADA systems.

In addition to understanding the value of vulnerability and exploit density, the importance of CVSS, impact score, and exploitability score to evaluating risk was sought for SCADA systems considering Allodi and Massacci [2] determined these scores were not strong indicators of exploit for IT systems. To do this, regression analyses were performed on these variables to determine the likelihood that an exploit exists for a given CWE.

Before investigating the SCADA relationship of exploit density and the First.org risk scores, Allodi and Massacci's [2] findings were verified by regressing the number of non-SCADA exploits with non-SCADA CWE frequency, CVSS, exploitability, and impact scores. Non-SCADA scores by First.org were indeed found to have no correlation with exploit density with an adjusted R^2 value of 0.098. The results of the test can be found in Table IV.

Moving forward to understand SCADA's relationship with these scores, a test was then performed to understand the relationship between number of SCADA exploits and the SCADA CVSS scores. The hypothesis was that the higher the average CVSS score was for a set of CVEs in a CWE, the more likely there would be exploits associated with the CWE. As a reminder, CVSS scores are metrics of risk evaluated based on factors including impact and exploitability scores for a CVE. However, the CVSS score is not an average or sum of impact and exploitability scores. First.org provides the equations for calculating the seemingly complex CVSS scores on their website and it is replicated on NIST's NVD [26].

When conducting a linear regression of CVSS scores on exploits, it was surprising to find no correlation between CVSS scores and exploits with an adjusted R^2 value of -0.074. This indicated that in our SCADA prioritization schema, CVSS scores should not be a factor in determining which CWEs should be prioritized.

Next, a regression was run to determine if the number of vulnerabilities per CWE, the average impact score for CVEs related to a respective CWE and the average exploitability score for CVEs related to a respective CWE were correlated with a CWE having exploits. Similar to the assumption with the CVSS scores' relationship with the presence of exploits, the hypothesis was that a high number of vulnerabilities and high impact and exploitability scores were correlated with the existence of an exploit for a given CWE. In this case, the multiple regression model corroborated the hypothesis with an adjusted R^2 value of 0.924 showing a strong relationship between the presence of an exploit and the number of vulnerabilities for the given CWE, the average impact score and exploitability score. The results of the analysis can be found in Table V. These results were surprising as they indicate that there is something unique about SCADA CWE frequency and exploitability and impact scores' relationship with exploit density that is not true of IT systems as found by Allodi and Massacci [2]. Further, this indicates that in First.org's complex equation that converts impact and exploitability scores to CVSS scores, the correlation with the presence of an exploit for a given CWE is lost. This could suggest that the CVSS score is a flawed indicator of risk whereas the exploitability and impact scores are not (assuming risk can be accessed via the presence of an exploit as per the suggestion of this paper).

To further validate the assertion that CVSS scores do not correlate with the presence of an exploit, other multiple regressions were run regressing exploits on variations of CVSS scores and other variables. All of these regressions consistently showed a weak relationship between exploits and CVSS scores, even when coupling CVSS scores with exploitability and impact scores.

Based on this analysis, the magnitude of exploitability and impact scores for a given CWE are important. The top ten CWEs for impact and exploitability scores can be found in rank order in Table VI. It is interesting to note that while the top ten CWEs for impact and exploitability are not the same rank, all top impact score CWEs are also found in the top exploitability score CWE list and vice versa.

C. Scoring

To develop a SCADA prioritization schema, the above analysis was used to evaluate which variables are most relevant to determining the SCADA IIoT risk. The variables of CWE density, CWE exploit density, and impact and exploitability scores were ultimately used. Additional variables can be included for a prioritization schema if data is available and the data is found to correlate with exploit density. While there are many

TABLE VI
TOP TEN SCADA CWES BY IMPACT AND EXPLOITABILITY SCORES

Rank	CWE by Impact Score	CWE by Exploitability Score
1	119: Buffer Overflow	119: Buffer Overflow
2	20: Improper Input Validation	20: Improper Input Validation
3	264: Permissions, Privileges, and Access Controls	200: Information Exposure
4	200: Information Exposure	22: Path Traversal
5	22: Path Traversal	79: Cross-Site Scripting
6	255: Credentials Management	264: Permissions, Privileges, and Access Controls
7	399: Resource Management Errors	399: Resource Management Errors
8	287: Improper Authentication	255: Credentials Management
9	310: Cryptographic Issues	287: Improper Authentication
10	79: Cross-Site Scripting	310: Cryptographic Issues

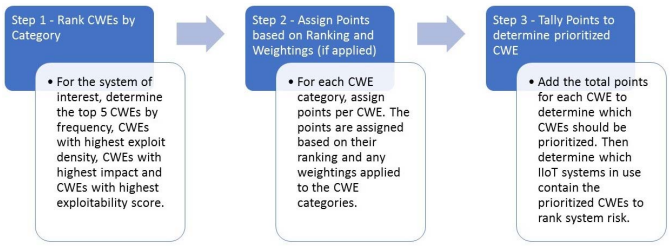


Fig. 2. Prioritization schema steps.

options to determine how to score each variable for the prioritization order, for purposes of this paper, a rudimentary system was selected intentionally for transparency. More sophisticated weight-based prioritization schemes can be created and customized for various organizations. The purpose of this paper is not necessarily to generate the "correct" or ultimate prioritization order for SCADA system vulnerabilities, rather it is to establish a framework for how a data-driven study can be used to develop customized SCADA risk prioritization schemes. Future work is encouraged to address how to weight each variable for the prioritization schema.

Point values were assigned based on the ranked position of the CWE in each category. Each category (i.e., CWE density, CWE exploit density, etc.) were weighted equally. For purposes of this analysis, the top five CWEs from each category were ranked where the top ranked CWE receives a point value of 5 and the fifth CWE in the ranking receives a value of 1.

The top five ranked CWEs can be found for all four categories in Table VII and the total allocated points per CWE can be found in Table VIII. Fig. 2 represents the steps required to generate the prioritization schema including the inputs and outputs of the model.

This prioritization schema for SCADA vulnerabilities logically makes sense based on the characteristics of SCADA operations. A closer look at the top three prioritized SCADA vulnerability types helps illustrate this. Buffer overflows are defined as a vulnerability where software can read or write to a memory location that is outside the intended boundary of the

TABLE VII
TOP FIVE RANKED CWES PER CATEGORY

Rank	CWE by Frequency	CWE by Exploit Density	CWE by Impact Score	CWE by Exploitability Score
1 (5 points)	119: Buffer Overflow	119: Buffer Overflow	119: Buffer Overflow	119: Buffer Overflow
2 (4 points)	200: Information Exposure	22: Path Traversal	20: Improper Input Validation	20: Improper Input Validation
3 (3 points)	20: Improper Input Validation	20: Improper Input Validation	264: Permissions, Privileges, and Access Controls	200: Information Exposure
4 (2 points)	79: Cross-Site Scripting	264: Permissions, Privileges, and Access Controls	200: Information Exposure	22: Path Traversal
5 (1 point)	22: Path Traversal	94: Code Injection	22: Path Traversal	79: Cross-Site Scripting

TABLE VIII
TOTAL SCORES FOR TOP-RANKED CWES

CWE	Total Points
Buffer Overflow	20
Improper Input Validation	14
Information Exposure	9
Path Traversal	8
Permissions, Privileges, and Access Controls	5
Cross-Site Scripting	3
Code Injection	1

memory buffer. It is not surprising that buffer overflows warrant the highest priority for SCADA vulnerabilities as buffer overflows are inherent in older, low-level programming languages such as C which is common to SCADA. Further, SCADA devices are rarely rebooted due to their constant operating requirements. Systems that have not been rebooted for years will accumulate memory fragmentation. This makes devices substantially more vulnerable to buffer overflow vulnerabilities [7]. Improper input validation is when software does not check input which enables an attacker to enter values that could cause control flow changes that are not expected by an operator. Considering one of the key differentiators of ICS versus IT systems is that ICSs are deterministic, this vulnerability is clearly a threat [9]. SCADA systems require low jitter and any disruption of the deterministic processes such as an attack exploiting the vulnerability class of improper input validation would severely impact operations. Finally, information exposure is the disclosure of information to an unauthorized person. This vulnerability type is also logical

for SCADA considering the prevalence of default usernames and passwords used across systems [7]. Because default usernames and passwords are frequently used, attackers can easily obtain this information from an instruction manual or from a vendor discussion forum. Also, information exposure as a prioritized exploit is logical considering the prevalence of phishing attacks used to collect credentials from critical infrastructure operators. This was seen for the Ukrainian electric grid cyberattack and UglyGorilla's cyber espionage program against 23 U.S. natural gas pipelines [1], [27].

While information exposure is a borderline priority with path traversal, it is important to remember that information exposure lacked technical exploits publicly available in the databases searched because it is more of a managerial exploit than technical. Therefore, it was not appropriately captured in the exploit density data set, and indeed belongs at the top of the list.

IV. Research Implications

A. Operator Implications

This paper, while niche to a subsector of IIoT, can have considerable impact for urban critical infrastructure security. Our findings indicate that there is a strong relationship between First.org risk metrics and exploit density, specifically for SCADA systems. There are three groups of critical urban infrastructure security experts that can benefit from this insight chief information security officers (CISOs), security operations center (SOC) analysts, and system architects.

CISOs who oversee all security operations of an organization generally have the difficult responsibility to develop and manage programs to secure the organization at scale. Because of our findings, CISOs can streamline their programs for securing SCADA systems. Rather than establishing programs meant to help create metrics that can be used to assess the risk of various IIoT systems, CISOs could instead refer to First.org's metrics of exploitability and impact to evaluate IIoT risk of exploit. There is no longer a need to start from scratch developing metrics considering we demonstrated that exploitability and impact metrics are valid predictors of exploit risk for SCADA systems.

SOC analysts are another group of security experts that can benefit from our findings. SOC analysts are often responsible for monitoring and fixing security risks as they occur. Instead of reactively seeking out security threats to address, our risk prioritization schema will help analysts proactively seek out which IIoT systems are likely to be attacked. SOC analysts can cross-check IIoT devices with CVEs and CWEs that we identified to be most exploited to arrive at their prioritized device list.

System architects responsible for selecting components for urban critical infrastructure should use our findings to carefully select systems based on their vulnerability profile. While we acknowledge most urban critical infrastructure IIoT consists of legacy devices that are not often replaced, when new devices are procured, our risk prioritization schema can be used to assess which SCADA systems should be installed. IIoT devices with the most vulnerabilities in the categories we discover to be of highest risk of exploit should be avoided.

B. Technical Design Implications

Future SCADA IIoT systems should be designed and developed with the intent to "design out" the prioritized vulnerabilities indicated in this paper. Addressing the prioritized vulnerabilities in the design phase could help reduce the number of future attacks against this class of IIoT. Based on recommendations of the top three prioritized vulnerabilities of buffer overflows, improper input validation, and information exposure, we can propose technical design strategies to help avoid these vulnerabilities.

Buffer overflows are prevalent in operating environments that are programmed in C. The language provides direct memory access, which can be used to help reduce the device's energy consumption. Energy efficiency is important for the cost efficiency of SCADA systems especially considering their highly distributed nature in locations where resource availability might be limited. Further, C can be very memory efficient, which is also valuable for small devices required for urban critical infrastructure. Despite these benefits of C, the buffer overflow vulnerabilities that result from coding mistakes are a considerable downside. This prioritized vulnerability can be "designed out" by using a memory safe programming language when developing future SCADA systems. One memory safe language that is also memory efficient is Rust [28]. If future IIoT systems can be programmed in Rust, buffer overflows will no longer be an issue therefore removing this attack vector for IIoT SCADA systems.

SCADA design traditionally focuses on detecting and classifying control conditions that enables accurate monitoring in various states [29]. With focus on the functional operation of the SCADA system, proper input to the system is assumed and not accounted for in the design process. With increased skepticism of IIoT device inputs based on recent attacks, and the associated vulnerabilities involving improper input, SCADA designers must take measures to validate input. Design recommendations that could reduce the number of improper input validation vulnerabilities in systems include using an input validation framework such as Struts or OWASP ESAPI Validation API when creating the system or by identifying all possible areas where an attacker could input data and employ a whitelist strategy [30]. Frameworks like Struts help to guide software development so that there are few validation issues. A whitelisting strategy entails rejecting all inputs other than the few that are actually appropriate for the design specifications of the system's purpose. The whitelist should account for all input properties ranging from length to syntax.

Information exposure may perhaps be the most challenging vulnerability to design out of a SCADA system. This is because many information exposure attacks happen as a function of the human element either by error or intentionally. A potentially effective mechanism to mitigate the damage caused by information exposure is to compartmentalize data systems [31]. Designing SCADA IIoT to be compartmentalized can limit the data leak or attack to only the compartment

that was breached. If a centralized data store for SCADA IIoT is used, compromised access to the central hub will leave all data vulnerable. These proposed SCADA IIoT technical design strategies may help to reduce the prevalence and risk of the top vulnerabilities identified in this paper. Each SCADA designer will need to evaluate if these strategies can be used based on their specific technology requirements as not all design mitigation techniques will necessarily be appropriate for every IIoT system.

V. Conclusion

Unique contributions of this paper are significant for security researchers investigating SCADA systems, SCADA IIoT designers and critical infrastructure operators working with IIoT. The research reveals that SCADA systems as a software subclass were found to have exploits that target a distinct set of vulnerabilities compared with non-SCADA systems. This indicates that the risk profile for SCADA systems varies compared with that of non-SCADA. The study also identifies highly correlated relationships between First.org vulnerability risk metrics and the density of SCADA exploits. These findings could encourage security researchers to reconsider their assertions that exploitability and impact scores are inaccurate predictors for the risk of exploit. Researchers should repeat these studies on risk metrics' relationship with exploits specifically for subsets of software as was done for SCADA. Finally, findings suggest that security researchers, SCADA IIoT designers and SCADA operators should focus on a core set of vulnerability types for SCADA systems. Considering the unique requirements of SCADA systems and the associated challenges with vulnerability patching, alternative security strategies concerning prioritized vulnerabilities should be investigated. The prioritization framework provided can be customized based on organizational requirements and parameters. Urban critical infrastructure operators can use the prioritization in parallel with NIST's more comprehensive cybersecurity framework to understand their SCADA risk.

Because the SCADA prioritization schema is based on empirical, data-driven findings, it will need to be updated continuously as new exploits are published. If a series of new SCADA exploits are released that target a specific vulnerability class, the prioritization schema will be outdated. It is recommended that this prioritization is updated annually as was the CWE/SANS top 25 list.

There are several future research opportunities related to this paper. CVSS and exploitability and impact scores are being transitioned from version 2 to version 3 which entails new scores that are more specific. Once this new scoring methodology has been completed and vetted for accuracy, this paper should be repeated with updated data so that the exploitability and impact scores can be normalized appropriately. Testing additional characteristics of vulnerabilities as variables to determine their association with the risk of exploit could be included in future work. As previously indicated, other sources of exploits can be compiled from repositories such as Github or sources that may reference managerial related exploits rather than technical ones to better capture the exploit potential of CWEs such as information exposure. Future research could also investigate the scoring mechanisms used for the prioritization schema, which can be further customized through weightings and new point allocation systems. Finally, further studies should investigate opportunities to incorporate this SCADA prioritization approach to the existing NIST framework to provide a data-driven approach to evaluating system risk. This should accompany IIoT security policy research intended to encourage a robust, quantitative approach for evaluating urban critical infrastructure risk.

Acknowledgment

The authors would like to thank A. Sanchez, S. Madnick, L. Susskind, D. Serpanos, A. Kam, H. Okhravi, A. Viswanathan, R. Oppliger, V. Roth, L. Uzeda, and R. Yahalom for ideas, edits, and contributions to this paper.

References

[1] "Analysis of the cyber attack on the Ukrainian power grid, defense use case," Electricity Inf. Sharing Anal. Center, Washington, DC, USA, Rep., 2016.

[2] L. Allodi and F. Massacci, "A preliminary analysis of vulnerability scores for attacks in wild: The ekits and sym datasets," in *Proc. ACM Workshop Build. Anal. Datasets Gather. Exp. Returns Security (BADGERS)*, Raleigh, NC, USA, 2012, pp. 17–24. [Online]. Available: http://doi.acm.org/10.1145/2382416.2382427

[3] K. Nayak, D. Marino, P. Efstathopoulos, and T. Dumitraş, "Some vulnerabilities are different than others," in *Research in Attacks, Intrusions and Defenses*, A. Stavrou, H. Bos, and G. Portokalidis, Eds. Cham, Switzerland: Springer Int., 2014, pp. 426–446.

[4] Dimensional Research. (Mar. 2016). *Dimensional Research. Trends in Security Framework Adoption: A Survey of it and Security Professionals*. [Online]. Available: https://static.tenable.com/marketing/tenable-csf-report.pdf

[5] T. H. Grubesic and T. C. Matisziw, "A typological framework for categorizing infrastructure vulnerability," *Geo J.*, vol. 78, no. 2, pp. 287–301, 2013.

[6] C. Pak, *Typologies of Attacks and Vulnerabilities Related to the National Critical Infrastructure*. London U.K.: Palgrave Macmillan, 2015, pp. 169–180, doi: 10.1057/9781137455550_11.

[7] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on SCADA systems," in *Proc. 4th Int. Conf. Internet Things Int. Conf. Cyber Phys. Soc. Comput.*, Dalian, China, Oct. 2011, pp. 380–388.

[8] J. Slay and M. Miller, "Lessons learned from the Maroochy water breach," in *Critical Infrastructure Protection*, E. Goetz and S. Shenoi, Eds. Boston, MA, USA: Springer, 2008, pp. 73–82.

[9] B. Galloway and G. P. Hancke, "Introduction to industrial control networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 860–880, 2nd Quart., 2013.

[10] J. Weiss, *Protecting Industrial Control Systems from Electronic Threats*. New York, NY, USA: Momentum Press, 2010.

[11] K. A. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," *NIST Special Publ.*, vol. 800, no. 82, p. 16, 2011.

[12] P. A. S. Ralston, J. H. Graham, and J. L. Hieb, "Cyber security risk assessment for SCADA and DCS networks," *ISA Trans.*, vol. 46, no. 4, pp. 583–594, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0019057807000754

[13] B. Obama, *Executive Order 13636: Improving Critical Infrastructure Cybersecurity*, White House, Washington, DC, USA, 2013.

[14] W. Miron and K. Muita, "Cybersecurity capability maturity models for providers of critical infrastructure," *Technol. Innov. Manag. Rev.*, vol. 4, pp. 33–39, Oct. 2014. [Online]. Available: http://timreview.ca/article/837

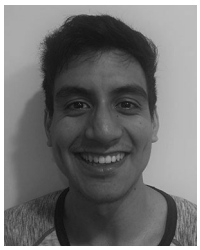
[15] Z. Zhang, "Environmental review & case study: NERC's cybersecurity standards for the electric grid: Fulfilling its reliability day job and moonlighting as a cybersecurity model," *Environ. Pract.*, vol. 13, no. 3, pp. 250–264, 2011. doi: 10.1017/S1466046611000275.

[16] "Mandatory reliability standards for critical infrastructure protection," Federal Energy Regul. Comm., Washington, DC, USA, Rep. RM06-22-008, Jan. 2008.

[17] R. Ellis, "Regulating cybersecurity: Institutional learning or a lesson in futility?" *IEEE Security Privacy*, vol. 12, no. 6, pp. 48–54, Nov./Dec. 2014.
[18] MITRE, DHS. *Common Weakness Enumeration National Vulnerability Database 2016*. Accessed: Oct. 31, 2016. [Online]. Available: https://cwe.mitre.org/index.html
[19] MITRE. *2011 CWE/SANS Top 25 Most Dangerous Software Errors*. Accessed: Nov. 12, 2016. [Online]. Available: http://cwe.mitre.org/top25/
[20] M. Luallen, *Breaches on the Rise in Control Systems: A SANS Survey*, SANS Inst., North Bethesda, MD, USA, Apr. 2014.
[21] A. Sarwate, *Scada Security: Why Is it so Hard?* Blackhat, San Francisco, CA, USA, Nov. 2011.
[22] Electronic Database. *Offensive Security Exploit Database Archive*. Accessed: Oct. 31, 2016. [Online]. Available: https://www.exploit-db.com/
[23] MITRE. *CVE Security Vulnerability Database. Security Vulnerabilities, Exploits, References and More*. Accessed: Feb. 1, 2016. [Online]. Available: https://www.cvedetails.com/
[24] Rapid7. *Exploit Database*. Accessed: Oct. 31, 2016. [Online]. Available: https://www.rapid7.com/db/modules/
[25] MITRE. *CVE–About CVE*. Accessed: Mar. 18, 2016. [Online]. Available: https://cve.mitre.org/about/index.html
[26] FIRST. *Common Vulnerability Scoring System (CVSS-SIG)*. Accessed: Feb. 1, 2016. [Online]. Available: https://www.first.org/cvss/
[27] "APT1 exposing one of china's cyber espionage units," MANDIANT, Alexandria, VA, USA, Rep., 2013.
[28] N. D. Matsakis and F. S. Klock, II, "The Rust language," *Ada Lett.*, vol. 34, no. 3, pp. 103–104, Oct. 2014. [Online]. Available: http://doi.acm.org/10.1145/2692956.2663188
[29] M. Kezunovi, T. Djoki, and T. Kosti, "Automated monitoring and control using new data integration paradigm," in *Proc. 38th Annu. Hawaii Int. Conf. Syst. Sci. (HICSS)*, vol. 2, 2005, p. 66a, doi: 10.1109/HICSS.2005.112.
[30] MITRE. *CWE-20: Improper Input Validation (3.0)—Potential Mitigations*. Accessed: Mar. 10, 2018. [Online]. Available: https://cwe.mitre.org/data/definitions/20.html
[31] MITRE. *CWE-200: Information Exposure (3.0)—Potential Mitigations*. Accessed: Mar. 10, 2018. [Online]. Available: https://cwe.mitre.org/data/definitions/20.html
[32] I. T. Laboratory. *National Vulnerability Database*. Accessed: Apr. 20, 2016. [Online]. Available: https://nvd.nist.gov/

Gregory Falco is currently pursuing the Ph.D. degree in cybersecurity at the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA.

He is a Cyber Research Fellow with the Belfer Center, Harvard University, Cambridge, and an Adjunct Professor with Columbia University, New York, NY, USA. He is the co-founder and the CEO of NeuroMesh, Cambridge, a security company that uses the blockchain to secure Industrial Internet of Things embedded systems.



Carlos Caldera received the Bachelor of Science degree in computer science and electrical engineering and Master's of Engineering degree in computer science and electrical engineering, with a concentration in computer systems from the Massachusetts Institute of Technology, Cambridge, MA, USA.

He is currently a Software Engineer with Oracle, Redwood City, CA, USA.

Howard Shrobe is a Principal Research Scientist with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA, where he is also the Director of CyberSecurity@CSAIL. He has served as the Assistant Director and Chief Scientist of the Information Technology Office at DARPA, Arlington, VA, USA. He has also served as a DARPA Program Manager.