The Common Vulnerability Scoring System vs. Rock Star Vulnerabilities: Why the Discrepancy?

Doudou Fall and Youki Kadobayashi
Laboratory for Cyber Resilience, Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara, Japan

Keywords: CVSS Scores, Well-known Vulnerabilities.

Abstract: Meltdown & Spectre came as natural disasters to the IT world with several doomsday scenarios being professed. Yet, when we turn to the de facto standard body for assessing the severity of a security vulnerability, the Common Vulnerability Scoring System (CVSS), we surprisingly notice that Meltdown & Spectre do not command the highest scores. We witness a similar situation for other rock star vulnerabilities (vulnerabilities that have received a lot of media attention) such as Heartbleed and KRACKs. In this manuscript, we investigate why the CVSS 'fails' at capturing the intrinsic characteristics of rock star vulnerabilities. We dissect the different elements of the CVSS (v2 and v3) to prove that there is nothing within it that can indicate why a particular vulnerability is a rock star. Further, we uncover a pattern that shows that, despite all the beautifully elaborated formulas, magic numbers and catch phrases of the CVSS, there is still a heavy presence human emotion into the scoring as rock star vulnerabilities that were exploited in the wild before being discovered tend to have a higher score than those that were discovered and responsibly disclosed by security researchers. We believe that this is the principal reason of the discrepancy between the scoring and the level of media attention as the majority of 'modern' high level vulnerabilities are introduced by security researchers.

1 INTRODUCTION

In the cybersecurity world, we are gratified from time to time with security vulnerabilities that are so popular that even people from other professional and academic domains are aware of them; those vulnerabilities are what we denominate the rock star vulnerabilities in this manuscript. We qualify those vulnerabilities as rock stars because they are heavily covered by general news outlets, not only specialized ones. For instance, Meltdown (Lipp et al., 2018) & Spectre (Kocher et al., 2018) were covered as if they were natural disasters to the cybersecurity world and by extension the information technology word with many doomsday scenarios being professed. After witnessing such coverage, it is only normal to wonder how exactly dangerous those types of vulnerabilities are. One of the most accepted metrics for assessing the severity of a vulnerability is the Common Vulnerability Scoring Score (CVSS) [(Mell et al., 2007), (FIRST, 2015)]. Based on their popularity, one might think that all rock star vulnerabilities have the highest scores in the CVSS (which is 10) but it is surprisingly not the case. Indeed, recent high-level vulnerabilities such as Meltdown & Spectre and KRACKs (Vanhoef et al., 2010), which all subject-matter experts unanimously agree that they are disastrous, have CVSSv2 scores of 4.7, 4.7 – 4.7, and 5.4 – 2.9 - 2.9 – 2.9 – 2.9 – 5.8 – 5.4 – 5.4 – 2.9 – 2.9 respectively. This paper is not the first critical work of the CVSS. Many other researchers have finger pointed the shortcomings of the CVSS in different areas but, to the best of our knowledge, we are the first to investigate the discrepancy that exist between the scores of rock star vulnerabilities and their level of popularity. Despite being considered as unavoidable in vulnerability assessment, the CVSS's legitimacy is disputed from the industry to the academia. CVSS version 3 was introduced to correct the flaws of the version 2 but it does not solve our problem as the vulnerabilities we used as example above have the same *Medium* to *Low* scores. In this manuscript we first create a dataset of rock star vulnerabilities and investigate what actually makes them rock stars by dissecting the CVSS. We show that there is nothing within the CVSS that can indicate why a vulnerability becomes popular. We

Fall, D. and Kadobayashi, Y.
The Common Vulnerability Scoring System vs. Rock Star Vulnerabilities: Why the Discrepancy?.
DOI: 10.5220/0007387704050411
In *Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP 2019)*, pages 405-411
ISBN: 978-989-758-359-9

find an interesting pattern that might explain the discrepancy between the scores and the popularity of certain vulnerabilities: vulnerabilities exploited in the wild before being discovered tend to have higher scores than vulnerabilities that are discovered and responsibly disclosed by security researchers. This small discovery speaks volume about the fact that, despite all the formulas, magic numbers and catch phrases of the CVSS, human emotion is involved in the scoring which is another argument that could be used against the scientific aspect of computer security research. We believe that a good vulnerability assessment framework should be independent of human emotion and should take into account the analysis of the experts i.e., an artificial intelligence powered vulnerability assessment framework is primordial for the modern cybersecurity world. The remainder of the paper is structured as follows: Section 2 introduces important details of the CVSS. In Section 3, we peruse over the related work. Section 4 constitutes our main contribution, therein we discuss the shortcomings of the CVSS concerning rock star vulnerabilities. In Section 5, we discuss the advantages and limitations of our work before concluding the paper in Section 6.

2 BACKGROUND: CVSS

In this section, we perform a brief analysis of the CVSS, which is necessary to comprehend the subsequent sections.

We first talk about the National Vulnerability Database (NVD) (NVD, 2018) which is a publicly available database for computer-related vulnerabilities. It is a property of the United States (US) government, which manages it throughout the computer security division of the U.S. National Institute of Science and Technology (NIST). The NVD is also used by the U.S. government as a content repository for the Security Content Automation Protocol (SCAP). The primary sources of the NVD are as follows: Vulnerability Search Engine (Common Vulnerability Exposure (CVE) and CCE misconfigurations), National Checklist Program (automatable security configuration guidance in XCCDF and OVAL), SCAP and SCAP compatible tools, Product dictionary (CPE), Common vulnerability Scoring System for impact metrics, and Common Weakness Enumeration (CWE). The Common Vulnerability Scoring System (CVSS) is a vendor-neutral open source vulnerability scoring system. It was established to help organizations efficiently plan their responses regarding security vulnerabilities. Currently, the people at the MITRE Corporation are using both CVSSv2 (Mell et al., 2007) and CVSSv3 (FIRST, 2015) to score the vulnerabilities. As all the vulnerabilities have not been yet evaluated with CVSSv3, most of our work relates to CVSSv2. The CVSS is comprised of three metric groups classified as base, temporal, and environmental. The base metric group contains the quintessential characteristics of a vulnerability. The temporal metric group is used for non-constant characteristics of a vulnerability, and the environmental metric group defines the characteristics of vulnerabilities that are tightly related to the environment of the users. We want our analysis to be sufficiently generic so that it can be utilized at any time by any organization. For that reason, we opted to make exclusive use of the base metric group which provides the constant characteristics of a vulnerability. In doing so, the vulnerabilities will not change in relation to either time or organization. Consequently, the temporal and environmental metric groups do not feature prominently in our research. In CVSSv2, the base metric group regroups essential metrics that are used to compute the score of a vulnerability: Access Vector (AV) is the metric reflecting how the vulnerability is exploited; Access Complexity (AC) is the metric that defines how difficult it is to exploit a vulnerability once an attacker has gained access to the target system; Authentication (Au) is the metric that reflects the number of times an attacker must authenticate to a target in order to exploit a vulnerability; Confidentiality Impact (C) is the metric that measures the impact on confidentiality of a successfully exploited vulnerability; Integrity Impact (I) is the metric that measures the impact to integrity of a successfully exploited vulnerability; and Availability Impact (A) is the metric that measures the impact to availability of a successfully exploited vulnerability. In CVSSv3, the exploitability metrics are replaced by: Attack Vector (AV) and Attack Complexity (AC) which play similar roles as AV and AC in CVSSv2; Privileges Required (PR) determines the level of privileges an attacker must have to be able to exploit the vulnerability, User Interaction (UI) which “captures the requirement for a user, other than the attacker, to participate in the successful compromise of the vulnerable component,” and Scope (S) which is the ability for a vulnerability in one software component to impact resources beyond its means.

3 RELATED WORK

This section mostly comports articles that are critical of the CVSS in diverse ways but none of them tackled the CVSS as we did in this paper.

(Bozorgi et al., 2010) were among the first critiques of the CVSS. They argued about its weighting system and the broadness and subjectivity of the questions before proposing a severity assessing system based on machine learning. (Johnson et al., 2016) analysed whether the CVSS could be trusted for assessing the score of the vulnerabilities found in 5 databases: NVD, IBM X-force, OSVDBD, CERT-VN, and Cisco. Using Bayesian analysis and after carefully selecting priors, they concluded that the CVSS was in fact robust and trustworthy as it has a consistency throughout the databases with each having their own scoring team. The difference with our paper is that these authors did not question semantic operations but rather whether the CVSS have a standard scoring ability i.e. if we give a vulnerability characteristic to anyone, they could generate a score that is similar to the one that a subject-matter expert would have generated. Nuthan Manaia and Andrew Meenely studied the disconnect that exist between the CVSS and bug bounties (Munaia and Meenely, 2016). After their investigations they found a weak correlation between the two with a Spearman correlation of 0.34, "with the CVSS being more likely to underestimate bounty." They investigated the discordance of their measurements and found that the bounty criteria is geared towards code execution and privilege escalation whereas the CVSS has a generic evaluation for all types of vulnerabilities. Furthermore, the relative lack of academic attention may explain the shortcomings of the CVSS as it is quite difficult to find a paper that address these issues in the so-called top 4 security and privacy conferences: ACM CCS, Oakland, USENIX Security, and NDSS. Although, while acknowledging its shortcomings, some of those researchers do not hesitate to use it to make their point come across. Franck Li and Vern Paxon epitomized that fact in (Li and Paxon, 2017): "While the CVSS standard is imperfect ..., it provides one of the few principled ways to characterize vulnerability risk and potential impact. We use this score as is, however acknowledging the difficulties in objectively assessing vulnerability severity."

4 CVSS vs. ROCK STAR VULNERABILITIES

In this section, we explain our dataset collection methodology, examine what makes a vulnerability a rock star and expose the most interesting finding of our study.

4.1 Rock Star Vulnerabilities Collection

We mainly made use of our own knowledge and of Google search to find the most notorious security vulnerabilities of all time. We used keywords such as "worst vulnerabilities of computer history," "timeline of computer worms," "computer viruses hall of fame." At the end, based on the results we obtained (Vijayan, 2016) (Zdnet.com, 2014) (Norton.com, 2016) (Jamaluddin, 2017) (Wikipedia.org, 2017) (Ward, 2017), we carefully selected the vulnerabilities represented in Table 1 with their CVSS details. There are several honourable mentions such as the Morris worm which does not feature in our research because of the absence of a CVE-ID. We understand that people could argue with our selection of vulnerabilities but after surveying many specialised and less specialised websites that we found through our web search, we believe that the content of Table 1 is accurate. We will not give details of each of the vulnerabilities, instead we invite the reader to consult the many resources that are available on the Internet.

4.2 What Makes a Vulnerability a Rock Star

The most obvious results from Table 1 is that, despite the fact that all these vulnerabilities are deemed as very dangerous, they do not all feature the highest scores in the CVSS (which is 10) pushing us to seriously wonder: what makes a vulnerability a rock star? Only *Shellshock* vulnerabilities, *Conficker* and one of the vulnerabilities of *Code red* have the maximum scores in CVSSv2. CVSSv3 was introduced to correct the shortcomings of CVSSv2. Unfortunately, not all the vulnerabilities have CVSSv3 scores but for the ones that have it, we observe higher scores than CVSSv2 except for WannaCry vulnerabilities. However, CVSSv3 does not also provide the maximum scores for the rock star vulnerabilities for which it is eligible. We have better qualitative results as 55% of the concerned vulnerabilities are

Table 1: Rock Star Vulnerabilities and their CVSS details.

Name	CVE-ID	v2	Clv2	Iv2	Ev2	v3	Clv3	Iv3	Ev3
Meltdown	CVE-2017-5754	4.7	Medium	6.9	3.4	5.6	Medium	4.0	1.1
Spectre	CVE-2017-5753	4.7	Medium	6.9	3.4	5.6	Medium	4.0	1.1
	CVE-2017-5715	4.7	Medium	6.9	3.4	5.6	Medium	4.0	1.1
KRACKs	CVE-2017-13077	5.4	Medium	6.4	5.5	6.8	Medium	5.2	1.6
	CVE-2017-13078	2.9	Low	2.9	5.5	5.3	Medium	3.6	1.6
	CVE-2017-13079	2.9	Low	2.9	5.5	5.3	Medium	3.6	1.6
	CVE-2017-13080	2.9	Low	2.9	5.5	5.3	Medium	3.6	1.6
	CVE-2017-13081	2.9	Low	2.9	5.5	5.3	Medium	3.6	1.6
	CVE-2017-13082	5.8	Medium	6.4	6.5	8.1	High	5.2	2.8
	CVE-2017-13084	5.4	Medium	6.4	5.5	6.8	Medium	5.2	1.6
	CVE-2017-13086	5.4	Medium	6.4	5.5	6.8	Medium	5.2	1.6
	CVE-2017-13087	2.9	Low	2.9	5.5	5.3	Medium	3.6	1.6
	CVE-2017-13088	2.9	Low	2.9	5.5	5.3	Medium	3.6	1.6
WannaCry	CVE-2017-0144	9.3	High	10	8.6	8.1	High	5.9	2.2
	CVE-2017-0145	9.3	High	10	8.6	8.1	High	5.9	2.2
POODLE attack	CVE-2014-3566	4.3	Medium	2.9	8.6	6.8	Medium	4	2.2
	CVE-2014-8730	4.3	Medium	2.9	8.6	N/A	N/A	N/A	N/A
Heartbleed	CVE-2014-0160	5.0	Medium	2.9	10	N/A	N/A	N/A	N/A
Shellshock	CVE-2014-6271	10	High	10	10	N/A	N/A	N/A	N/A
	CVE-2014-7169	10	High	10	10	N/A	N/A	N/A	N/A
	CVE-2014-7186	10	High	10	10	N/A	N/A	N/A	N/A
	CVE-2014-7187	10	High	10	10	N/A	N/A	N/A	N/A
	CVE-2014-6277	10	High	10	10	N/A	N/A	N/A	N/A
	CVE-2014-6278	10	High	10	10	N/A	N/A	N/A	N/A
Stuxnet	CVE-2010-2568	9.3	High	10	8.6	N/A	N/A	N/A	N/A
	CVE-2010-2729	9.3	High	10	8.6	N/A	N/A	N/A	N/A
	CVE-2010-2743	7.2	High	10	3.9	N/A	N/A	N/A	N/A
	CVE-2010-3338	7.2	High	10	3.9	N/A	N/A	N/A	N/A
	CVE-2010-2772	6.9	Medium	10	3.4	N/A	N/A	N/A	N/A
Kaminsky Bug	CVE-2008-1447	5.0	Medium	2.9	10	N/A	N/A	N/A	N/A
VENOM	CVE-2015-3456	7.7	High	10	5.1	N/A	N/A	N/A	N/A
SQL Slammer	CVE-2002-0649	7.5	High	6.4	10	N/A	N/A	N/A	N/A
Conficker (Stuxnet)	CVE-2008-4250	10	High	10	10	N/A	N/A	N/A	N/A
Blaster	CVE-2003-0352	7.5	High	6.4	10	N/A	N/A	N/A	N/A
Sasser	CVE-2003-0533	7.5	High	6.4	10	N/A	N/A	N/A	N/A
Code red	CVE-2001-0500	10	High	10	10	N/A	N/A	N/A	N/A
	CVE-2001-0506	7.2	High			N/A	N/A	N/A	N/A
Welchia	CVE-2003-0109	7.5	High	10	3.9	N/A	N/A	N/A	N/A
Nimda	CVE-2000-0884	7.5	High	6.4	10	N/A	N/A	N/A	N/A
	CVE-2001-0154	7.5	High	6.4	10	N/A	N/A	N/A	N/A

Legend : v2 = CVSSv2 base score, Clv2 = qualitative classification for CVSSv2 base score, Iv2 = impact sub score of CVSSv2, Ev2 = exploitability sub score of CVSSv2, v3 = CVSSv3 base score, Clv3 = qualitative classification of CVSSv3 base score, Iv3 = impact sub score of CVSS v3, Ev3 = exploitability sub score of CVSSv3

classified as *High*, 30% as *Medium*, and 15% as *Low* in CVSSv2. Whereas in CVSSv3, 19% of the vulnerabilities are *High*, and 81% are *Medium*. But the only fact that all of them are not classified as *High* is an issue. As we cannot have a definitive answer from the overall score, we venture out to the *impact* and *exploitability* scores of the vulnerabilities. Figure 1 shows that there is no clear indication of which one makes a vulnerability a rock star in CVSSv2. One would argue that when experts and general news outlets journalists write about some high level vulnerabilities, it is to talk about

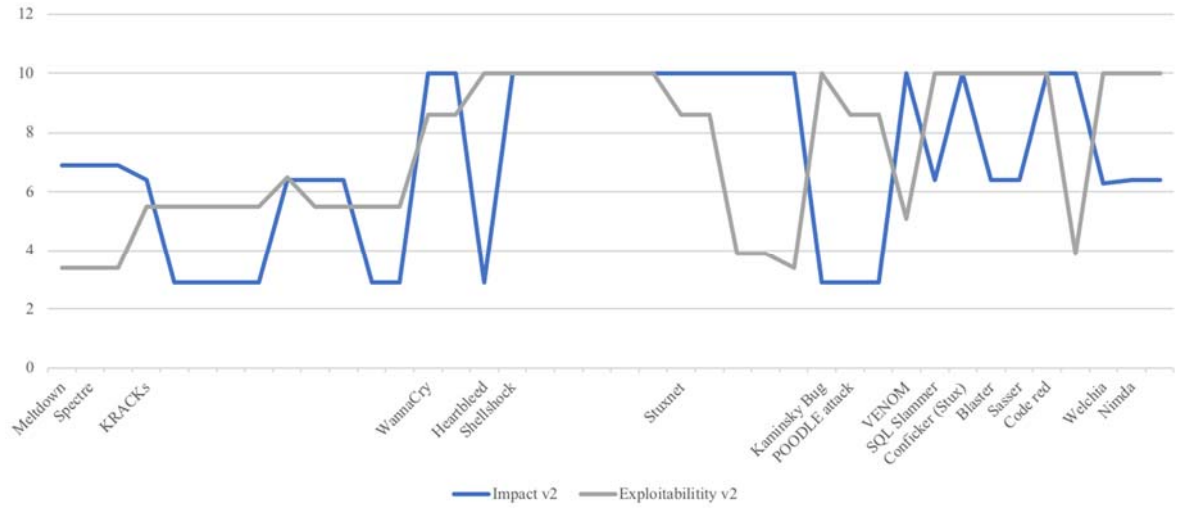

Figure 1: Relationship between the exploitability and the impact scores of CVSSv2.

their damages or impacts i.e., we could neglect the exploitability scores. However, the same observation with overall scores applies for the impact scores: we do not have the max scores for rock star vulnerabilities. For CVSSv3, Fig. 2 somehow follows the assertion we made earlier as all impact scores are higher than their respective exploitability counterpart but they still do not feature the maximum scores.

4.3 Wilderness (or Not) of Rock Star Vulnerabilities

The most important result we found is that when we look closely into the data, we realize that vulnerabilities that were exploited in the wild before being discovered tend to have a higher score than vulnerabilities that were discovered and responsibly reported by security researchers. The only reason we can find to explain this phenomenon is that, despite all the magic numbers and the beautifully crafted formulas, the scoring in the CVSS still depends on human emotion. For instance, all the subject-matter experts in cybersecurity agree that Meltdown & Spectre are devastating vulnerabilities yet they have *Medium* to *Low* scores within CVSS. Some people might argue that the reason of the low scores is that the vulnerabilities are hard to exploit. The counterargument is that despite the vulnerability being hard to exploit, in practice, whenever an exploit is made available, through *https://www.exploit-db.com* for instance, it is not hard anymore. Additionally, even if we accept their hardness of exploitation, we all agree that they would have a big impact once they are exploited so why their impact scores are not the maximum (10). (Vanhoef et al., 2016) discovered serious weaknesses in WPA2, a protocol that secures all modern Wi-Fi networks. A miscreant who is within the vicinity of a victim could exploit these weaknesses. Despite these being vulnerabilities of the WPA2 protocol itself, the scores of the vulnerabilities range from *Low* (2.9*6) to *Medium* (5.4*3, 5.8). The same analysis could be made for the POODLE attack (4.3, 4.3) (Möller et al., 2014), Heartbleed (5.0) (Durumeric et al., 2014), and the Kaminsky bug (5.0) (Kaminsky, 2008). The common denominator of these vulnerabilities is that they were discovered and responsibly reported by security researchers. We conclude that this is the main reason of the discrepancy between the actual scores and the level of popularity of the said vulnerabilities. Among the vulnerabilities that are exploited in the wild before being discovered only one, Stuxnet CVE-2010-2772, is classified as *Medium*. The remaining vulnerabilities – *Nimda*, *Welchia*, *Code red*, *Sasser*, *Blaster*, *Conficker*, *SQL Slammer*, *VENOM*, the other CVEs of *Stuxnet*, *Shellshock*, and *WannaCry* – are classified as *High*.

5 DISCUSSION AND FUTURE WORK

The main weakness of our paper is the limited number of vulnerabilities we consider in the dataset. Although our aim is to only examine the scoring behaviour of well-known vulnerabilities, one might argue that these flaws only apply to those types of

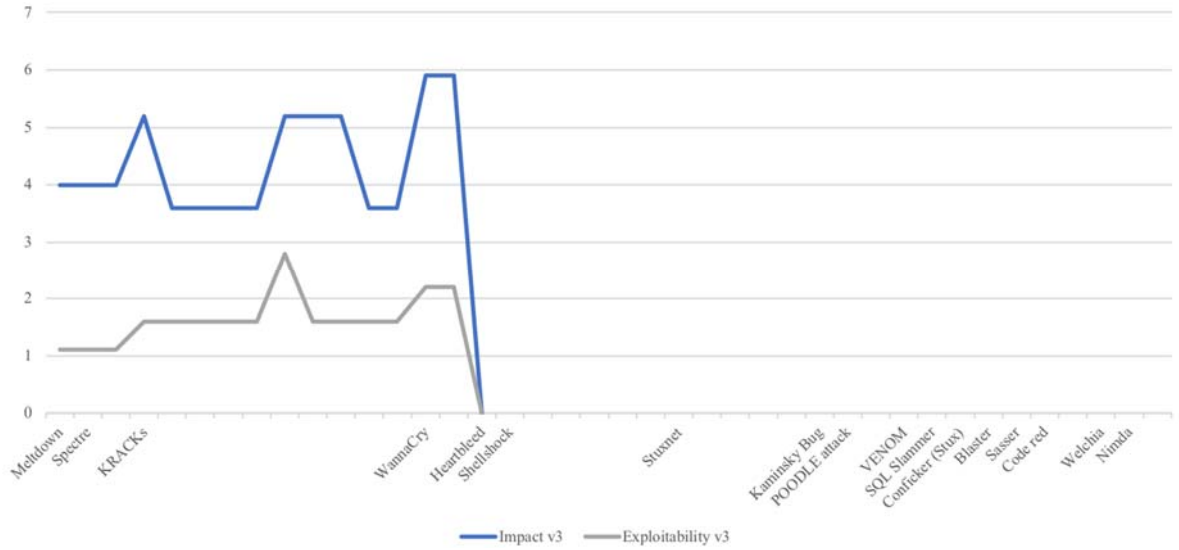

Figure 2: Relationship between the impact and exploitability scores of CVSSv3.

vulnerabilities therefore a study with a larger dataset is needed in order to confirm our findings.

One can actually argue that vulnerabilities that have been exploited in the wild have more accurate scoring than those discovered by security researchers. In fact, with the former, experts already know what the vulnerabilities are capable of thus, they may have a better judgement. Whereas with the latter, experts are mostly speculating based on theoretical descriptions of the vulnerabilities. We are not suggesting that we should wait for a vulnerability to be exploited to evaluate its severity but, in general, actual damages are more accurate than forecasts.

Using qualitative instead of quantitative values: Looking at the dataset, we realize that the qualitative values (LOW, MEDIUM, HIGH, CRITICAL) give more information about the severity of the vulnerabilities than the numerical values. We contend that the CVSS should drop numerical evaluations as they tend to add confusions because one cannot tell the real difference between a vulnerability that has a score of 9.1 and another that has 9.8 other than they are both critical (based on CVSSv3 evaluation).

We also noticed that there is a sort of systematic mapping of the scores between CVSSv2 and CVSSv3. Indeed, as we can see in Table 1, the same CVSSv2 scores always have the same CVSSv3 scores. We believe that this is another weakness of the CVSS in the general as in a perfect vulnerability assessment system, each vulnerability should be assessed in a unique way and that is not the case currently.

The debate of this research concerns popularity vs. severity. Should a popular vulnerability automatically command the highest score in the CVSS? It is obvious that the answer to that question is negative. It is well-known that the general population might have a more alarming reaction to a certain situation than experts in the field. We are probably having the same phenomenon happen in vulnerability assessment as, usually, the journalists or bloggers who bring those vulnerabilities to a certain level of stardom are not as educated on the field as those who assess the severity of the vulnerabilities. But the fact that vulnerabilities exploited in the wild have higher scores than those that are responsibly disclosed means the popularity of the vulnerabilities has an influence on the way their severities are assessed. In contrast, several vulnerabilities have the highest scores in the CVSS but very few people know about them.

One might argue that recent well-known vulnerabilities have more accurate scores than earlier ones. Around the end of the 90s beginning of the 2000s, the phenomenon of vulnerabilities being exploited in the wild was relatively new and that may have played a big role on their severity assessment because the vulnerabilities were not technically sophisticated. Whereas recent rock star vulnerabilities tend to be more sophisticated.

There is an ongoing argument that, before attempting to solve a problem with artificial intelligence (AI), human beings should first be able to perfectly deal with the issue. Vulnerability severity assessment falls in that category of problems, we have yet to figure out how to do it right.

Vulnerabilities such as Spectre (speculative execution), which many people have deemed safe for many years, will become prevalent in the cyberspace. Nevertheless, we believe that, first and foremost, a modern vulnerability severity assessment framework should not be rigid and should take into account many evolving factors.

6 CONCLUSIONS

In this paper, we analysed what we call rock star vulnerabilities which we identified after a thorough and rigorous selection procedure. We showed that despite the level of stardom of those vulnerabilities, only one of them has the maximum numerical score in the CVSS. Further analysis showed that there is not a single metric in the CVSS that capture the real state of those vulnerabilities. Additionally, we found that rock star vulnerabilities that have been discovered after exploitation and vulnerabilities that were discovered before exploitation are rated differently with the latter having lower scores than the former as if the evaluators were reacting to the amount of damage that the vulnerabilities have caused in the real world. In conclusion, we believe that as a community we have failed to propose a standard that succeeds to capture all the facets of a vulnerability in order to give it the score it deserves. We should devise a vulnerability scoring system that is immune from human emotion and yet can capture all the facets of a vulnerability.

REFERENCES

Mell, P., Scarfone, K. and Romanosky, S., 2007, June. A complete guide to the common vulnerability scoring system version 2.0. In Published by FIRST-Forum of Incident Response and Security Teams (Vol. 1, p. 23).

FIRST, CVSS (SIG) members, 2015. Common Vulnerability Scoring System v3.0: Specification Document. In Published by FIRST. Available at https://www.first.org/cvss/cvss-v30-specification-v1.8.pdf [Accessed 27 Sept. 2018].

NVD, 2018. National Vulnerability Database. [online] Available at https://nvd.nist.gov [Accessed on 29 Sept. 2018].

Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Mangard, S., Kocher, P., Genkin, D., Yarom, Y. and Hamburg, M., 2018. Meltdown. arXiv preprint arXiv:1801.01207.

Kocher, P., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M. and Yarom, Y., 2018. Spectre attacks: Exploiting speculative execution. arXiv preprint arXiv:1801.01203.

Vanhoef, M. and Piessens, F., 2016, August. Predicting, Decrypting, and Abusing WPA2/802.11 Group Keys. In USENIX Security Symposium (pp. 673-688).

Johnson, P., Lagerstrom, R., Ekstedt, M. and Franke, U., 2016. Can the Common Vulnerability Scoring System be Trusted? A Bayesian Analysis. IEEE Transactions on Dependable and Secure Computing, (1), pp.1-1.

Li, F. and Paxson, V., 2017, October. A large-scale empirical study of security patches. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 2201-2215). ACM.

Munaiah, N. and Meneely, A., 2016, November. Vulnerability severity scoring and bounties: Why the disconnect?. In Proceedings of the 2nd International Workshop on Software Analytics (pp. 8-14). ACM.

Bozorgi, M., Saul, L.K., Savage, S. and Voelker, G.M., 2010, July. Beyond heuristics: learning to classify vulnerabilities and predict exploits. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 105-114). ACM.

Vijayan, J., 2016. The 10 worst vulnerabilities of the last 10 years. [online] Darkreading.com. Available at https://www.darkreading.com/vulnerabilities---threats/the-10-worst-vulnerabilities-of-the-last-10-years/d/d-id/1325425 [Accessed 9 July 2018].

Zdnet.com, 2014. Before Heartbleed: Worst vulnerabilities ever. [online] Available at https://www.zdnet.com/pictures/before-heartbleed-worst-vulnerabilities-ever/ [Accessed 6 June 2018].

Norton.com, 2016. The 8 most famous computer viruses of all time. [online] Available at https://uk.norton.com/norton-blog/2016/02/the_8_most_famousco.html [Accessed 20 June 2018].

Jamaluddin, A., 2017. 10 most destructive computer viruses. [online] Hongkiat.com. Available at https://www.hongkiat.com/blog/famous-malicious-computer-viruses/ [Accessed 20 July 2018].

Wikipedia.org, 2017. Timeline of computer viruses and worms. [online] Available at https://en.wikipedia.org/wiki/Timeline_of_computer_viruses_and_worms [Accessed 27 August 2018].

Ward, M., 2017. WannaCry and the malware hall of fame. [online] BBC News. Available at https://www.bbc.com/news/technology-39928456 [Accessed 15 Sept. 2018].

Möller, B., Duong, T. and Kotowicz, K., 2014. This POODLE bites: exploiting the SSL 3.0 fallback. Security Advisory.

Durumeric, Z., Li, F., Kasten, J., Amann, J., Beekman, J., Payer, M., Weaver, N., Adrian, D., Paxson, V., Bailey, M. and Halderman, J.A., 2014, November. The matter of Heartbleed. In Proceedings of the 2014 conference on internet measurement conference (pp. 475-488). ACM.

Kaminsky, D., 2008. Black ops 2008: It's the end of the cache as we know it. *Black Hat USA*.