Learning from Vulnerabilities - Categorising, Understanding and Detecting Weaknesses in Industrial Control Systems

Thomas, Richard J.; Chothia, Tom

DOI:
10.1007/978-3-030-64330-0_7

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Thomas, RJ & Chothia, T 2020, Learning from Vulnerabilities - Categorising, Understanding and Detecting Weaknesses in Industrial Control Systems. in S Katsikas, F Cuppens, N Cuppens, C Lambrinoudakis, C Kalloniatis, J Mylopoulos, A Antón, S Gritzalis, W Meng & S Furnell (eds), *Computer Security - ESORICS 2020 International Workshops, CyberICPS, SECPRE, and ADIoT, 2020, Revised Selected Papers.* Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12501 LNCS, Springer, pp. 100-116, 6th International Workshop on Security of Industrial Control Systems and Cyber-Physical Systems, CyberICPS 2020, 2nd International Workshop on Security and Privacy Requirements Engineering, SECPRE 2020, and 3rd International Workshop on Attacks and Defenses for Internet-of-Things, ADIoT 2020, held in conjunction with 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, United Kingdom, 14/09/20. https://doi.org/10.1007/978-3-030-64330-0_7

Learning from Vulnerabilities - Categorising, Understanding and Detecting Weaknesses in Industrial Control Systems

Richard J. Thomas and Tom Chothia

School of Computer Science, University of Birmingham, Birmingham, UK
{R.J.Thomas,T.P.Chothia}@cs.bham.ac.uk

Abstract. Compared to many other areas of cyber security, vulnerabilities in industrial control systems (ICS) can be poorly understood. These systems form part of critical national infrastructure, where asset owners may not understand the security landscape and have potentially incorrect security assumptions for these closed source, operational technology (OT) systems. ICS vulnerability reports give useful information about single vulnerabilities, but there is a lack of guidance telling ICS owners what to look for next, or how to find these. In this paper, we analyse 9 years of ICS Advisory vulnerability announcements and we recategorise the vulnerabilities based on the detection methods and tools that could be used to find these weaknesses. We find that 8 categories are enough to cover 95% of the vulnerabilities in the dataset. This provides a guide for ICS owners to the most likely new vulnerabilities they may find in their systems and the best ways to detect them. We validate our proposed vulnerability categories by analysing a further 6 months of ICS Advisory reports, which shows that our categories continue to dominate the reported weaknesses. We further validate our proposed detection methods by applying them to a range of ICS equipment and finding four new critical security vulnerabilities.

1 Introduction

Industrial Control Systems (ICS) form a key part of the critical national infrastructure and industrial environments. Attacks against ICS devices, such as Stuxnet [9], BlackEnergy [18] and Triton [16], have aim to cause disruption and damage ICS equipment. Many ICS environments were segregated from IT networks, however most now exist on the same, heterogeneous network, opening them up to a wide range of attacks.

In recognition of the importance of ICS cybersecurity, the European Union Network and Information Systems (NIS) Directive came into force in May 2018. Member states are required to define essential services and improve infrastructure security and resilience in identified sectors. The NIS Directive shifted responsibility for assurance onto asset owners, who may lack cybersecurity understanding of what vulnerabilities and issues exist in the ICS space. IT Security

is considered a well-understood problem, with insights available to OT operators. However, there are different assumptions and requirements placed on OT devices, for example operational lifespan measured in the order of decades, not years, and safety which may not exist in IT environments. By reviewing what vulnerabilities exist in the industrial space, we can define what priorities for asset owners and the supply chain should be and how they can be addressed and detected to improve industrial security.

ICS environments are typically made up of Programmable Logic Controllers (PLCs), automating a process given a set of inputs, controlling outputs. PLCs may be connected to sensors, actuators and Human Machine Interfaces (HMIs), operator control panels displaying the state of the system and allow an operator to interface with the process. Other components, such as Supervisory Control and Data Acquisition (SCADA) may be integrated for logging, analytics and control purposes, and Remote Terminal Units (RTUs), enable remote management for devices. The exception of some SCADA systems that run on standard PCs is that these are usually provided to the ICS owner as proprietorial, closed source software running on unidentified hardware. Sometimes, the security assumptions made by the designers of this equipment are not clear, and there is no easy way for ICS owners to inspect and run their own software on the ICS equipment. This makes the security controls and issues for ICS equipment quite different from, e.g., securing desktop machines and servers in a company setting. Therefore, general work on common vulnerability categories and detection methods does not carry over to the ICS domain.

To provide insight into vulnerabilities in ICS environments, this paper carries out a detailed review of nine years of ICS-CERT Advisories and related data. This tells us the kinds of vulnerabilities that commonly occur in ICS environments. We analyse this data, identifying trends and what kind of detection methods could find the vulnerabilities. Based on this analysis, we suggest eight categories for the vulnerabilities based on concrete detection methods. These categories cover 95% of all vulnerabilities in our dataset and give the ICS owner clear steps they can follow to find the weaknesses, and advise vendors on priority areas to resolve. It is important to note that our analysis is purely concerned with the kind of vulnerability in ICS systems that leads to an ICS advisory, i.e., a new flaw in the security of a system. There are many other weaknesses which might be exploited to attack an ICS systems, such as phishing e-mails, or the use of unpatched systems with known vulnerabilities. Most of these issues are well addressed by existing IT security methods and practices. This gives ICS owners an understanding of the types of vulnerabilities that exist in an accessible way, where current information is ambiguous and, as a result, not actionable. Our categories enable an asset owner to act, with appropriate tooling and techniques such that they can be confident in the security of their infrastructures.

To validate the categories and trends we identify, we look at an extra six months of ICS advisory reports. We find that our eight categories continue to dominate the advisories (accounting for 96% of the new vulnerabilities) and each category is well represented. To validate the category detection methods

we apply the automated methods to three PLCs and two HMIs from major ICS manufacturers. As a result we find six new attacks against the ICS equipment four of which would be ranked as critical: two denial of service attacks, an open redirect on a web control panel, and an authentication bypass. Responsible disclosure for these vulnerabilities is ongoing and we will make the information public once ICS advisories have been released.

Our contributions are as follows:

- combining a number of data sources together, we present a detailed analysis of ICS vulnerabilities and trends,
- suggesting new eight categories for classifying ICS vulnerabilities that are based on detection methods,
- validating the trends and categories against 6 months of new vulnerability reports, and our detection methods by applying them to five pieces of ICS equipment finding 4 new critical vulnerabilities.

In Section 2, we outline our process for connecting sources of data, outlining key statistics and priority areas. We go further to define detectable vulnerability classes to test for such vulnerabilities in Section 3 and predict future ICS vulnerabilities and validate these predictions in Section 4, concluding in Section 5.

Related Work: ICS security research is an active field, where most research focuses on vulnerability research in specific devices and implementations [4, 20, 13, 5] which highlight particular flaws and are aimed towards finding new flaws and proposing resolutions to improve collective ICS security. These however do not consider the security of ICS as a whole and what common types of vulnerability exist. On the other hand, a chronology of ICS security incidents provides a thorough analysis of high-profile incidents [12], but some commercially-led research papers [1, 7, 8, 17, 19] have carried out ICS vulnerability analysis and provide highlights of some of the vulnerability categories that exist, but do not consider all vulnerabilities, or show only a few categories. The last assessment report from ICS-CERT highlighting the top weakness categories in ICS was published in 2016, where no authoritative report has since replaced it [14]. More recently, the authors in [10] review ICS vulnerability reports to determine how many resulted from architectural design decisions, but simply state common root causes. In [15], the authors propose a linked and correlated database for ICS vulnerabilities, to support security operations centres, but additionally categorise vulnerabilities into 6 categories, which do not lend themselves towards detecting such vulnerability categories, and do not provide the level of detail required for ICS owners and supply chain to improve their respective security models. Similar work as part of the OpenCTI project[1] has attempted to make vulnerability information accessible, however, it requires significant expert efforts to integrate for ICS vulnerability reports, and again, informs the ICS owner and vendor what has happened, but not how it can be resolved, or detected.

[1] `https://opencti.io`

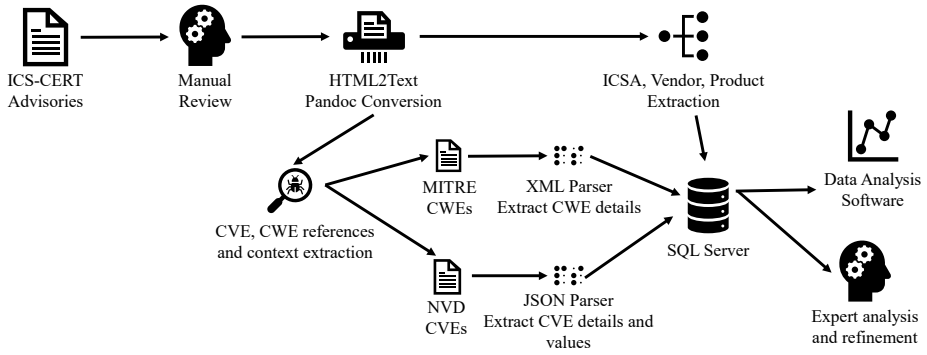

Fig. 1. Parsing and Processing Workflow to create our Dataset. ICS Advisories are parsed, and when a CWE/CVE ID is found, the corresponding record is parsed and the information brought together before being committed into the database.

2 Connecting Sources of Data for Vulnerability Insights

Data Sources and Building the Dataset: ICS vulnerabilities are reported and published in a number of places, for example vendor websites, CVE listings and ICS-CERT. For ICS owners and vendors, it is not clear which source is authoritative and provides the best whole-of-sector coverage. A number of vendors individually publish advisories, however in a survey of a number of common ICS vendors, we found that some required a support contract/approval to gain access to security reports, which limits this coverage if vendors were used as the primary source. Our ICS vulnerability dataset is therefore built up from three sources: ICS-CERT advisories, MITRE and the National Vulnerability Database (NVD). ICS-CERT advisories are the root source of information, where references to MITRE and the NVD are used to extract further information. A workflow which imports these sources is given in Figure 1.

ICS-CERT Advisories: These are published by the USA Cybersecurity and Infrastructure Security Agency (CISA), providing authoritative vulnerability information to the ICS community. To the best of our knowledge, it is the most comprehensive source of ICS vulnerability information. These reports are in HTML format[2], which we convert into plaintext and markdown to flatten all formatting, making it easier to extract reference fields for CWEs and CVE numbers. When we find these we retrieve the corresponding record and import some fields from those sources to provide context to the vulnerability information.

MITRE CWEs: The MITRE Corporation is responsible for two schemes used within our dataset; Common Weakness Enumeration (CWE) and Common Vulnerability and Exposure (CVE). While MITRE provides CVE information in a machine-readable format, it is not as full-featured as the National Vulnerability Database's input to CVE information, for example appraisals of the impact and

[2] An example is available at `https://us-cert.cisa.gov/ics/advisories/ICSA-17-157-01`.

criticality of that vulnerability in addition to further analysis, such as listing affected products.

The CWE or root cause identifier stated in ICS advisories, however, is used in our analysis. These are unique, distinct vulnerability patterns and anti-patterns in software development, which can express the type of vulnerability that exists. One benefit of CWEs is that they can be grouped together to give types of vulnerability, e.g. memory or web vulnerabilities [21].

National Vulnerability Database (NVD) CVEs : The NVD, provided by NIST, takes the CVE information, analyses and assesses the vulnerability. This assessment allocates the CVSS score, used to define the criticality and impact of the vulnerability, how it may be exploited and under what conditions the system was exploitable. NIST CVEs are provided in JSON files, which we parse. When a CVE reference is found, the corresponding CVE record is retrieved, tagged with the CWE referenced in the ICS advisory and is imported.

The Combined Dataset Our dataset[3] is built from 1,114 ICS vulnerability reports, with 283 distinct CWE references, and 2,232 CVEs, collected from 2011, when ICS Advisories started to be published to August 2019 (this cut-off was chosen to allow sufficient new vulnerabilities to be produced, allowing validation of our results). The dataset contains the ICS advisory number, release and update dates, the name of the vendor affected and a short description which includes the product affected. For CWEs found in the ICS advisory, we include the CWE ID, the name of the CWE, a brief description and contextual background details, and the CWE status (e.g. if it has been deprecated). CVEs stated in the ICS advisory include the number, description, base, impact[4] and exploitability[4] scores, CVSS vector, severity, access vector, complexity to exploit, availability, integrity and confidentiality impact, the list of privileges required, the impact on system privileges and whether user interaction is required for the exploit to be successful.

Limitations of Existing Data Sources In isolation, these sources provide little contextual information and means to identify trends and types of vulnerabilities that exist in ICS systems. By connecting ICS advisories to CWE-specific information, we can categorise the type of vulnerabilities that arise in the ICS domain, identify patterns and follow trends. With CVSS, we do not use the assigned scores, where the vector components provide concrete information about the impact of the vulnerability, as the impact scores does not exist in CVSS v2, but all have a defined vector, where 791 of 2,232 CVEs in our dataset do not have these numerical scores. Other fields and content are also not imported, e.g. acknowledgements, researchers or URLs as these are not relevant in our analysis.

Accuracy of the data being used is critical, where a survey of ICS vulnerability data quality showed that in 2018, 32% of ICS CVEs had the wrong CVSS score assigned [7], improving to 19% [8] in 2019. The combination of

[3] Available at `https://github.com/uob-ritics/esorics2020-dataset`

[4] Exists only in CVSS v3

Grouping	Prevalence	Availability	Integrity
CWE/SANS Top 25 (2011)	24%	11%	10%
CWE Weaknesses on the Cusp (2011)	2%	1%	1%
CWE Top 25 (2019)	48%	28%	18%

Table 1. Coverage of ICS Vulnerabilities for existing mappings. For the number of ICS vulnerabilities which have a *'COMPLETE'* or *'HIGH'* impact value.

these sources mitigates the risk, where we use the most current version of the record, rather than its first instance. Out of 1097 ICS Advisories surveyed up to August 2019, 197 had been updated, which may include additional products affected, new vulnerabilities identified, or new mitigations, for example patch availability. Without using the most current information, these additional vulnerabilities may have been overlooked. These, however, generally do not require further analysis, unless some new, unclustered CWEs were introduced.

3 Understanding and Classifying Vulnerabilities

Understanding the Type of ICS Vulnerabilities: In order to categorise all ICS vulnerabilities and define detection strategies, we must first understand the ICS vulnerability landscape. In order to categorise these vulnerabilities, as we explain later in this section, we must consider existing groupings.

MITRE offers a number of groupings of CWEs, for example based on the OWASP Top 10 and the CWE Top 25 Most Dangerous Software Errors. In Table 1, we map our dataset onto existing clusters (groupings) based on prevalence, and the number of CVEs that had a high impact on the integrity or availability of the system. We note that firstly, these mappings are not mutually exclusive; one CWE may exist in more than one SFP cluster. Secondly, these mappings do not capture the majority of the dataset, where the CWE Top 25 leaves over 50% unclassified.

The CWE contains clusters defined around the concept of software fault patterns (SFPs), which contains most of the CWEs specified in ICS advisories with no overlap between clusters. These CWEs are mapped to a specific cluster, e.g. memory access (CWE-890) and cryptography (CWE-903). Out of the 2,232 CVEs in our dataset, 1,801 could be mapped directly to a SFP cluster. For the remaining 431 CVEs, we manually assigned them to a respective cluster based on the stated issue (e.g. buffer overflow or cross-site scripting attack) in the ICS Advisory and CVE description. As an example, Cross-Site Request Forgery attacks (CWE-352) have no mapping, but are web-based attacks which are exploited through malicious input, and thus, we categorise it as 'Tainted Input', where other web-based weaknesses sit. This manual expert analysis ensures that all CWEs are represented within the correct categories rather than 'Other'. The result of this classification is shown in Figure 2, where we group the vulnerability subclasses, highlight the types of vulnerability that exist within the category and rank vulnerabilities based on the number of CVEs with a 'high' or 'complete' (critical) CVSS availability and integrity impact. The objective of introducing a 'Zone 1' and 'Zone 2' is to highlight priority areas, where it is expected that, as these vulnerability classes are investigated and resolved, the next class can be

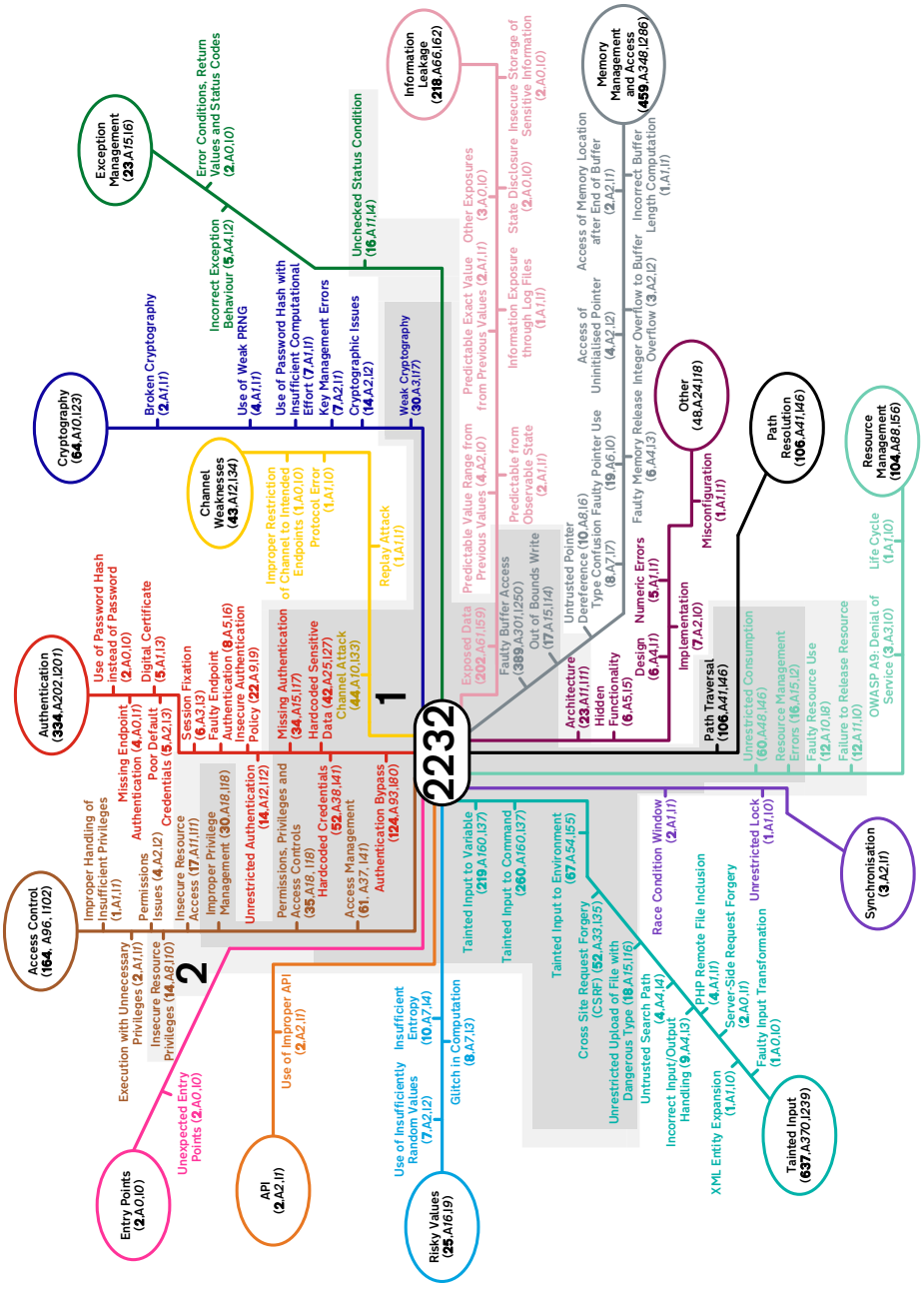

Fig. 2. A map of vulnerabilities from our dataset where each subclass (mark) is ranked based on the prevalence, availability (A) and integrity (I), and subclasses with a higher number of CVEs with critical impact rank higher. Each line represents a MITRE CWE Grouping, and each 'station' represents a subclass within that grouping. Zone 1 subclasses where ≥ 15 CVEs has a critical availability/integrity impact, and Zone 2 has ≥ 10 CVEs with a critical impact.

addressed, where classes with 15 or more critical impact CVEs are in 'Zone 1' and those with 10 or more are in 'Zone 2'.

Defining a Better Classification for ICS Vulnerabilities: While the groupings and subclasses provided in Figure 2 have distinct types of ICS vulnerabilities, they do not guide ICS owners and vendors in their detection, and contain some ambiguity. This means that, for a given grouping, many detection methods may apply but have different outcomes. Such examples include 'Tainted Input to Variable' and 'Information Leakage' where it may not be clear to an ICS owner what the effect was or how it may be detected.

We propose 8 new detectable, evidence-driven, vulnerability categories, defined below, which categorise vulnerabilities based on the detection method and techniques that can be used by ICS owners and vendors. These categories enable vendors and asset owners to understand the type of vulnerabilities that exist, where the current information is vague and lacks an application context. Our categories capture 95%[5] of all vulnerabilities within our dataset with a clear definition and specific detection methods. An overview of these methods is given in Table 2.

By classifying vulnerabilities in this way, ICS owners and vendors are able to identify techniques in which such classes of vulnerability can be found, how such vulnerabilities are manifested, and furthermore, aids in the validation of ICS device security. In Figure 3, we show the flow of ICS CVEs from their CWE groupings to our new categories. These flows are built by tagging each CVE in the dataset with its old grouping and its new detectable category and mapping changes from the old tag to the new one, where we clustered like CWEs together under a common name, based on their detectable method. For the purposes of legibility, where the count of CVEs flowing from one grouping to a category was less than 10, we exclude it from the figure (137 of 2,232 CVEs). [6]

For each of our detectable vulnerability categories, we give a precise definition and an example where a prominent ICS attack fits into that category.

Web-based Weaknesses: These vulnerabilities represent flaws and weaknesses that exist in web-based applications, for example path traversal, cross-site scripting (XSS), and cross-site request forgery (CSRF), which can be detected through conventional web scanners.

Example: The BlackEnergy [18] malware campaign, which targeted the Ukrainian power grid in 2015, used a vulnerability (CVE-2014-0751) in a GE SCADA web interface that allowed the attack to execute shell code.

Default Credentials: The use of default and hardcoded sensitive credentials has a distinct detection method, where this category has a clear proportion of vulnerabilities over time as shown in Figure 4. Vulnerabilities in this category consider

[5] Of all CVEs categorised, 94% with high availability and integrity impacts were categorised.

[6] A full Figure including these individual flows is given in our longer version of this paper.

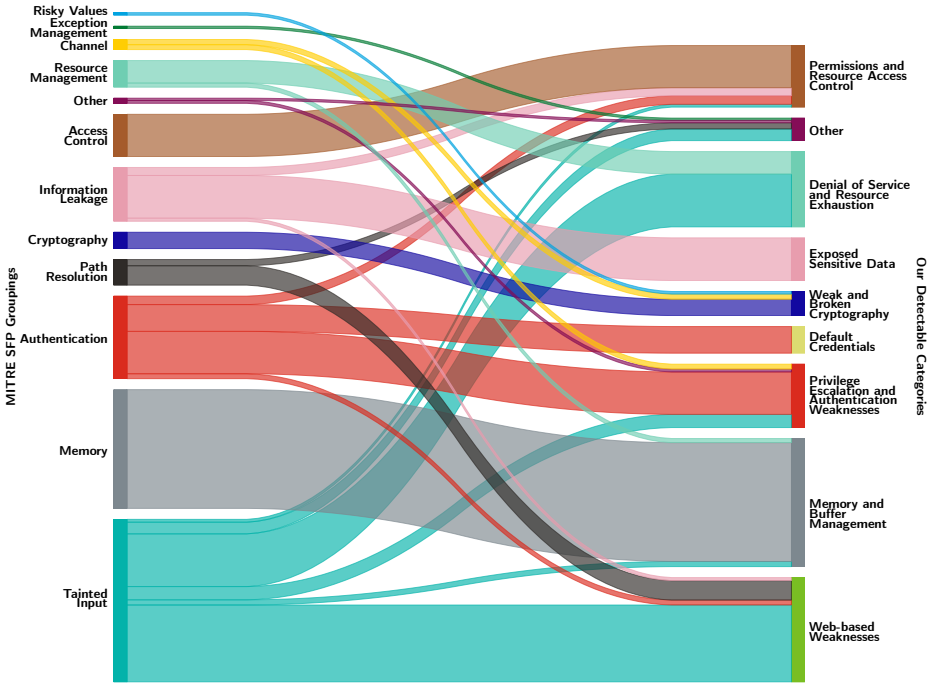

Fig. 3. Flow of CVEs from their original CWE groupings to our detectable classes.

a system, as delivered, having hardcoded credentials or sensitive data (e.g. SSH keys) which an adversary can recover and use.

Example: Stuxnet [9] targeted and damaged Iranian nuclear centrifuges. In the case of the Siemens system affected by Stuxnet, it contained hard-coded passwords, allowing the adversary to gain access to privileged functions (CVE-2010-2772).

Denial of Service and Resource Exhaustion: These vulnerabilities result in the loss of availability given a non-standard input which does not trigger some memory-related flaw in the system.

Example: CRASHOVERRIDE [6] was an attack which affected a Ukrainian power tranmission system, forcing the circuit breakers to remain in an open position, even if override commands were issued. The Siemens SIPROTEC protection relay was vulnerable to a denial of service attack (CVE-2015-5374).

Exposed Sensitive Data: Vulnerabilities in this category allow unauthenticated users to access sensitive information. Such information could be leaked via log and debug messages or stored in an openly accessible location. From our dataset, most vulnerabilities classed as information leakage either leaked user credentials or some other sensitive information.

Category	Easy to Use (new vulnerabilities)	Expert tooling (new vulnerabilities)	Tools to find existing vulnerabilities
Permissions and Resource Access Control	Access Control Policy Tooling (NIST ACPT), testing functions as a non-privileged user	Nothing Recommended	Attack Frameworks (e.g. ISF)
Privilege Escalation and Authentication Weaknesses	Check for no authentication	Network Capture and Replay tools (e.g. Wireshark)	Device-specific tools (e.g. PLC Inject, Project Basecamp)
Weak and Broken Cryptography	Source Code Scanner (SonarQube), Read Papers, Crypto Implementation Scanners (Crypto Detector)	Reverse Engineering (e.g. IDA, GHIDRA, dnspy), Manual Cryptanalysis	Device-specific tools (e.g. s7cracker, ISF)
Default Credentials	Use stated default credentials (e.g. from manuals)	Firmware Analysis (e.g. Binwalk) and search for specific artefacts, e.g. keys, shadow files	SCADA StrangeLove Default Password CSV
Denial of Service and Resource Exhaustion	Packet Storm simulators (Low Orbit Ion Cannon)	Fuzzing (e.g. AEGIS Protocol Fuzzer, Codenomicon)	Device-specific tools (e.g. EtherSploit-IP)
Exposed Sensitive Data	Simple Packet Captures (Wireshark) and search for artefacts	Manual Expert Analysis (detailed packet captures and protocol reverse engineering)	Device-specific tools (e.g. ISF, Metasploit modules, Project Basecamp)
Memory and Buffer Management	Source Code Scanner (SonarQube, Veracode)	Memory Assessment Tools (e.g. VALGRIND)	Device-specific tooling (e.g. EtherSploit-IP, ics_mem_collect)
Web-based Weaknesses	Source Code Scanner (SonarQube), Web Application Scanners (OWASP ZAP, Burpsuite)	Manual Expert Analysis (e.g. using Burpsuite)	Nothing Recommended

Table 2. Comparison of detection methods for our proposed categories.

Example: Of the 202 vulnerabilities classed using the SFP clusters as 'Exposed Data' in Figure 2, only 143 were cases of sensitive information leakage, specifically around the insecure storage of data. One of the high impact vulnerabilities in our dataset affected a Kunbus Modbus gateway, where credentials were stored in plain-text XML configurations, accessible via an FTP server on the device (CVE-2019-6549). Another vulnerability, from a LOYTEC industrial router, allowed password hashes to be read from the device and then recovered (CVE-2015-7906).

Weak and Broken Cryptography: In this category we include cryptography that is weak by design (e.g. proprietorial crypto), as well as strong cryptography that is used incorrectly, allowing it to be broken. This extends the definition of Cryptography in the SFP cluster with e.g. weak PRNGs, use of low entropy keys and certificate misuse.

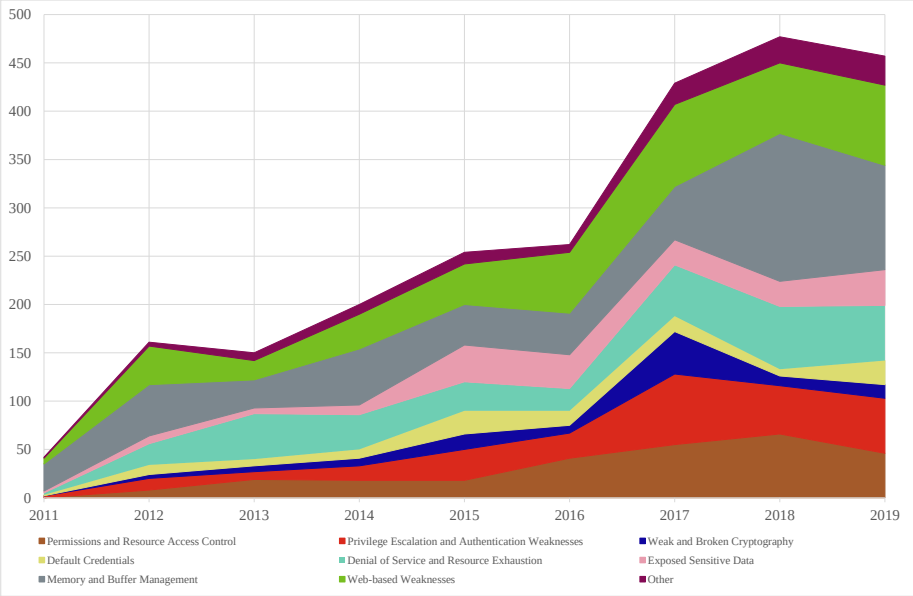

Fig. 4. ICS Vulnerabilities mapped against our new detectable classes

Example: In the case of Rogue7 [5], the MAC scheme implemented to guarantee integrity and authenticity of data between an engineering workstation (TIA Portal) and a Siemens PLC was weak, allowing an adversary to impersonate a genuine workstation to program the PLC (CVE-2019-10929). In another example using weak cryptography is the use of MD5 highlighted by CVE-2019-6563, this allows an adversary to recover passwords and gain full access to a Moxa industrial switch.

Memory and Buffer Management: Vulnerabilities which specifically relate to memory and buffer implementation flaws, for example buffer overflows, allowing an adversary to influence functionality by manipulating the memory of a system.

Example: Two example CVEs arising out of the Triton attack [16], targeting Schneider Electric safety management systems and modifying their configurations to modify, or in some cases disable the fail-safe protocols, are CVE-2018-8872 and CVE-2018-7522. In the first, memory was read directly from addresses without any verification and attacker-controlled data could be written anywhere in memory. In the case of CVE-2018-8872, the system registers were located in fixed areas of memory where modifying these registers would allow the adversary to control the system state.

Permissions and Resource Access Control: These vulnerabilities allow a user to carry out arbitrary actions on a system using standard interfaces with the privilege of another user. This could, for instance, be due to incorrect assignment

of privileges, functions being executed with excessive permissions, or a lack of access control for a given resource.

Example: On an Emerson SCADA system, an authenticated user's actions were not restricted, allowing executables and library files to be changed (CVE-2018-14791), potentially affecting the integrity of the system configuration and its availability. In another case, by using standard interfaces on a Schneider Electric PLC, an unauthenticated adversary could overwrite the password which protects the running program (CVE-2018-7791).

Privilege Escalation and Authentication Weaknesses: These vulnerabilities allow a user, privileged or not, to change their state of privilege in the system through non-standard means. The most prevalent type of vulnerability in this category is 'Authentication Bypass', where an unauthenticated user can become privileged by interacting with the system via an alternative entry point.

Example: Stuxnet, one of the first prominent attacks against an ICS system, used a vulnerability in the Siemens programming software that allowed adversaries to gain privileges by using a trojan DLL (CVE-2012-3015). This would give the adversary full control of the system state.

Discussion: 5% of vulnerabilities held in our dataset do not map directly into these distinct categories, shown in Figure 3. These vulnerabilities do not have high levels of prevalence or critical impact and require more manual, case-by-case inspection by an expert.

To demonstrate the continued prevalence of these categories over time, Figure 4 shows these remain largely within proportion over time. We also note that the distribution of detectable categories against a sample of CVEs is even, where the same detectable categories exist across most ICS device types as well as vendors, as shown in Figures 5 and 6. It is important, however, to note that in Figure 5, we show the top 6 vendors by CVE prevalence, and this should not be interpreted such that one vendor is considered more vulnerable than another, as different vendors have very different market shares.

Table 2 shows tooling that can be used to detect vulnerabilities for our 8 categories. Other tooling can be used by ICS owners to discover assets (e.g. GRASSMARLIN) and identify whether their product is vulnerable to existing, known, CVEs (e.g. Simaticscan [3], PIVoT [2], Modscan [11] and Nessus).

4 Validating Our Categories and Detection Methods

Assessing Our Categories Against New Data: For our categories to be useful for detecting new vulnerabilities we must be sure that future vulnerabilities follow the same trends as those in our dataset. At some point it is likely that industry will improve its practices, for instance, fixing the use of default passwords or carefully checking the security of all web interfaces, meaning that categories we have identified may no longer be relevant.

To test this we compared our finding to the ICS advisories issued between September 2019 and March 2020 (which were not included in our original dataset).

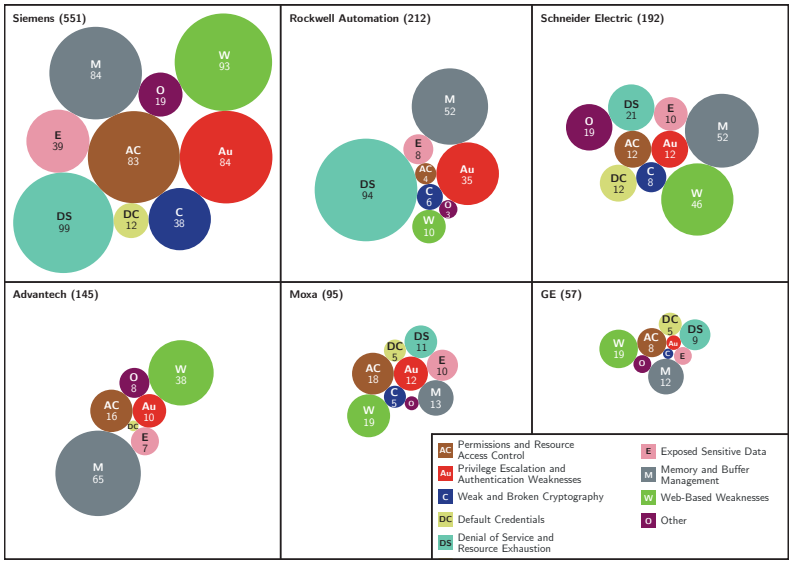

Fig. 5. The prevalence of detectable categories for the top 6 vendors by CVE count.

In this period, 126 new ICS advisories were published with 334 CVE references, which were parsed into our database and the same process to automatically classify the CVEs based on their CWE ID was taken, with final manual refinements for some CVEs that had previously unmapped CWE references.

Of the 334 CVEs parsed, 322 were directly mapped into our vulnerability classes, and 12 were not classified, an accuracy of 96%. Of the 12 vulnerabilities which were not classified, 3 related to vulnerabilities affecting 'Path Traversal' within software, a different type of vulnerability to web path traversal, 2 related to input validation, and 7 individual vulnerabilities, two of which occurred in the same product.

What this level of accuracy shows is that, with 6 months of new data, we are able to predict with high confidence which of 8 possible categories new ICS CVEs will map to, with specific tooling to support validation and verification activities. There is no sign of industry having seriously addressed any of the categories we identify. However, by using this data-driven approach and the tools we identified which support the identification of issues within our detectable categories, there is an opportunity to reduce the vulnerability space.

Validating Security Tooling and Techniques for ICS In order to validate our suggested detection methods, and to provide evidence that our categorisation does assist in finding new vulnerabilities, we applied our "easy to use" detection methods listed in Table 2 to five ICS devices - two HMIs (Phoenix Contact and Siemens) and three PLCs (two Siemens PLCs and one ABB PLC). Neither HMI had a web server, whereas all PLCs have web-servers enabled, two of which required some form of authentication to gain access to privileged functions. We did

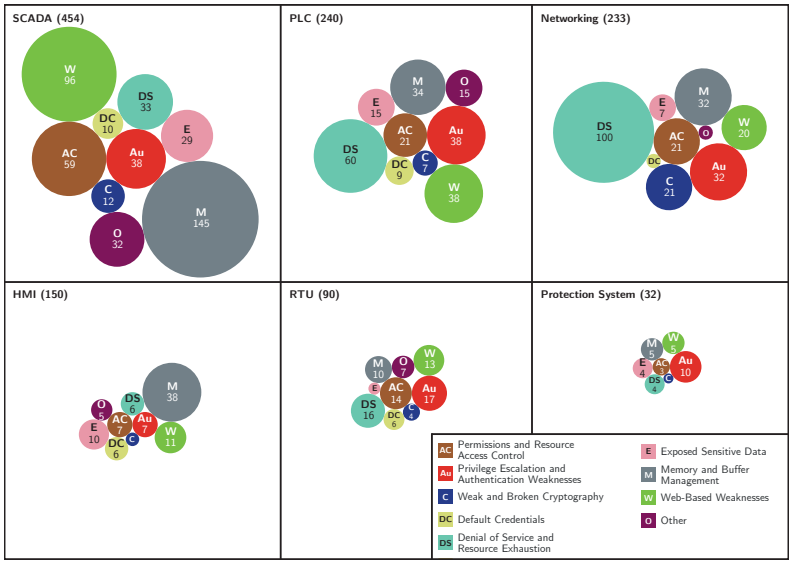

Fig. 6. The prevalence of detectable categories across different product types from a sample of 1199 CVEs.

	PLC1	PLC2	PLC3	HMI1	HMI2	Result
Default Credentials (from user manual)				✓		Manual Updated
NMAP (authentication bypass)	✓					Discussed in text
Wireshark (information leakage)					✓	CVE-2020-7592 Issued
Low Orbit Ion Cannon (Denial of Service)			✓	✓		CVE due to be released
OWASP ZAP (Web Vulnerabilities)		✓				Update to be issued

Table 3. Results from our tooling validation, ✓ = new vulnerabilities discovered

not have access to the source code of the devices, so we did not run tools which required this, or tools that required paid licenses. These results of our analysis are summarised in Table 3. It is important to note that using our categories, we are able to use the most appropriate tooling, and demonstrate how they can be taken from an IT environment and applied to OT systems. Some devices remain anonymised as the disclosure and resolution is ongoing with the vendors.

In one PLC, a previous CVE for denial of service was issued where long inputs to the web server would cause the device to enter 'Stop' mode and crashing. In our testing using Low Orbit Ion Cannon[7], we found that in the patched version, the web server would stop responding during and after a packet flood, but unlike the CVE (where the PLC would also crash), we found the PLC would continue running in this patched version. This new vulnerability would cause the ICS owner to lose visibility of the PLC via the web portal. Using the same tool on a HMI, we found that it would become unresponsive during a flood, resuming some time after the flood stopped. On that same HMI, we found default credentials in an online manual which would provide access to change its configuration, where no credentials are given with the device. For the HMI DoS, a CVE will be issued, and for the default password, the manual will be revised to state this risk.

[7] `https://github.com/NewEraCracker/LOIC`

Using standard web-scanners as unauthenticated/authenticated users, OWASP ZAP found that the Siemens S7-1200 web portal was found to have a high-criticality Open Redirect (CWE-601), which, given a malicious URL, the PLC would redirect users to an arbitrary website. We manually validated the scanner's findings for each identified weakness as part of a validation exercise. Burpsuite, however, did not find this vulnerability due to the implementation of the login form, where ZAP was able to follow the login process without any issues. Siemens confirmed this was an related issue to CVE-2015-1048 and will be patched.

For HMI2 (Siemens KTP700), which was unaffected by the Denial of Service tests, we found that part of its configuration was sent in the clear using Wireshark to capture the configuration process, leaking content which would be displayed on the screen. Siemens confirmed this as an issue, issuing CVE-2020-7592 in response to our disclosure. Finally, on the PLC which was not vulnerable to Web and Denial of Service issues, we found that where the web server requires authentication, an alternative entry point (found by using `nmap`) was found, where reverse engineering the app commands and submitting them to this entry point would give the user access to the same functionality without using a web portal. The vendor said the device should only be used on trusted networks. The vector was valid, but users should use the PLC in a secure environment.

Discussion: Using these techniques, we find six new vulnerabilities in ICS devices for which we are completing responsible disclosure with the respective vendors. All are CVE-worthy but have differing severity. For PLC1, an adversary could control the PLC state, and the web interface of PLC2 could redirect the user to a malicious website with more serious consequences. In PLC3, the visibility of the PLC is lost via the web portal, but its logic continues to run, which we believe not to be critical. For HMI1, the default credentials is not a critical issue, with the denial of service a more critical issue, as the operator may not be able to interact with the system. In the case of HMI2, the cleartext data issue is not critical, as more sensitive information, e.g. credentials, are encrypted.

5 Conclusion

ICS security has important differences to standard IT, such as vulnerability classes and detection. By analysing nine years of ICS vulnerability reports, identifying trends and suggesting eight new categories for classifying ICS vulnerabilities based on detection methods, we can better inform ICS owners and vendors on the types of ICS vulnerabilities, how they can be detected and prioritised for resolution. We discuss easy automated and in-depth testing methods for ICS owners and experts, validating our results on six months of new reports and analysing five pieces of ICS equipment, finding four new critical vulnerabilities.

Acknowledgements: Funding for this paper was provided by the National Cyber Security Centre UK (NCSC UK), Research Institute in Trustworthy Inter-Connected Cyber-Physical Systems (RITICS) and the UK Rail Research and Innovation Network (UKRRIN). We thank the Bristol Cyber Security Group for providing access to an additional device for testing.

References

1. Andreeva, O., Gordeychik, S., Gritsai, G., Kochetova, O., Potseluevskaya, E., Sidorov, S.I., Timorin, A.A.: Industrial Control Systems Vulnerabilities Statistics. Kaspersky Lab, Report (2016)
2. Antrobus, R., Green, B., Frey, S., Rashid, A.: The Forgotten I in IIoT: a vulnerability scanner for industrial internet of things. IET (2019)
3. Antrobus, R., Frey, S., Green, B., Rashid, A.: Simaticscan: Towards a specialised vulnerability scanner for industrial control systems. In: 4th International Symposium for ICS & SCADA Cyber Security Research (2016)
4. Beresford, D.: Exploiting Siemens Simatic S7 PLCs. Black Hat USA (2011)
5. Biham, E., Bitan, S., Carmel, A., Dankner, A., Malin, U., Wool, A.: Rogue7: Rogue Engineering-Station attacks on S7 Simatic PLCs. Black Hat USA (2019)
6. Dragos: CRASHOVERRIDE: Analysis of Threat to Electric Grid Operations (2017)
7. Dragos: 2018 Year in Review - Industrial Controls System Vulnerabilities (2018)
8. Dragos: 2019 Year in Review - ICS Vulnerabilities (2019)
9. Falliere, N., Murchu, L.O., Chien, E.: W32. stuxnet dossier. White paper, Symantec Corp., Security Response (2011)
10. Gonzalez, D., Alhenaki, F., Mirakhorli, M.: Architectural security weaknesses in industrial control systems (ics) an empirical study based on disclosed software vulnerabilities. In: 2019 IEEE International Conference on Software Architecture (ICSA) (2019)
11. Hankin, C., Chothia, T., M3, P., Popov, P., Rashid, A., Sezer, S.: Availability of Open Source Tool-Sets for CNI-ICS (2018)
12. Hemsley, K.E., Fisher, E., et al.: History of Industrial Control System Cyber Incidents. Tech. rep. (2018)
13. Hui, H., McLaughlin, K.: Investigating current PLC security issues regarding Siemens S7 communications and TIA Portal. In: 5th International Symposium for ICS & SCADA Cyber Security Research (2018)
14. Industrial Control Systems Cyber Emergency Response Team: ICS-CERT Annual Assessment Report FY 2016 (2016)
15. Jiang, Y., Atif, Y., Ding, J.: Cyber-Physical Systems Security Based on a Cross-Linked and Correlated Vulnerability Database. In: Critical Information Infrastructures Security (2020)
16. Johnson, B., Caban, D., Krotofil, M., Scali, D., Brubaker, N., Glyer, C.: Attackers Deploy New ICS Attack Framework "TRITON" and Cause Operational Disruptionto Critical Infrastructure (2017)
17. Kaspersky ICS CERT: Threat Landscape for Industrial Automation Systems (2019)
18. Khan, R., Maynard, P., McLaughlin, K., Laverty, D., Sezer, S.: Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid. In: 4th International Symposium for ICS & SCADA Cyber Security Research (2016)
19. Nelson, T., Chaffin, M.: Common cybersecurity vulnerabilities in industrial control systems. Control Systems Security Program (2011)
20. Niedermaier, M., Malchow, J.O., Fischer, F., Marzin, D., Merli, D., Roth, V., Von Bodisco, A.: You snooze, you lose: measuring PLC cycle times under attacks. In: 12th USENIX Workshop on Offensive Technologies (WOOT) (2018)
21. OWASP: OWASP Top 10 - 2017: The Ten Most Critical Web Application Security Risks (2017)