Towards Context-Aware Risk Assessment Scoring System for IoT/IIoT Devices

Valerio G. Massaro[1,2], Luca Capacci[2] and Rebecca Montanari[1]

[1]*Alma Mater Studiorum - University of Bologna, Bologna, Italy*
[2]*CryptoNet Labs s.r.l, Milano, Italy*

Abstract

The widespread adoption of the (Industrial) Internet of Things (IoT/IIoT) has increased the attack surface for malicious attackers who exploit smart devices as entry points for accessing the internal systems of industries, institutions, and home networks. The need to carry out vulnerability assessment and to assign vulnerabilities a severity score as accurate as possible, is therefore of primary importance also in the IoT/IIoT domain. The current reference scoring system is the Common Vulnerability Scoring System (CVSS) [1]. It is used to assign a severity score to vulnerabilities and misconfigurations - often found in MITRE's Common Vulnerability and Exposures (CVE) database [2] [3] - on both commercial and open-source software products. The CVSS assigns a severity score based on several metrics, which model the characteristics of vulnerabilities in a general-purpose manner. However, IoT/IIoT software products operate in environments with unique characteristics that require dedicated contextualization. Prioritizing and resolving a vulnerability using a calculated level of risk that does not consider the technical specificities of the context in which software product operates - in our case IoT/IIoT - means spending resources and time incorrectly either overestimating or underestimating the problem. The following paper intends to present a context-aware scoring system for IoT/IIoT environment that, taking CVSS as starting point, extends its scope by presenting an overall assessment method of the severity and risk of vulnerability that best reflects the peculiarities of this environment.

Keywords

Internet of Things, Vulnerability risk quantification, CVSS

1. Introduction

The (Industrial) Internet of Things refers to the increasingly widespread practice of connecting everyday objects to the network in order to create systems for monitoring and controlling the physical environment around them [4]. These devices are often equipped with actuators and appropriate hardware sensors properly controlled by software components which allow first, to measure the surrounding environmental parameters, then, to act on the same environment, possibly modifying it.

One of the main characteristics of the IoT/IIoT is its extraordinary applicability in both the industrial and domestic fields, which at the same time determines a strong application heterogeneity. Today, we live in smart homes, wear smart devices, and largely work with automated

ITASEC 2023: The Italian Conference on CyberSecurity, May 03–05, 2023, Bari, Italy
valerio.massaro2@unibo.it (V. G. Massaro); luca.capacci@cryptonetlabs.it (L. Capacci); rebecca.montanari@unibo.it (R. Montanari)

systems. This wide diffusion also arouses lavish interest from malicious actors who exploit these new access points to enter the systems of industries, institutions, and home networks and take control of them [4]. This is further reinforced by the fact that most smart devices are outdated and/or have low computational capabilities which are often unable to guarantee minimum cyber security requirements. An example is Stuxnet, a malware that disabled the centrifuges of an enrichment plant in Natanz, Iran, preventing them from malfunctioning [5]. Beyond the economic damage, the most relevant aspect in this case lies in the capacity of such systems to act on the actual physical environment and potentially cause damage to the individuals living there. A second peculiarity lies in their strong contextual specificity. This means that ecosystems and devices can be widely different and, certainly, distant from a generic IT technological context. A smart device may be accessible from the Internet or only via LAN, it may be connected to other devices via wi-fi and/or bluetooth, it may be monitored via a mobile application, and so on. This makes the analysis of real security risks considerably complicated, and it is very difficult to define general security standards [4]. A second peculiarity lies in their strong contextual specificity. This means that ecosystems and devices can be widely different and, certainly, distant from a generic IT technological context. A smart device may be accessible from the Internet or only via LAN, it may be connected to other devices via wi-fi and/or bluetooth, it may be monitored via a mobile application, and so on. This makes the analysis of real security risks considerably complicated, and it is very difficult to define general security standards [4].

Up to now, in fact, the few relevant attempts to set security requirements are represented by (i) the OWASP IoT Security Verification Standard (ISVS) (2021) [6] and (ii) the ISO IEC-62443, which establish rules and key points to achieve certain levels of security within IoT and operational ecosystems in general. Although they comply with the peculiar aspects of this world, they do not deal with defining a quantitative vulnerability risk assessment system. This means that one still must rely on standard scoring systems to quantify the vulnerability ‘s associated risk.

Nowadays, the de facto standard is the CVSS, a calculation system designed by the National Infrastructure Advisory Council (NIAC) [7] and currently maintained by the Forum of Incident Response and Security Teams (FIRST) [8] that assigns a severity score from 0 to 10 to vulnerabilities identified in any software product. In its most recent version, CVSS 3.1 [9] distinguishes three different sets of metrics: basic, temporal, and environmental. In risk analysis activities, the base score is often preferred over the other two, mainly due to its temporal and spatial immutability and despite its general-purpose nature [10] [11] [12]. Whether considering a general-purpose context it may also have its own reason, the same cannot be said in those highly specialized use cases such as IoT/IIoT devices. A denial of service vulnerability, for instance, detected on a specific version of a PostgreSQL database installed on an IoT device, has a very high CVSS base score (e.g. CVE-20163065). If, however, the device is never connected to the network (neither Internet nor LAN) then the impact of the vulnerability is not that significant and certainly not prioritized over others.

Thus, strong inconsistencies arise between what is reported in the final analysis deliverable and the actual severities of the issues identified. This strong gap is certainly linked to the

difference between the need to generalize of CVSS - as a reference standard for scoring generic software products - and the strong specialization of the IoT/IIoT context - which requires finer-grained parameters for calculating the final score.

This paper makes the following contributions:

- Describes the need to expand the modelling scope of CVSS in order to encompass also those contextual technical features that affect the severity of a vulnerability, specifically for the IoT/IIoT world;
- Proposes IoT Context-Aware Scoring System (ICASS), a context-aware CVSS-based risk scoring system for IoT/IIoT, defining new technical context metrics and equations for the corresponding calculation.
- Performs a comparison between results of the proposed scoring system and CVSS, considered in its entirety (base, temporal, environmental, and overall score), emphasizing similarities and differences.

2. Motivation

The research carried out and the experience gained over the years in the field of Vulnerability Assessment and Penetration Testing (VAPT) of IoT/IIoT devices, show that they should be treated with specific techniques and methods compared to generic IT technologies. This also leads to the issue of how to accurately quantify the risk associated with a vulnerability. Traditional CVSS is too general-purpose to capture the complexity of IoT/IIoT environments and to precisely model the vulnerability risks, motivating several research efforts to revise current CVSS solutions to better adapt to the specific operating characteristics of IoT/IIoT devices.

2.1. CVSS shortcomings

The CVSS is the current reference risk scoring system. In its newest released version 3.1 it presents three different scoring metrics: base, temporal, and environmental. The base score measures both the exploitation difficulty level and the vulnerability impact in case it is exploited and remains constant over time and space. The temporal score, on the other hand, considers characteristics of the vulnerability that change over time such as the level of maturity of available remediation techniques, the current state of exploitation techniques, and the public availability of information related to the vulnerability itself. Finally, the environmental score provides a fine-grained parameterization of the severity beyond the base score, also considering the nature of assets potentially affected. The CVSS also provides an overall score that combines the results of the previous three scores. However, some metrics are not properly aligned with the IoT/IIoT context, while other ones should be added to better address the specific context, so it is important to understand the motivations that led us to reconsider and expand the scope of CVSS metrics.

The base score considers, among the exploitability metrics, the Attack Vector (AV) which measures the "distance" from which an attacker can carry out an attack. However, the operating

conditions in which IoT/IIoT devices operate are often more limited. It may often happen that these devices are not connected to the Internet, are physically protected, and/or are difficult to access from the outside. The concept expressed by the AV must therefore be reconsidered based on the operating context in which the vulnerability is detected. If "Network" is assigned to AV, but the device is never connected to the public network, the final base score must certainly be leveled to be more accurate compared to the generic case. The same argument can be repeated for the values "Adjacent", "Local", and "Physical": the device can prevent others from establishing connections with it even in the local network, it can have shell-based access disabled by design, and it can be protected through physical mechanisms that prevent an attacker from touching it, respectively. Continuing on, the temporal metrics take into account the existence of appropriate remediations (i.e. Remediation Level, RL) decrementing the base score depending on the maturity level of the fixes. However, the mere availability of a remedy does not justify a decrease in the severity score. It should also be added that most maintainers of supported software products provide official fixes shortly after the publication of the related CVEs. This means that of the available values to assign to RL, "Official fix" is probably the most relevant (i.e. no base score decreases). Similarly, for products with longer maintenance and replacement times - such as IoT/IIoT devices and firmware - the mere presence of fixes does not provide information about the actual time of their adoption: proceeding with a decrease in the base score therefore would only create misunderstandings during the correction prioritization phase, resulting in an incorrect allocation of time and resources. Likewise, the Report Confidence parameter, which measures the level of confidence in the technical details related to the vulnerability, determines a negative change in the base score: the less details there are, the more the score decreases. Factors such as the reputation of the source, the quality of the information provided, and the level of testing performed can all affect the level of confidence in a vulnerability report, but these factors may be difficult to quantify or compare across different reports and may introduce subjective judgement. Furthermore, neither vulnerability discovered and not been (yet) published in the MITRE CVE database nor a known CVE with a dubious source are necessarily less severe and less worthy of being prioritized than official and confirmed ones. In light of this, the only relevant temporal metric remains the exploit code maturity, since it is reasonable to assign a higher priority to vulnerabilities with publicly available exploits and lower the score of issues without them. The same issues described for base metrics also apply to the environmental score, since a part of this score reflects the same metrics of the base score (i.e. exploitability metrics) and the remaining part is strictly related to asset company CIA (Confidentiality, Integrity, and Availability) requirements and are heavily based on subjective requirements, which do not represent the focus of this work as described in Section 4.1.

At the end of the day, therefore, by taking the CVSS and considering it in light of the peculiarities of IoT/IIoT, the shortcomings and criticalities described previously are highlighted. This leads us to fine-tune the risk scoring system in a way it is tailor-made to the IoT/IIoT context and therefore to propose and compute new overall scoring values more adherent to the considered environment.

3. Related Works

Several articles have highlighted the incompleteness of CVSS in accurately modeling a vulnerability in several contexts. In [13], they emphasize how an incorrect attribution of the severity level to the vulnerabilities of a system results in a wrong prioritization of correction activities, leading to a waste of time and resources. They therefore conclude that there is a need to improve scoring systems by using contextual information that increases accuracy in risk quantification. The same occurs in [10] where, after presenting the limitations of the 2.0 version of CVSS, a new CVSS-based severity score calculation method is proposed with the addition of some parameters to outline the state and type of the analyzed system. Other research, on the other hand, supports the thesis that emerging technologies (e.g. IoT and Cloud) require an expansion of the CVSS scope due to the highly specialized context in which they operate [12] [11]. In [12], starting from an example of CVSS base score calculated on a smart-car vulnerability, the context in which the car works is carefully analyzed, concluding that the conditions in which the car operates are so adverse to make more difficult exploiting that vulnerability. Thus, the resulting high score should be reconsidered, first by integrating it with already available CVSS metrics (temporal and environmental), and second by introducing new ones strictly linked to the application context. The same argument is presented in [11] but focusing on a different target, i.e., Cloud Services. The framework presented, from a non-intrusive contextual analysis of any cloud service, produces a severity score considered more accurate because it is modeled precisely with respect to the characteristics of the service itself. However, differently from our paper, none of the previously mentioned works focus on risk scoring for IoT/IIoT products despite the pressing need for a more precise modeling through new purpose-built metrics, as also demonstrated in [14].

4. The Iot Context-Aware Scoring System

This paper presents a different scoring system called IoT Context-Aware Scoring System (ICASS) which proposes a set of new IoT/IIoT context-based metrics - derived from the contextual analysis of related software product - and a formulation that, starting from the CVSS base score of a vulnerability, better adapts the score associated with the application context.

4.1. Metrics

Below, we propose metrics grouped into semantically related categories based on external exposure degree of the device. One aspect to emphasize is the type of metrics introduced. The proposed system adds purely technical parameters related to the peculiarities of the IoT/IIoT context. For this reason, neither metrics such as confidentiality, integrity, and availability requirements are touched in their definition, nor safety metric is introduced (i.e. the measure of the software's danger in terms of physically damaging individuals and things). Although very interesting, this is not the focus of this work.

Internet Exposure As many IoT devices often do not have access to a large number of resources and/or do not require external services, it is common to find IoT devices that are

not exposed on the Internet network (e.g. printers, smart meters, PLCs, etc.). Therefore, a vulnerability that can be exploited only via the public network has a different impact compared to the generic IT context. For this reason, the associated risk requires more precise modeling. The following two metrics belong to this semantic group:

- is_exposed_to_internet: indicates whether the device is exposed to the public Internet network through listening ports (True if the device has one or more exposed services; False otherwise).
- can_connect_to_internet: indicates whether the device, in turn, invokes services exposed by others on the Internet network (True if the device invokes one or more exposed services; False otherwise).

Intranet Exposure The discussion for the Internet Exposure group is repeated for local networks. The possibility to have an IoT device not connected to the local network - acting as an embedded device - exists as well. There are two parameters identified:

- is_exposed_to_lan: indicates if the device is exposed on a local network (True if the device has one or more services; False, otherwise).
- can_connect_over_lan: indicates if the device, in turn, invokes services exposed on a local network (True if the device invokes one or more exposed services; False, otherwise).

Shell Exposure The heterogeneous nature of IoT/IIoT devices and their limited computational capabilities mean that they are not necessarily equipped with the conventional resources that a generic IT system provides. In the case of bare-metal as well as real-time operating systems described in [4], no one neither guarantees that a shell to access to the device exists nor that it is designed to be accessible. For this semantic group, the parameter is unique:

- is_shell_accessible: indicates if the firmware is remotely accessible through a shell. In some cases, the shell might not be included in the firmware or made inaccessible (True if it is possible to access the shell; False otherwise)

Physical Protection Many devices are protected by physical defense mechanisms (e.g. mechanical, thermal, etc.), used to prevent physical access to the device. An attack that requires physical contact with the device's hardware components is more complex to carry out. Once again, a more precise modeling is required. The following parameter belongs to this group:

- is_physically_protected: indicates the presence of physical protection mechanisms for the device and its components (True if a physical protection mechanism is provided; False otherwise).

Exploit Code Maturity The "Exploit Code Maturity" parameter defines the likelihood of the vulnerability being exploited and is typically based on the current state of the art about exploit techniques and general availability of exploit code and software. The heterogeneous hardware components, the use of custom-designed boards and proprietary protocols and standards, makes

it very difficult to prepare custom exploits, adapt existing exploits, and use ready-to-use exploits. We then understand how this parameter represents a good metric for contextualizing the final score. Since CVSS 3.1 already provides d values for the Exploit Code Maturity, we simply reuse them in our scoring system [9].

4.2. Risk calculation

The final score calculation is performed as follows. Firstly, given one vulnerability, the corresponding CVSS base score is calculated. Secondly, the new metrics are evaluated based on well-defined rules available in Table 2. Depending on their occurrence, the weights included in Table 1 are considered or not in the final calculation. Finally, the initial CVSS base score is multiplied by each valid weight. The result is represented by the approximation to the first decimal place of the smallest value between the product and 10 or the largest value between 0 and the product. The final weights are the result of an analysis done in the field with the help of feedback from CryptoNet Labs S.r.l. (Milan, Italy) internal pentesters. The goal is to find the right balance between appreciating significant changes (e.g. moving from one severity to another) and not completely altering the base score. As a result, all chosen weights change the starting score by a value close to 10% for each, when applied. Regarding the exploitability code maturity, the contextual aspects described in Section 2.1 are also considered. As a result, each associated weight starts from the definition of CVSS given by FIRST and is decremented by one or more decimal points, highlighting the practical complexities of exploitability of IoT/IIoT devices.

Table 1
Weights

	Metric's group	Value	Weight
1	Physical Protection	if corresponding conditions are True	0.90
2	Shell Exposure	if corresponding conditions are True	0.89
3	Intranet Exposure	if corresponding conditions are True	0.89
4	Internet Exposure	if corresponding conditions are True	0.88
5	Exploit Code Maturity	N (Not Defined)	1
		H (High)	1
		F (Functional)	0.93
		P (Proof of Concept)	0.89
		U (Unproven)	0.84

5. Experimental evaluation

In this Section, an evaluation of the described scoring system is performed once applied to existing vulnerabilities identified during VAPT activities on a commercially available device whose details, for confidentiality reasons, are anonymized. Specifically, the results are compared with both the base, temporal, environmental, and overall score of CVSS.

Table 2
Conditions and Formulas

	Rule	Formula
	IoT Context-Aware Exploitability (ICAE) =	
1	If 'is_physically_protected' is True and CVSS AV is 'P'	$CVSSBaseScore \times PhysicalProtection$
2	If 'is_shell_accessible' is False and CVSS AV is 'L'	$CVSSBaseScore \times ShellExposure$
3	If 'is_exposed_to_lan' is False and 'can_connect_over_lan' is False and CVSS AV is 'A is True	$CVSSBaseScore \times IntranetExposure$
4	If 'is_exposed_to_internet' is False and 'can_connect_to_internet' is False and CVSS AV is 'N'	$CVSSBaseScore \times InternetExposure$

5.1. Testbed description

The software product under examination is a real-world Linux-based firmware designed for embedded devices. The device it runs on has a LAN port which enables access to a local network where it exposes several services. Moreover, the device is never connected to the Internet and is physically unprotected. Finally, it is classified as critical with respect to the CIA requirements. The results of the software composition analysis activities revealed various components affected by numerous CVEs. Forty-six of them were selected as a case study in order to have a heterogeneous sample to evaluate all the introduced metrics. The list of selected vulnerabilities with scores is available in the Appendix A and summarized in the Table 3.

Table 3
Summary of detected CVEs which affect a commercial off the shelf device.

Component	Version	#CVEs
U-Boot	2021.02.3	13
Nginx	1.18.0	4
wpa_supplicant	2.9	4
LibTIFF	4.4.0	2
lua	5.4.4	12
hostapd	2.9	3
BusyBox	1.33.0	4
OpenSSL	1.1.1n	4

5.2. CVSS in real-world IoT scenarios

In this section, we calculate the temporal and environmental score of CVE-2022-23304, step-bystep. The same procedure is repeated for the other vulnerabilities under examination, but for brevity, the results are directly available in Appendix A.

The CVE-2022-23304 affects version 2.9 of a software named hostapd, used for creating wi-fi access points and setting up authentication servers, and has a base score vector of AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H. Below is the calculation:

- *Temporal Score*: there is no evidence of existing exploits for the vulnerability, neither on the network, nor in the hyperlinks related to the corresponding NIST page, nor on exploitdb, nor finally on the Rapid7 Vulnerability & Exploit Database. At the same time, among the hyperlinks available on the corresponding NIST page, there is a URL to the official fix for the vulnerability. Finally, being included in the CVEs, this vulnerability can be considered confirmed. Final values: 'Unproven' (exploitability code maturity), 'Official fix' (remediation level), and 'Confirmed' (report details).
- *Environmental Score*: given that the device cannot communicate with the public network, it is necessary to modify the attack vector. Being a vulnerability that affects a wi-fi access point creation tool, we overwrite with the value 'Adjacent'. The device also requires high levels of confidentiality, integrity, and availability. With respect to the reference context, all other metrics are confirmed. Final values: 'Adjacent' (attack vector), High, High, High (confidentiality, integrity, and availability requirements).

The temporal score is 8.5, while the environmental and overall scores are 7.7. For the calculation of all the other CVEs, newer versions with corresponding vulnerability resolutions are always identified and no exploits are detected from the sources mentioned above except for CVE-2021-23017 for which a public exploit is found on exploit-db. Additionally, all CIA requirements metrics are setup to 'High' value and modified AV environmental metric is decreased to 'Adjacent' just in those cases it is equal to 'Network'.

5.3. ICASS in real-world IoT scenarios

Continuing with the metrics proposed in this paper, evaluating the resulting conditions, and applying the formula presented, we proceed with the calculation of the new score. Given the assumptions defined in Section 5.1, we assign the values for each metric as shown in Table 4.

We now proceed with the calculation of ICA Score on CVE-2022-23304. Since the device does not expose or use services exposed on the public network, and at the same time has a base score attack vector set to 'Network', the rule (4) in Table 2 applies. The corresponding weight of 0.88 is multiplied by the CVSS base score which is 9.8. This results in IoT Context-Aware Exploitability = 8.6. As stated in Section 5.2, there is no evidence of exploits for the vulnerability. Consequently, the exploit code maturity is 'Unproven'. The corresponding weight is multiplied by IoT Context-Aware Exploitability.

The same calculation is summarized as follows:

$$ICAScore = CVSSbasescore \times 0.88 \times 0.84 = 7.2$$

Table 4
Values assigned to the metrics according to the testbed description.

Metric	Value
is_exposed_to_internet	False
can_connect_to_internet	False
is_exposed_to_lan	True
can_connect_over_lan	True
is_shell_accessible	False
is_physically_protected	False
Exploit Code Maturity	Variable, depending on individual CVE

5.4. Comparison

From the table presented in the Appendix A, we note that Severity CVSS score range Critical ICASS consistently reclassifies scores of CVEs downward compared to CVSS. The reason behind this is related to a more precise modeling of both the exploitability complexity and technical exposure characteristics of IoT/IIoT devices. Specifically, the percentage decrease of the ICA score in reference to the CVSS base score – calculated as the percentage of the CVSS score of the absolute difference between the scores in question - is equal to 23.4%, which is a significant difference. Since other CVSS parameters' goal is to better contextualize the CVSS base score, a comparison between them and the new ICA score is useful. The values differ in a percentage ranging between 12.5% and 15.8%, which again is significant. This is because CVSS parameters are not designed with the IoT context in mind, but with a general-purpose goal. On the contrary, ICASS uses custom built parameters for the IoT context, so its score is helpful in risk analysis activities and represents a key indicator for prioritizing fixes and allocating resources for IoT/IIoT companies.

Table 5
CVSS 3.1 qualitative severity levels.

CVSS 3.1 score range	Severity level
9 - 10	Critical
7 - 8.9	High
4 - 6.9	Medium
0.1 - 3.9	Low
0	Information

Let us now consider the score with respect to the severity categories defined by FIRST and assigned to vulnerabilities based on their numerical score (Table 5). From the graph in Figure 1,

we note that severities obtained from the temporal and environmental metrics vary the severity of the base CVSS by at most one level. They are designed "to produce a severity more accurate for their organizational environment" [9] without aiming to disrupt the base calculation. The same behavior is observed in ICASS (Figure 2), highlighting how the presented system inherits the same philosophy as the additional metrics of CVSS, aiming not to revolutionize the entire system - still a reference point for an entire industry - but to make specific modifications to it based on a precise application context.

Figure 1: CVSS Base severity vs CVSS Temporal severity vs CVSS Environmental severity

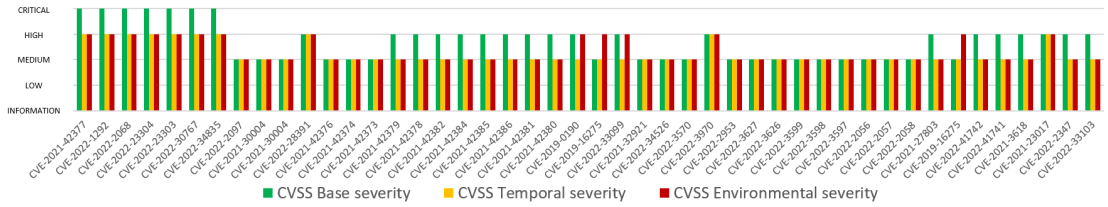
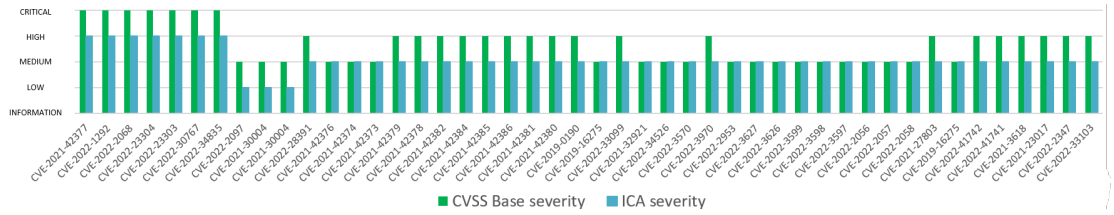


Figure 2: CVSS Base severity vs ICA severity

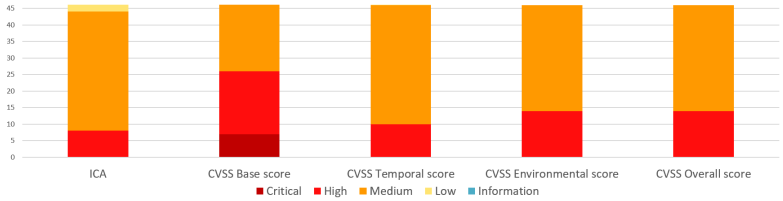


Finally, Figure 3 summarizes the results obtained with respect to severity levels. The overall image reflects what has been previously discussed, with a general decrease in severity level compared to all scores for forty-six CVEs under examination. In particular with respect to this case study, one of the most evident benefits is the shift of the severity average from the high/critical categories of the CVSS base severity to the medium/low of the ICA severity. This leads to a more effective prioritization of vulnerabilities and resource allocation, and at the same time to less waste of time and money.

6. Conclusion

This paper proposes a method for quantifying the severity of vulnerabilities in IoT/IIoT devices, which builds upon the de facto standard CVSS and introduces context-specific metrics to improve the accuracy of the resulting score. The proposed scoring system overcomes shortcomings and limitations of CVSS with respect to the IoT/IIoT field. It begins by calculating the CVSS base score, then by applicating correctives based on the values chosen for the introduced metrics.

Figure 3: Severity count comparison



These metrics are quantified based on weights derived from real-world cases and scenarios and reflect the IoT/IIoT context. Experiments conducted show that without revolutionizing the de facto standard CVSS but simply making corrections and additions related to the IoT technical context, we can obtain a greater degree of accuracy in evaluating vulnerabilities severities. The consequences of a better severity attribution are reflected in benefits in terms of prioritization of vulnerabilities and, thus, in a better allocation of economics and time resources.

References

[1] FIRST, Common vulnerability scoring system, 2023. URL: https://www.first.org/cvss/.

[2] MITRE, Mitre, 2023. URL: https://attack.mitre.org/.

[3] Wikipedia, Common vulnerabilities and exposures, 2023. URL: https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures.

[4] M. Abdur Razzaq, S. Habib, M. Ali, S. Ullah, Security issues in the internet of things (iot): A comprehensive study, International Journal of Advanced Computer Science and Applications 8 (2017). doi:10.14569/IJACSA.2017.080650.

[5] R. Langner, Stuxnet: Dissecting a cyberwarfare weapon, IEEE Security and Privacy (2011) 49–51. doi:10.1109/MSP.2011.67.

[6] OWASP, Iot-security-verification-standard-isvs, 2023. URL: https://owasp-isvs.gitbook.io/owasp-isvs-pr.

[7] Wikipedia, National infrastructure advisory council, 2023. URL: https://en.wikipedia.org/wiki/National_Infrastructure_Advisory_Council.

[8] FIRST, First - improving security together, 2023. URL: https://www.first.org/.

[9] FIRST, Common vulnerability scoring system version 3.1: Specification document, 2023. URL: https://www.first.org/cvss/specification-document.

[10] R. Wang, L. Gao, Q. Sun, D. Sun, An improved cvss-based vulnerability scoring mechanism, in: 2011 Third International Conference on Multimedia Information Networking and Security, 2011, pp. 352–355. doi:10.1109/MINES.2011.27.

[11] H. Zhuang, F. Pydde, A non-intrusive and context-based vulnerability scoring framework for cloud services, 2016.

[12] D. J. Klinedinst, Sei blog, 2023. URL: https://insights.sei.cmu.edu/blog/cvss-and-the-internet-of-things/.

[13] C. Fruhwirth, T. Mannisto, Improving CVSS-based vulnerability prioritization and response with context information, Addison-Wesley, 2009, pp. 535–544.

[14] G. Gori, A. Melis, L. Rinieri, M. Prandini, A. Al Sadi, F. Callegati, Metrics for Cyber-Physical Security: a call to action, Addison-Wesley, 2022, pp. 1–4. doi:10.1109/ISNCC55209.2022.9851735.

A. Calculated scores for CVEs

Vulnerability	ICA score	Base score	Temporal score	Env. score	Overall score
CVE-2021-42377	7.2	9.8	8.5	7.7	7.7
CVE-2022-1292	7.2	9.8	8.5	7.7	7.7
CVE-2022-2068	7.2	9.8	8.5	7.7	7.7
CVE-2022-23304	7.2	9.8	8.5	7.7	7.7
CVE-2022-23303	7.2	9.8	8.5	7.7	7.7
CVE-2022-30767	7.2	9.8	8.5	7.7	7.7
CVE-2022-34835	7.2	9.8	8.5	7.7	7.7
CVE-2022-2097	3.9	5.3	4.6	4.4	4.4
CVE-2021-30004	3.9	5.3	4.6	4.4	4.4
CVE-2021-30004	3.9	5.3	4.6	4.4	4.4
CVE-2022-28391	6.5	8.8	8.1	7.3	7.3
CVE-2021-42376	4.6	5.5	4.8	6.4	6.4
CVE-2021-42374	4.5	5.3	4.6	5.9	5.9
CVE-2021-42373	4.6	5.5	4.8	6.4	6.4
CVE-2021-42379	5.3	7.2	6.3	5.9	5.9
CVE-2021-42378	5.3	7.2	6.3	5.9	5.9
CVE-2021-42382	5.3	7.2	6.3	5.9	5.9
CVE-2021-42384	5.3	7.2	6.3	5.9	5.9
CVE-2021-42385	5.3	7.2	6.3	5.9	5.9
CVE-2021-42386	5.3	7.2	6.3	5.9	5.9
CVE-2021-42381	5.3	7.2	6.3	5.9	5.9
CVE-2021-42380	5.3	7.2	6.3	5.9	5.9
CVE-2019-0190	5.5	7.5	6.5	7.2	7.2
CVE-2019-16275	5.5	6.5	5.7	7.2	7.2
CVE-2022-33099	5.5	7.5	6.5	7.2	7.2
CVE-2021-32921	4.4	5.9	5.2	6.2	6.2
CVE-2022-34526	4.8	6.5	5.7	6.5	6.5
CVE-2022-3570	4.6	5.5	4.8	6.4	6.4
CVE-2022-3970	6.5	8.8	7.7	7	7
CVE-2022-2953	4.6	5.5	4.8	6.4	6.4
CVE-2022-3627	4.8	6.5	5.7	6.5	6.5
CVE-2022-3626	4.8	6.5	5.7	6.5	6.5
CVE-2022-3599	4.8	6.5	5.7	6.5	6.5
CVE-2022-3598	4.8	6.5	5.7	6.5	6.5
CVE-2022-3597	4.8	6.5	5.7	6.5	6.5
CVE-2022-2056	4.8	6.5	5.7	6.5	6.5
CVE-2022-2057	4.8	6.5	5.7	6.5	6.5
CVE-2022-2058	4.8	6.5	5.7	6.5	6.5
CVE-2021-27803	6.3	7.5	6.5	6.5	6.5

CVE-2019-16275	5.5	6.5	5.7	7.2	7.2
CVE-2022-41742	6	7.1	6.2	6.8	6.8
CVE-2022-41741	6.6	7.8	6.8	6.8	6.8
CVE-2021-3618	5.5	7.4	6.4	6.5	6.5
CVE-2021-23017	6.8	7.7	7.4	7.2	7.2
CVE-2022-2347	5.4	7.1	6.2	6.3	6.3
CVE-2022-33103	6.6	7.8	6.8	6.8	6.8