The Common Vulnerability Scoring System (CVSS) generations – usefulness and deficiencies

Attila Horváth PhD

Principal researcher
Foundation for Information Society
e-mail: horvath.attila@infota.org

Péter Máté Erdősi

PhD student
National University of Public Service Doctoral School of Public Administration
e-mail: erdosi.peter.kdi@office.uni-nke.hu

Ferenc Kiss PhD

Researcher
Foundation for Information Society
e-mail: kiss.ferenc@infota.org

Abstract

The Common Vulnerability Scoring System (CVSS) is one of the most common tools to assess vulnerability threats on IT-systems. We have used it excessively in our research, it is a useful tool but we soon met its week points as we haves started to use version 2 of the framework at the beginning of the research, and we have stick to it during the whole process, as the scores for vulnerabilities published before the release of version 3 has not been converted into the new system. So in order to maintain the continuity, we used v2 during the whole research, nevertheless we have studied the new version closely, to ascertain, whether the deficiencies has been corrected. This article aims the reader to get familiar with the CVSS metrics, the main problems with the former version and how things changed for the most up-to-date methodology. The project was founded by the National Research, Development and Innovation Office - NKFIH under registration number PD-109740.

Keywords: *CVSS, Common Vulnerability Scoring System, Base, Temporal, Environmental, score, metrics, IoT, Internet of Things*

1. The CVSS framework

The Common Vulnerability Scoring System (CVSS) is a free and open industry standard for assessing the severity of computer system security vulnerabilities. CVSS attempts to

assign severity scores to vulnerabilities, allowing responders to prioritize responses and resources according to threat. Scores are calculated based on a formula that depends on several metrics that approximate ease of exploit and the impact of exploit.

A huge amount of vulnerabilities are identified each day. Vendors, decision makers, IT-professionals and end- users all require a common way to assess the severity for the vulnerabilities discovered. Because CVSS is vendor-independent, users need a consistent methodology across various systems without having to go through a series of vendor-specific inconsistent scoring systems that are possibly inaccurate and more troublesome to maintain. Scores are calculated through a series of equations in a clear, consistent and easy to use way.

Research by the National Infrastructure Advisory Council (NIAC) in 2003/2004 led to the launch of CVSS version 1 (CVSSv1) in February 2005, with the goal of being "designed to provide open and universally standard severity ratings of software vulnerabilities". This initial draft had not been subject to peer review or review by other organizations. In April 2005, NIAC selected the Forum of Incident Response and Security Teams (FIRST) to become the custodian of CVSS for future development. Feedback from vendors utilizing CVSSv1 in production suggested there were "significant issues with the initial draft of CVSS".

Work on CVSS version 2 (CVSSv2) began in April 2005 with the final specification being launched in June 2007. Further feedback resulted in work beginning on CVSS version 3 in 2012, ending with CVSSv3.0 being released in June 2015.

2. The metrics baselines

CVSS consists of three metric groups: Base, Temporal, and Environmental.

The Base group represents the intrinsic qualities of vulnerability, the Temporal group reflects the characteristics of a vulnerability that change over time, and the Environmental group represents the characteristics of a vulnerability that are unique to a user's environment.

The Base metrics produce a score ranging from 0.0 to 10.0, which can then be modified by scoring the Temporal and Environmental metrics.

A CVSS score is also represented as a vector string, a compressed textual representation of the values used to derive the score.

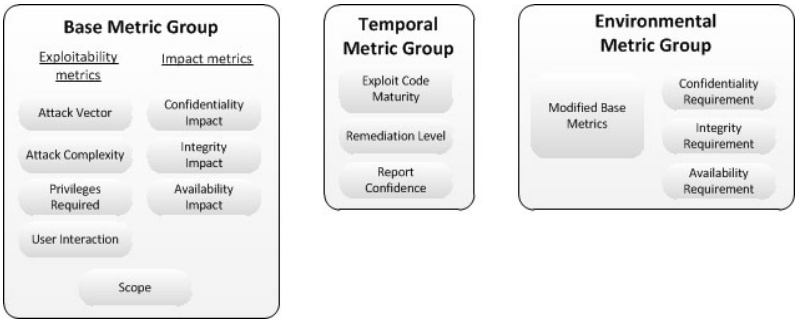

Figure 1 : CVSS v3 Metric Groups (Source: FIRST, 2015)

"The Temporal metric group reflects the characteristics of a vulnerability that may change over time but not across user environments." (FIRST, 2015a) For example, the existence of a simple-to-use exploit kit would increase the score, while the release of a patch will decrease it.

"The Environmental metric group represents the characteristics of a vulnerability that are relevant and unique to a particular user's environment." (FIRST, 2015a) These metrics let us examine security controls which may mitigate the consequences, as well as raise or lower the importance of a vulnerable system according to her business risk.

3. CVSS v2 problems

Although we have used CVSS v2 in our research as v3 was not accessible in the first two years and our former data were also evaluated only based on version 2. We have soon faced many flaws and weaknesses of the system. In this chapter we tend to enumerate some in order provide an all-round picture of the most common flaws of the scorings.

3.1. The "Partial" category

CVSS is using a three step evaluation in most cases: None/Partial/Complete. A single "Partial" category essentially ranges from "just a bit" to "almost everything but not quite". So it is quite ambiguous when we evaluate something with this category. There is a need for a Partial+ category (Eiram-Martin, 2013), to at least divide the distance between "None" and "Complete" into two parts.

3.2. Independent vulnerabilities?

When scoring separate vulnerabilities, they should be scored completely independently of each other and not take into account any interaction. Per CVSSv2 scoring tip #1:

"Vulnerability scoring should not take into account any interaction with other vulnerabilities. That is, each vulnerability should be scored independently." (Mell et. al., 2007)

Many exploits today do not utilize a sole vulnerability only to fully compromise a system, but combination of vulnerabilities instead. In the real world, there is no such a thing as vulnerability independency, and vulnerabilities can modify the severity of each other. Not recognizing this in the scoring system generates a significant limitation. As security mechanisms and vendor software becomes more mature, these attacks will be increasingly important.

3.3 The Zero score

There are rare cases when a vulnerability is scored as a 0.0 by CVSS v2 standards. There is no such thing as these vulnerabilities or weaknesses do not represent any threat at all. It can be a minor concern, but still, in some industries, in some special cases it can really do harm, so it should get maybe a low, but real score.

3.4. The open source libraries

Software complexity is rising steadily. Common applications nowadays are usually assembled from 50 or more open source products, integrating hundreds of thousands of lines of code from numerous developers who have never even met each other.

A vulnerability that causes a denial of service in a popular library, as this affects only the given library, will receive a relatively low score in CVSS. But what happens if that library is in use by an other software. That exact same vulnerability can have a considerably more severe impact depending on how the functionality is implemented.

So it is virtually impossible to give an exact score, because it is highly dependent on the software products within the system that utilize the same library.

This element can never really be assessed fully by any scoring system in my opinion.

3.5. Unknown impact

How to assess vulnerabilities where we do not know the details? CVSSv2 assumes that a

certain set of information is always available when scoring, although this is often not the case. Based on NVD's scoring customs, the the basic assumption is the worst case scenario" and score it based on the worst-case impact. If there are absolutely no details, this results in many categories a score of 10.0, even if in reality it should be significantly lower. This makes it more difficult for security professionals to accurately prioritize mitigation efforts.

The CVSS guidelines should clearly document how to deal with these cases; either not scoring if the required level of detail is not available, allowing an expert experience based scoring based, or assume a worst case impact and score according to that.

The examples mentioned above are only some of the many questions that have been raised during the research, during the use of CVSS v2 framework.

4. CVSS v3 differences – focusing on the base score

CVSS version 3.0 was released on June 10th 2015 after having been under development for 3 years. This process was overseen by the CVSS Special Interest Group (SIG) that consists of representatives of a broad range of industry sectors, from banking and finance to technology and academia.

There are many improvements introduced in version 3.0. The most important is the change of the base scores. Base score, although specialist often advise otherwise, is the main, and most times only score that really interest decision makers, so it is worth to look at it more deeply.

The newer version of CVSS has introduced a number of changes in the scoring system, that handles web application type vulnerabilities far more accurately. This is very important because of the excessive use of such components in todays' ICT-systems.

While the three metric groups: the Base Score, the Temporal Score and the Environmental Score have remained the same, new cathegories, such as Scope (S) and User Interaction (UI) were added, moreover some existing metrics such as Authentication (Au) has been changed to new ones such as Privileges Required (PR).

The Environmental Metrics group also saw an new addition with the Modified Base Metrics—allowing analysts to customize CVSS scores based on the host that has been affected in the analyst's organization, making it contextual when required to be.

The Access Vector has been a bit widened with the addition of the Physical form, which requires the attacker to physically access the affected resource or component. This is an important addition to the previous indicators that focused only on electronic access via the

network stack or adjacent networks. The Physical vector is different from the local option because, although a malicious code can be executed by a local user, the attacker does not need to be present.

A base metric called Scope was added to CVSS v3 to represent cases in which a vulnerability impacts a component other than the vulnerable component itself, such as a vulnerability in a Web server may affect the victim's Web browsers that is connected to it, or a vulnerability that breaks out of the sandbox that helps to contain it to a small set of permissions. If a vulnerability Scope is "Unchanged", the vulnerable component and the affected component are the same; that means, only the vulnerable server is affected. If the scope metric is "Changed", then like above, a vulnerable server also affects the browser for example.

Moreover, the Impact vectors were changed to show the difference, when sensitive systems or information are affected, such as usernames and passwords being exposed. Impact has been divided into several areas, such as confidentiality, or whether the vulnerability can allow an attack to access confidential data through actions like stealing an admin password (scored High) or access some restricted data or the amount of confidential data is limited (scored Medium).

Another component of Impact is Integrity, or the truthfulness of data, including modifying files protected by the impacted component. The Availability component of Integrity rates the potential interruption for the affected component; a High rating on Availability means that the attacker can fully deny sustained (while the attack is active) or persistent (even after the attack is complete) use of the affected component. (Cobb-Moore, 2015)

Likewise, a Low rating in Availability can mean that the component availability will be affected, but the attacker cannot control the impact.

4.1. Scope

One of the major development goals for CVSS version 3.0 was to reconsider the meaning of "Scope". In CVSS version 2, vulnerabilities are scored according to the "target host operating system". As a result, vulnerabilities which have severe impact on applications, but only limited impact to the target host operating system are rated as less severe ("partial impact"). For example, vulnerability, that allows an attacker to perform a Denial-of-Service (DoS) attack with impact to the availability to a vulnerable application. Even with a total take down of the targeted application this would result only a "Partial impact" rating for Availability metrics in CVSS v2. A "Complete impact"' rating for Availability can only be given if the host operating system can be taken down as well. This is clearly a problem for many applications and software vendors, like Oracle, SAP, etc.

To improve this issue, CVSS v3 have introduced "Authorization Scope", or simply "Scope" metric as a new element of the Base metric group. This new metric allows security professionals to assess the extent of a vulnerability with impact that reaches over the vulnerable component.

The definition of "Scope" contains a series of privileges defined by computing authorities (for example application, operating system, sandbox environment, etc.) which grant access to IT-resources like. files, CPU, memory, etc. Taking two discrete components which manage privileges to IT-resources one by one, these represent separate authorities. Taking the DoS attack example, a vulnerable application is managed by "Authority A," while the host Operating System is managed by "Authority B."

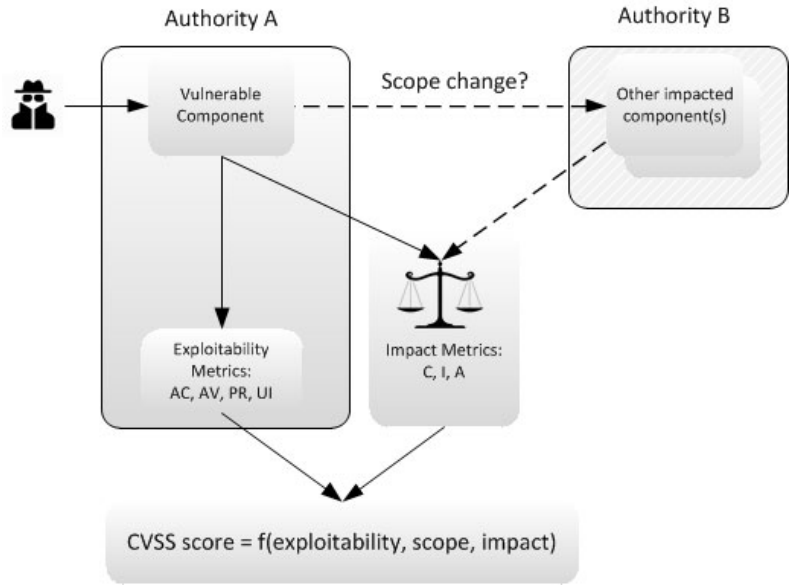

Figure 1: The change of scope (Source: FIRST, 2015b)

From Figure 1 it turns out, that Exploitability metrics aim is to assess the vulnerable component in an isolated way, while the Impact metrics are aiming the impacted component. When the vulnerable component is the same as the impacted component, there is no change of scope. However, a change of scope has happened when components outside the vulnerable component are impacted. In these cases, Confidentiality, Integrity and Availability impact metrics should generate a score to reflect the impact on either the vulnerable component, or the impacted component, depending on which is more severe.

Considering the DoS attack example, in CVSS v3, we can rate the Availability impact as 'High' (maximum score) to represent the real impact on the vulnerable application irrespective of its impact on Host Operating System. For cases where the Availability of the Host Operating System is also impacted, the increased severity is represented by the "Scope" metric category.

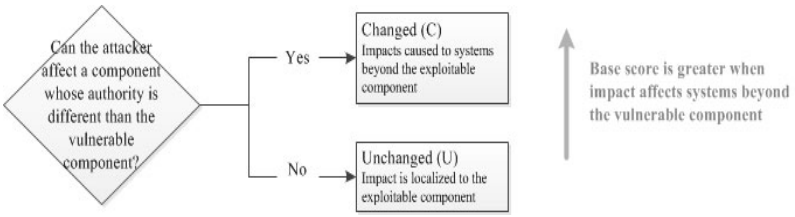

Figure 2: Scoring Rubic – The Scope (Source: FIRST, 2015b)

4.2. Impact

In version 3.0 the "Impact metric" (CIA - Confidentiality impact, Integrity impact and Availability impact) calculation represents the "reasonable final impact" caused by a successful exploit of the vulnerability. In CVSS version 2, the impact calculation only covers the "primary impact". The approach of CVSS version 3 is better, as it covers the bigger picture as well.

For example, a vulnerability that resulted a disclosure of administrator credentials was rated "partial" impact to Confidentiality and with no impact on Integrity or Availability in CVSS v2. Although a successful exploit in this case would result the attacker to obtain administrative privileges. In version 3.0, this vulnerability would be rated with "High" impact to CIA.

Also in CVSS v2, the possible values for CIA impact were 'None, Partial and Complete' which have been changed to "None, Low and High" in version 3.0. The new naming reflects the "degree of impact" caused by an attack. This is significant difference from the overall percentage of the systems impacted by an attack in CVSS v2. This change can be best illustrated via the 'Heartbleed' vulnerability (CVE-2014-0160). This 'buffer over-read' vulnerability which enabled an attacker to read up to 64 KB of victim's memory is rated as having a Base metric score of 5.1 (NLN|PNN) in CVSS v 2. A successful exploit can result in only some (up to 64 KB) amount of information disclosure. However the "Partial" rating calculated for "Confidentiality Impact" does not consider the sensitivity of the data (or information) which is disclosed. Since CVSS v2 methodology do not contain the sensitivity of data under attack, a vulnerability that results in disclosing information of lower sensitivity like version information of a web server, is also rated with the same Base metric

score of 5.1. Information of higher sensitivity like secret keys used for X.509 certificates, user names and passwords, emails, instant messages, business critical documents were disclosed in Heartbleed. In CVSS v3, 'Heartbleed' vulnerability has a Base score of 7.5 (NLNN|UHNN). The Confidentiality impact is rated as 'High' considering the sensitivity\criticality of the data under attack.

In the "IBM X-Force Threat Intelligence Quarterly, 2015 Q3" (IBM, 2015) IBM Security X-Force researchers looked at the 2008 DNS Kaminsky bug to see how the system builds up. For this special example, because the bug affects a DNS server, the scope goes beyond just the DNS server and can victimize an end user who connects to the server.

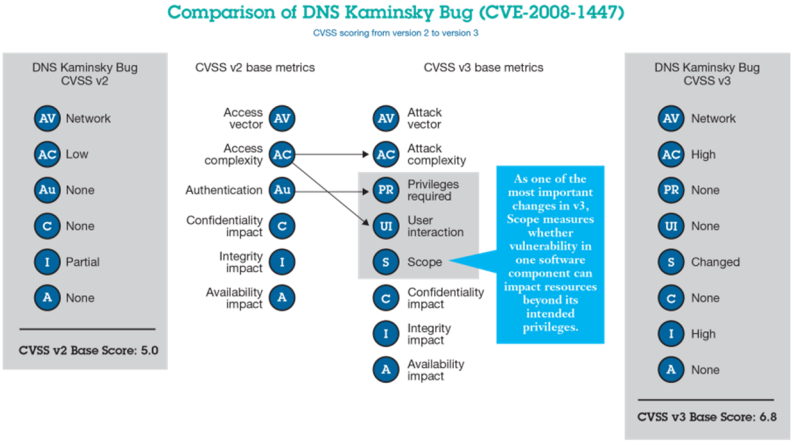

Figure 1: Comparison of vulnerability CVE-2008-1447 scoring according to CVSS v3 and v3 methodologies
Source: IBM X-FORCE R&D, 2015

Along with the publication of a scoring calculator, FIRST has provided a document full of sample vulnerabilities to help the users understand the scoring mechanism. (FIRST, 2016) Out of a total of 21 examples, two-thirds had their score rise, and half of those had their score rise by more than 1 point on the 10-point scale. Those with the biggest raise in score include the Kaminsky bug mentioned above, Heartbleed (up to 7.5 from 5), Juniper Proxy ARP DoS (CVE-2013-6014, up to a 9.8) and a SearchBlox XSS forgery bug (CVE-2015-0970, up to 8.8).

When Heartbleed appeared in 2014, the seriousness of the impact wasn't reflected in the 5.0 base score, and it is not surprising that its score, and those of many others, have increased. Those vulnerabilities, whose scores dropped, are also interesting. For example, Adobe Acrobat Buffer Overflow (CVE-2009-0658) dropped from 9.3 down to a 7.8 due to the limited scope of the vulnerability. The POODLE bug (CVE-2014-3566) dropped from

4.3 to 3.1, and specialists now consider, that only vulnerabilities over a 7.5 base score are the real critical ones, which have to be treated extremely seriously.

Only new vulnerabilities since the adoption of the v3 scheme will show the new score.

4.3. Attack Complexity

Access Complexity from CVSS v2 could mean two possible settings:

1. Any circumstance beyond the attacker's control that is a must in order for the vulnerability to be successfully exploited (for example a software race condition, or a specific configuration setting),
2. The requirement for human interaction (for example, requiring a user to click or open a link in case of phishing attack scenarios for example).

Because of these two quite different components, Access Complexity has been separated into two metrics in CVSS v3 - Attack Complexity (which embodies case 1), and User Interaction (which addresses the second option).

4.4. Privileges Required

"Privileges Required" is a new category, it replaces the "Authentication" score of CVSS v2. Instead of measuring the number of times an attacker must authenticate to a target system, Privileges Required captures the level of access required for an attacker to mount a successful attack. In CVSS v2, rating of "Single" for Authentication metric was unable to make aa difference whether an attacker is required to gain low level (for example Read-only) privileges or high level (for example Administrator) privileges to successfully exploit a vulnerability.

5. Migration from v2 to v3

In case a standard methodology is changing it is an acceptable need, to be able to convert the former scores into the new system through some kind of algorithm. In this case we do not have such possibility the calculation methods and the frame of the categories changed so much, that there is no mathematical function to convert the former scores into the new system. That is why NVD, for example, has marked out a transitional period, during which both the "old" and the new scores will be part of the common vulnerability database. Government and corporate analysts, decision models which have worked based on the CVSS v2 metrics when deciding, how serious a threat is, and what priority to give the possible solutions, have to completely rethink and remodel the decision –making criteria based on version 3.

The biggest and most painful change is concerning the base scores, as we have seen in the previous chapter. Most of the companies rely only on Base Metrics, which are provided by automated scanning tools or vulnerability databases. Based on the diagram, Base Metric is just the beginning. It acts as an input for Temporal Metrics and further to Environmental Metrics.

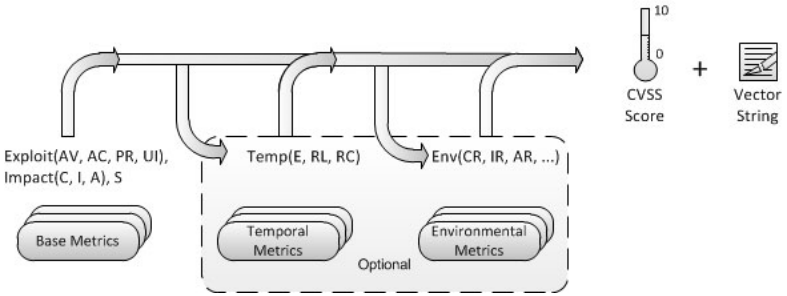

Figure1: Correlation between Base, Temporal and Environmental metrics Source: First.org, 2015

A reminder concerning the scores:

- Base Score - the intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments.
- Temporal Score - the characteristics of vulnerability that change over time but not among user environments.
- Environmental Score - the characteristics of a vulnerability that are relevant and unique to a particular user's environment.

The v2 and v3 scores are not even always comparable. Not just the transformation function is missing, but even the interpretation can be different due to the different building blocks for the given scores. For example, what could have been rated as partial in CVSS v2 for Confidentiality/Integrity/Availability would now be scored as High in CVSS v3. The effectiveness comes with v3 where the vulnerability is rated with respect to the impacted component, whereas in v2 it was with respect to the host application.

Security response depends a lot on vulnerability scoring system to prioritize actions. While CVSS base score is a good starting point, organizations really need to prioritize based on Environmental score specific to their network/solution being deployed. Even vulnerabilities with moderate or low-level base scores can have more damaging impact when Temporal or Environmental scores are considered.

6. Practical CVSS – CVSS and the IoT

IN the formerly published It-security trend analysis (Horváth et. al., 2016) we could see how smart devices and the Internet of Things is just changing every aspect of the ICT sector, and how serious the security problems are becoming in this area. So the question raises itself: Is the CVSS methodology, the scores themselves suitable for measuring the level of security and/or riskiness, of a software-intensive product or service on the market, for example a self-driving car?

More and more sources are actually finding vulnerabilities in vehicles as the on-board control and entertainment systems are becoming more and more complex, and standardized in order to be able to connect them to smartphones and web-applications, like the Jeep vulnerability, the Mobile Devices C4, Rolljam, Tesla, and other recent car-related vulnerabilities. (Klinedinst, 2015)

Reading about the topic I have met an interesting example vulnerability of an OBDII dongle, which can connect to any modern cars internal control software. The intended use is for your cell phone to communicate with the dongle over WIFI, and the dongle can use the phone's Internet connection if necessary. The device relays commands it receives over WIFI to the internal networks in the vehicle and vice versa. The problem is, that it uses a fixed and easy to note password: “password” to connect by default, and it can be changed through the devices mobile application, but it is not obligatory. This problem is definitely vulnerability, so let's try to score it based on the v2 methodology:

- Access Vector: The device's WIFI network is not accessible through the internet, so the attacker would need to be within WIFI range to connect. But it's not really a local vulnerability, since it requires an attacker to create a network connection. The closest option is “Adjacent Network”, (AV: A)
- Access Complexity: You can just Google the default password, so this is “Low”, actually very low. AC:L
- Authentication: There is no real authentication, so it is “None”. AU:N
- Confidentiality Impact: This vulnerability allows access to any data that is distributed by the dongle, including the signals of the CAN bus. However, it does not allow to access to data stored on the dongle. All in all it can be considered as “Partial”. C:P
- Integrity Impact: We can change any CANBUS message we want, but we cannot access the integrity of the firmware, so it is “Partial”. I:P
- Availability Impact: We can effectively DoS the dongle by flooding it with invalid packets, so this is “Complete”. A:C

The final numerical score is 7.3 and vectors of CVSS v2: AV:A/AC:L/AU:N/C:P/I:P/A:C.

Let us assess the same case based on the CVSS v3 metrics:

- Access Vector: Version 3.0 have added an additional option: Physical, but it's intended to indicate the need to actually touch the device. So it is still “Adjacent”. AV:A
- Attack Complexity: Access Complexity renamed, the new system has reduced the possible values to High or Low. Still “Low”. AC:L
- Privileges Required: Basically, this vector is the same as Authentication. Still “None”. PR:N
- User Interaction: This new vector differentiates attacks, whether an effective “help” from the user (click, open, install, etc.) is needed for the attacker to gain control. In this case no User Interaction is required, so the value is “None”. UI:N
- Scope: Another new element. It means "Does the scope of control the attacker gains, remain restricted to the device or app the vulnerability is present in?" In this case, no. While the vulnerability is in the OBDII dongle, the scope of control is the entire car. It's clearly “Changed”. S:C
- Confidentiality, Integrity, Availability: This is the same, the values have been renamed from None/Partial/Complete to None/Low/High.

The final numerical score is 8.8 in this case! The vectors are: AV:A/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:H in CVSS v3.

This is serious vulnerability: the attacker can steal the car, steal the contents of the car, steal data stored in it (i.e., contacts or GPS history), eavesdrop on the driver, or even cause the car to crash and possibly hurt or kill someone in it. Of course there are plenty other ways for an attacker to reach the same results.

7. Conclusions

There are numerous important conclusions. The pure, numeric scores from CVSS should not be used to compare vulnerabilities without the context. The individual metrics have to be examined to better understand the potential impact. Temporal and Environmental metrics should also be included, that were omitted here to keep this comparison at a manageable level.

Second CVSS metrics are much more suitable to assess the risks of software products running in some kind of IT-environment, than real life products that are out in the street or in homes and offices. Why is it so? Let us just consider a few things:

- Access vector: In case of software products it really counts that whether the can be accessed through the internet or the attacker has to get in a certain proximity

(Adjacent Network). If your IT is closed in a secure server room or farm, it makes you feel secure, that although, there is a vulnerability, the exploit requires physical proximity or contact, which is highly impossible due to the physical security. But a car, a smartwatch, a smart door lock etc. is not closed away in a secure room somewhere, but it is out there, where virtually anyone can get into the proximity.

- Attack complexity: although the CANBUS system can be easily accessed through this little device, mentioned above, but then what. Reverse engineering the specialized systems of the car are extremely difficult. And of course an Opel exploit is not compatible with a VW one. Still there can be ready packages by brands and types, but still, it is something different. The access protocols are standardized, but the internal systems are highly not.

- Scope: it is very important in this case, that this attack can affect not only the device but the car itself. Still, the weight of this component is not high enough, to make any difference. If we change Scope from “Changed” to “Unchanged”, we get down from 8.8 to 8.3. Not a big difference in scores, but in real life it is life or death: the whole vulnerability becomes indifferent if it cannot affect the car itself.

- Confidentiality, Integrity, and Availability: how to interpret these in this case and how to compare? How to compare a human life for example, with the loss of the confidentiality of company databases? Many other point affect these as well, for example the environmental issues: how strong is that car, how fast it is travelling, how well is the physical security of the passenger cell, these aspects will make a difference in case of a successful attack. The potential human death cannot be described through these metrics at all, as the most severe outcome possible ever.

So CVSS is extremely useful, but not for measuring all-round safety, not even in case of software intensive “smart” technologies. There are alternative means of risk assessment in these cases: the FIPS 199 "potential impact" levels, he ASIL standard (ISO-26262), which attempts to measure the risks of faults in cars, a more common version is the Safety Integrity Level (SIL - IEC-61508), it can be used in case of vehicles, power plants, and many other things as well, etc. These methods also have their weaknesses as these assume technology failures and disasters rather than intentional attacks as the cause.

IoT would need something combined: something not so industry specific, like the ones above, and more IT-centered, like CVSS, but not completely.

Maybe the next generation of CVSS will give answers to these questions as well, to measure "cyber-physical vulnerabilities". (Klinedinst, 2015) CVSS is a living methodology evolving with technology and the new challenges of security. As internet of thing is right here at our door, the next phase of CVSS and other risk-assessment methodologies will have to cover this area, the practical product and service side as well.

Acknowledgement

Preparation of this study is supported by "The effects of IT and network vulnerabilities on economy and society" no. PD-109740 from National Research, Development and Innovation Office - NKFIH.

References

[1] Abraham, Renchie Joan, 2015: Changes to CVSS in version 3.0, SAP Community Network – Security, 15 April 2015 http://scn.sap.com/community/security/blog/2016/04/15/changes-to-cvss-in-version-30

[2] Brigjaj, Juxhin Dyrmishi, 2016: What's new in CVSS version 3, Acunetix, 29 Ferbruary 2016, http://www.acunetix.com/blog/articles/whats-new-in-cvss-version-3/

[3] Cobb, Pamela – Moore, Scott 2015: Markdown: Designer Vulnerabilities Get a Fresh CVSS v3 Look, IBM X-FORCE and Security Intelligence, 08. September 2015, https://securityintelligence.com/markdown-designer-vulnerabilities-get-a-fresh-cvss-v3-look/

[4] Duncan, Jim 2015: CVSSv2 to v3; An Adolescent Standard Enters Adulthood, Juniper Networks Inc., 08. October 2015, http://www.triangleinfosecon.com/wordpress/wp-content/uploads/2015/12/CVSSv2-to-v3-An-Adolescent-Standard-Enters-Adulthood-Jim-Duncan-2015-10-08.pdf

[5] Eiram, Carsten – Martin, Brian, 2013: An Open Letter to FIRST - The CVSSv2 Shortcomings, Faults, and Failures Formulation, Risk Based Security - Open Security Foundation, 27 February 2013 https://www.riskbasedsecurity.com/reports/CVSS-ShortcomingsFaultsandFailures.pdf

[6] FIRST, 2015a: Common Vulnerability Scoring System v3.0: Specification Document (v1.7), FIRST.Org Inc. https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf

[7] FIRST, 2015b: Common Vulnerability Scoring System v3.0: User Guide (v1.4), FIRST.Org Inc. https://www.first.org/cvss/cvss-v30-user_guide_v1.4.pdf

[8] FIRST, 2016: Common Vulnerability Scoring System v3.0 Examples, FIRST.Org Inc., July 2016, https://www.first.org/cvss/cvss-v30-examples_v1.3.pdf

[9] IBM, 2015: IBM X-Force Threat Intelligence Quarterly, 3Q 2015, IBM Corporation, IBM Security, August 2015, http://public.dhe.ibm.com/common/ssi/ecm/wg/en/wgl03086usen/WGL03086USEN.PDF

[10] Klinedinst, Dan J., 2015: CVSS and the Internet of Things, Carnegie Mellon University Software Engineering Institute CERT, 02 September 2015, https://insights.sei.cmu.edu/cert/2015/09/cvss-and-the-internet-of-things.html

[11] Mell, Peter - Scarfone, Karen – Romanosky, Sasha, 2007: A Complete Guide to the Common Vulnerability Scoring System Version 2.0, National Institute of

Standards and Technology - Carnegie Mellon University – First.org, June 2007, https://www.first.org/cvss/cvss-v2-guide.pdf

[12] Sidhan, Britto, 2016: Uncover CVSS v3.0 Potential, Schneider Electric, 15 April 2016, https://www.linkedin.com/pulse/uncover-cvss-v30-potential-britto-sidhan

[13] Toomey, 2012: Patrick: CVSS – Vulnerability Scoring Gone Wrong, Neohapsis Labs, 25 April 2012, https://labs.neohapsis.com/2012/04/25/cvss-vulnerability-scoring-gone-wrong/