

A Historical and Statistical Study of the Software Vulnerability Landscape

Assane Gueye
Carnegie Mellon
University Africa
Kigali, Rwanda
assaneg@andrew.cmu.edu

Peter Mell
National Institute
of Standards and Technology
Gaithersburg MD, USA
peter.mell@nist.gov

Abstract—Understanding the landscape of software vulnerabilities is key for developing effective security solutions. Fortunately, the evaluation of vulnerability databases that use a framework for communicating vulnerability attributes and their severity scores, such as the Common Vulnerability Scoring System (CVSS), can help shed light on the nature of publicly published vulnerabilities. In this paper, we characterize the software vulnerability landscape by performing a historical and statistical analysis of CVSS vulnerability metrics over the period of 2005 to 2019 through using data from the National Vulnerability Database. We conduct three studies analyzing the following: the distribution of CVSS scores (both empirical and theoretical), the distribution of CVSS metric values and how vulnerability characteristics change over time, and the relative rankings of the most frequent metric value over time. Our resulting analysis shows that the vulnerability threat landscape has been dominated by only a few vulnerability types and has changed little during the time period of the study. The overwhelming majority of vulnerabilities are exploitable over the network. The complexity to successfully exploit these vulnerabilities is dominantly low; very little authentication to the target victim is necessary for a successful attack. And most of the flaws require very limited interaction with users. However, on the positive side, the damage of these vulnerabilities is mostly confined within the security scope of the impacted components. A discussion of lessons that could be learned from this analysis is presented.

Index Terms—Vulnerabilities, Statistics

I. INTRODUCTION

Understanding the landscape of software vulnerabilities is a key step for developing effective security solutions. It is difficult to counter a threat that is not well understood. Fortunately, there exist vulnerability databases that can be analyzed to help shed light on the nature of publicly published software vulnerabilities. The National Vulnerability Database (NVD) [1] is one such repository. NVD catalogs publicly disclosed vulnerabilities and provides an analysis of their attributes and severity scores using the Common Vulnerability Scoring System (CVSS) [2]. CVSS is used extensively by security tools and databases and is maintained by the international Forum of Incident Response and Security Teams (FIRST) [3].

The CVSS provides a framework for describing vulnerability attributes and then scoring them as to their projected severity. The attributes are metric values that are the input to a CVSS equation that generates the score. It is the vulnerability attribute descriptions (the metric values) that are of primary

interest to our work, although we also look at the raw scores. The use of CVSS by vulnerability databases provides a suite of low level metrics, encapsulated in a vector, describing the characteristics of each vulnerability. CVSS was initially released in 2005 [4], was completely revamped with version 2 (v2) in 2007 [5], and was updated with new and modified metrics in 2015 with the release of version 3 (v3) [6]. Note that a minor update version 3.1 was released in 2019 [7], but the changes therein do not affect our work. The software flaw vulnerability landscape was thoroughly analyzed in the scientific literature using v2 when it was first released [4], [8]–[13], but little work has been done since to evaluate changes to that landscape over time. Also in our literature survey, we did not find a single study that uses the updated and significantly modified v3 to understand the software vulnerability landscape.

In this paper, we use the CVSS v2 and v3 data provided by the NVD to undertake a historical and statistical analysis of the software vulnerabilities landscape over the period 2005 to 2019. More precisely, we conduct three studies analyzing the following:

- score distributions,
- metric value distributions,
- and relative rankings of the most frequent metric values.

For our first study, we analyze and compare the distributions of CVSS v2 and v3 scores as generated from the NVD data. We then compare the empirical distributions against the theoretical score distributions, assuming that all CVSS vectors are equally likely (which is not the case, but it is illustrative to evaluate the differences).

For our second study, we compute the distributions of the CVSS metric values (i.e., vulnerability characteristics) for each year. We then analyze the differences from 2005 to 2019 to determine if and how vulnerability characteristics change over time.

For our third study, we identify the most frequent metric values and analyze their relative rankings from 2015 to 2019. For each year and for both CVSS versions, we compute the values of the top 10 observed vulnerability metrics as well as their frequencies. We then generate parallel coordinates plots showing the values and frequencies of each metric for each year.

Our analysis shows that the software vulnerability landscape has been dominated by only a few vulnerability types and has changed very little from 2005 to 2019. For example, the overwhelming majority of vulnerabilities are exploitable over the network (i.e., remotely). The complexity to successfully exploit these vulnerabilities is dominantly low while attackers are generally not required to have any level of prior access to their targets (i.e., having successfully authenticated) in order to launch an attack. And most of the flaws require very limited interaction with users. On the positive side, the damage of these vulnerabilities is mostly confined within the security scope of the impacted components. Few vulnerabilities obtain greater privileges than are available to the exploited vulnerable component.

Our findings are consistent with previous studies [8] (mainly based on CVSS version 2). This indicates that the same vulnerabilities are still being found in our software, suggesting that the community has not been doing a great job correcting the most common vulnerabilities.

The remainder of this paper is organized as follows. Section II presents the CVSS data sets that constitute the basis of our study. Section III gives the details of our analysis and our discussion. Section IV provides a summary of related work and Section V concludes.

II. THE CVSS DATASETS

CVSS consists of three metric groups: base, temporal, and environmental. The base group represents the intrinsic qualities of a vulnerability that are constant over time and across user environments, 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 [6]. In this work, we evaluate only the base metrics as no extensive database of temporal scores exists and the environment metrics are designed for an organization to customize base and temporal scores to their particular environment.

Tables I and II show the base score metrics and possible values for v2 and v3, respectively. The CVSS base score takes into account the exploitability (how easy it is to use the vulnerability in an attack) and impact (how much damage the vulnerability can cause to an affected component) of a vulnerability apart from any specific environment.

The exploitability score is determined by the following:

- attack vector (v2 & v3): 'the context by which vulnerability exploitation is possible',
- attack complexity (v2 & v3): 'the conditions beyond the attacker's control that must exist in order to exploit the vulnerability',
- authentication (v2): 'number of times an attacker must authenticate to a target in order to exploit a vulnerability',
- privileges required (v3): 'the level of privileges an attacker must possess before successfully exploiting the vulnerability', and
- user interaction (v3): a human victim must participate for the vulnerability to be exploited.

TABLE I
CVSS v2 METRICS

| CVSS v2 Metrics | Metric Values |
|------------------------|--------------------------------------|
| Access Vector (AV) | Network (N), Adjacent (A), Local (L) |
| Attack Complexity (AC) | Low (L), Medium (M), High (H) |
| Authentication (Au) | Multiple (M), Single (S), None (N) |
| Confidentiality (C) | Complete (C), Partial (P), None (N) |
| Integrity (I) | Complete (C), Partial (P), None (N) |
| Availability (A) | Complete (C), Partial (P), None (N) |

TABLE II
CVSS v3 METRICS

| CVSS v3 Metrics | Metric Values |
|--------------------------|--|
| Attack Vector (AV) | Network (N), Adjacent (A), Local (L), Physical (P) |
| Attack Complexity (AC) | Low (L), High (H) |
| Privileges Required (PR) | None (N), Low (L), High (H) |
| User Interaction (UI) | None (N), Required (R) |
| Scope (S) | Unchanged (U), Changed (C) |
| Confidentiality (C) | High (H), Low (L), None (N) |
| Integrity (I) | High (H), Low (L), None (N) |
| Availability (A) | High (H), Low (L), None (N) |

The impact score (v2 & v3) is determined by measuring the impact to the confidentiality, integrity, and availability of the affected system. Also included (v3) is a scope metric that 'captures whether a vulnerability in one vulnerable component impacts resources in components beyond its security scope'.

A particular assignment of metric values is then used as input to the CVSS base score equations to generate scores representing the inherent severity of a vulnerability in general, apart from any particular environment. The raw score in the range from 0 to 10 is then often translated into a 'qualitative severity rating scale' (None: 0.0, Low: 0.1 to 3.9, Medium: 4.0 to 6.9, High: 7.0 to 8.9, and Critical: 9.0 to 10.0) [6].

Vulnerability analysts apply the metrics to vulnerabilities to generate CVSS vector strings. The vectors describe the metric values, but not the CVSS scores, for a particular vulnerability using a simplified notation.

The NVD is the 'U.S. government repository of standards based vulnerability management data' [1]. It provides CVSS vectors and base scores for all vulnerabilities listed in the Common Vulnerabilities and Exposures (CVE) [14] [15] catalog of publicly disclosed software flaws. We use NVD to evaluate both CVSS v2 and v3 vectors and scores. The v2 data covers all CVE vulnerabilities published between 2005 and 2019. The v3 data ranges from 2015 to 2019 (only limited v3 data is available prior to 2015). These coverage dates result in the inclusion in our study of 118 173 v2 vectors and scores and 55 441 v3 vectors and scores.

III. DATA ANALYSIS

We analyze the NVD CVSS data in order to better understand the software vulnerability landscape. We investigate both the current nature of the threat posed by the existence and public disclosure of these vulnerabilities as well as how this threat has changed over time. To achieve this, we conduct the three studies described previously where we analyze the

following: score distributions, metric value distributions, and relative rankings of the most frequent metric values.

A. Score Distributions

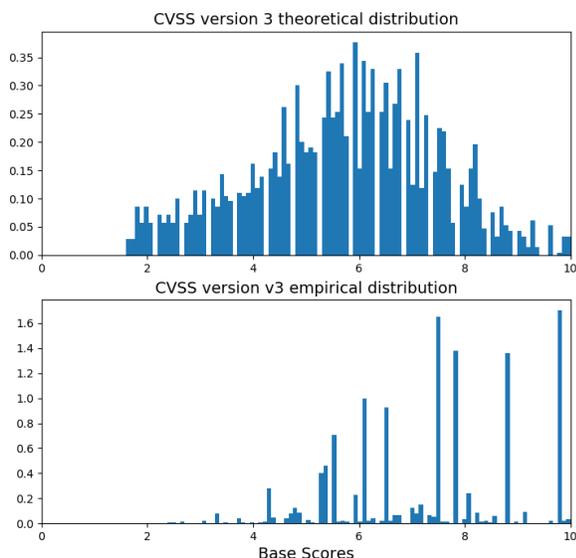


Fig. 1. Theoretical vs Empirical Score Distributions for CVSS version 3. The y-axis shows the numerical values of the base scores of vulnerabilities. The top figure is obtained by considering all possible assignments of metric values, while the bottom figure corresponds to scores of actual vulnerabilities discovered in software.

The top graph of Figure 1 shows the theoretical distribution of the v3 scores (v2 scores are similar and not shown in the paper due to space limitation. They can be found in the appendix of [16]). These plots show what is expected if all CVSS vectors (i.e., vulnerability types) are equally likely to occur. Note how the theoretical distribution was designed, by the FIRST CVSS committee, to spread CVSS scores throughout the range with a somewhat normal distribution with the most probable scores occurring in the middle of the distribution (a little biased to the right). That said, it is interesting in that for both v2 and v3 some scores are not possible even though they lie within the valid range of score values.

The empirical distribution is shown in the bottom of Figure 1 for v3. The empirical data indicates a predominance of certain vectors (groupings of vulnerability characteristics) in the real world. Thus, only a few vulnerability feature sets describe the majority of publicly disclosed vulnerabilities. This leads to the frequent use of just a very small number of scores. A similar observation was made in a previous study of the v2 scoring system [8].

The results observed with v3, which uses data from 2015 to 2019 (since v3 vectors are not generally available prior to 2015) are similar to those with v2, which uses data from 2005 to 2019. Hence, the long-term obtained with CVSS v2 data is confirmed by the shorter-term data of CVSS v3.

B. Metric Value Distributions

To investigate more carefully (in order to identify) possible differences per year and trends over time, we focus on the distributions of each set of metric values per year over the time period of study. Figure 2 provides the histograms for v3 from 2015 to 2019. We have also plotted the histograms for v2 [16], which cover from 2005 to 2019. The inclusion of v2 in the study allows for a comparison over 15 years as opposed to being limited to just 5 years with v3, due to its more recent development.

The histograms for individual metric values for v3 appear almost the same year to year for the 5 years of study. This is the same in v2 over the longer time period of 15 years with some small exceptions: in 2014, the attack vector (AV) value of adjacent had some significance. According to the NVD team [17], this was a one time anomaly due to more than 800 CVEs all being announced simultaneously by an organization doing analyses on phone apps. The Attack Complexity (AC) value ‘Medium’ increased some from 2007 onward, but then was steady, the Authentication (Au) value ‘Single’ increased slightly over the years, and the Confidentiality (C), Integrity (I), and Availability (A) metric proportions between ‘None’, ‘Partial’, and ‘Complete’ varied slightly from year to year while generally maintaining themselves about the same.

Overall though, the software vulnerability landscape for publicly disclosed vulnerabilities has been almost static during the period of study. This said, doing comparisons between the v2 and v3 histograms, we see some differences, but this is due to differences in the approaches of the two versions of CVSS. These differences are primarily seen in regards to the metrics C, I, and A, which we will discuss shortly.

Consider the AV metric which reflects the context by which the vulnerability can possibly be exploited: Network (N), Adjacent (A), Local (L), or Physical (P). Both data sets show a high peak at N, a low peak at L and almost nothing at A and P. This indicates that the overwhelming majority of publicly disclosed software vulnerabilities are exploitable over the network (i.e., remotely), and it has been that way consistently through the period of study.

The AC metric describes the conditions beyond the attacker’s control that must exist in order to exploit the vulnerability. When it is low (AC:L), the attacker can expect repeatable easy successes, while when it is high (AC:H) the attack is less likely to be successful. The data shows that the AC metric is largely dominated by the values of AC:L for v3 and AC:L and AC medium (AC:M) for v2. This indicates that the set of publicly disclosed vulnerabilities have been predominantly easy to exploit.

This “easiness” to exploit vulnerabilities is confirmed by the other metrics for each CVSS version. For v3, the Privileges Required (PR) metric describes the level of privileges an attacker must possess before successfully exploiting a vulnerability. The User Interaction (UI) metric captures the requirements for a human user (other than the attacker) to participate in the successful compromising of the vulnerable

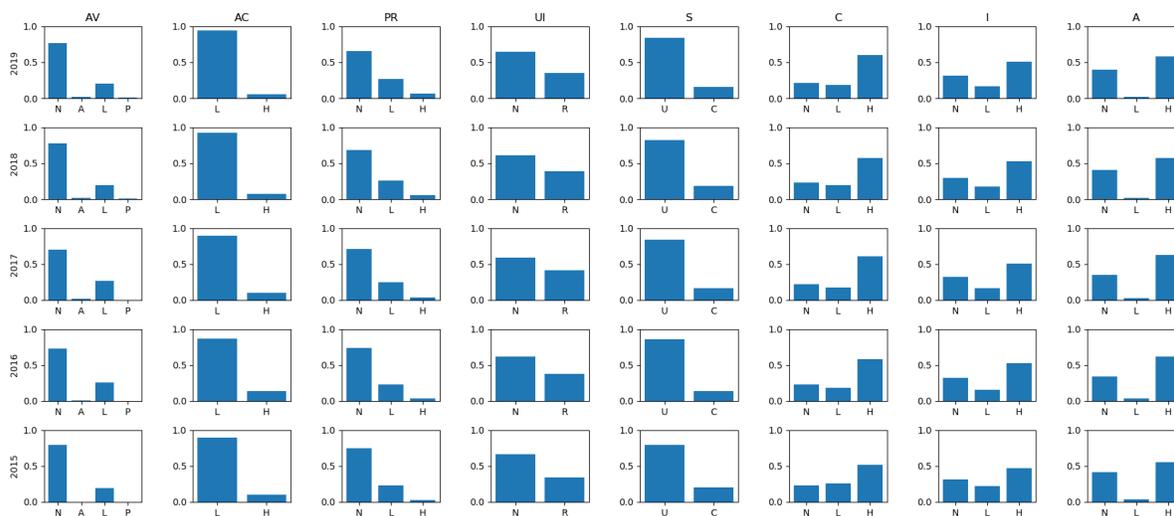


Fig. 2. CVSS v3 metrics' values distributions over the years

components. The data shows that in most of the cases, no privilege is required and very little user interaction is needed for a successful attack.

Similarly, with v2, the Au metric measures the number of times an attacker must legitimately authenticate to a target in order to be in a position to exploit a vulnerability. The data shows that almost always, there is no authentication required prior to exploiting a vulnerability. Sometimes a single authentication is required, but almost never is there a vulnerability that requires multiple authentications in order to be successfully exploited.

CVSS v3 introduced a new Scope (S) metric, which captures the spill-over effect: how much a vulnerability in one vulnerable component impacts resources in components outside of its security scope. When the scope is unchanged (S:U), there is no spill-over, while when the scope is changed (S:C) the vulnerability will very likely affect other components. The data shows that the scope metric has predominantly been S:U.

The last three metrics C, I, and A are common to both CVSS versions. They capture the extent to which a successful exploitation of a vulnerability will affect these three principles of security on the effected component. With respect to these metrics, the v3 data shows that the impact on C, I, and A has predominantly been high (C:H, I:H, and A:H) with very similar distributions for all the years. The v2 data also shows a similar stationary behavior in the distributions. However, the difference in the fraction of high for v3 and complete for v2 is notable. One might expect the high values in CVSS v3 to be equivalent to the complete values for v2. However, this is not the case as they are defined differently. According to the NVD team [17] “the CVSS scoring systems are fundamentally different regarding qualifications for CIA Partial/Complete and Low/High. This is a common misconception due to the nuances of the scoring systems. In addition to this, the NVD takes the approach of representing the worst-case scenario when information is lacking. This typically results in default

values of HIGH being attributed to a CVE unless data is available that identifies a limitation to the impact or meets qualifying text for the specification.”

C. Relative Rankings of the Most Frequent Metric Values

We now focus on the most prominent individual values of the metrics, evaluating the rankings of the top 10 metric values observed each year and providing a comparison between the years. Figure 3 shows the rankings for v3 (the same plots for v2 can be found here [16]). The y-axis show the top 10 most prevalent metric values, ordered from the least frequent to most frequent. Thus, the set of metric values included in the y-axis is significant (only the top ten are shown). The x-axis show the years. Each (x,y) point indicates that in year x the metric value at y has a rank indicated by the number in the circle. The size of the circle is proportional to the number of times that metric value appeared in a score in that year. For example in Figure 3, in 2017, the metric value AV-N was the fourth most frequent metric value within the set of all v3 vectors. However, in 2018 and 2019 this metric value became the third most frequent. Notice that in general, a value might appear in the top 10 of one year while not appearing in another year. Whenever that happens, we rank that particular value 11 for all the years in which it did not appear.

For v3 (see Figure 3), we observed that the same top 10 values appeared from 2016 to 2019. Furthermore, only one of those values is missing in the 2015 top 10. In addition, these values were ranked almost the same over the years. The top 2 are constant and in the same order over the time period 2015 to 2019. The top 4 and the bottom 4 (including the 11th appended value) are also constant with minor changes in the order they appear over the years. The v2 data shows similar results (see [16]). This is another illustration of the stationary threat landscape observed earlier. It also corroborates the observations in Figure 1, that the landscape has been dominated by just a few vulnerability types.

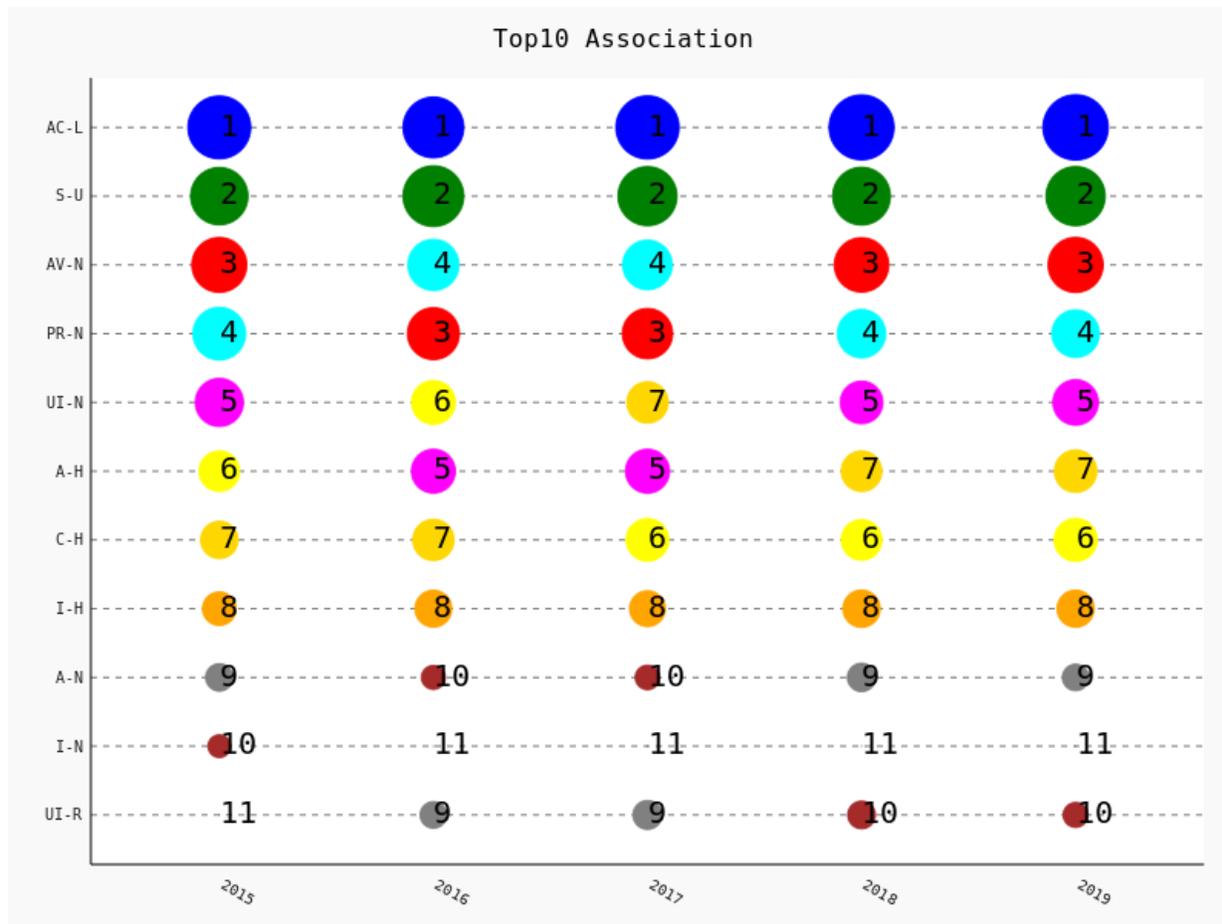


Fig. 3. CVSS v3 top 10 rankings

In conclusion, our data indicates that the vulnerability threat landscape has been dominated by a few vulnerability types and has not evolved over the years. The overwhelming majority of software vulnerabilities are exploitable over the network (i.e., remotely). The complexity to successfully exploit these vulnerabilities is dominantly low and very little authentication to the target victim is necessary for a successful attack. Moreover, most of the flaws require very limited interaction with users. The damage of these vulnerabilities has, however, mostly been confined within the scope of the compromised systems.

IV. RELATED WORK

There are many efforts to understanding the software vulnerability landscape. These efforts include reports by security solutions vendors [18], [19], white papers from non-profits such as MITRE [20] and SANS [21], as well as academic papers. For CVSS, most studies focused on the aggregation equation that produces the CVSS numerical scores representing the severity of the vulnerability. Surprisingly, we found no studies on v3 despite its preponderance in commercial security software.

Reference [8] is among the first statistical studies of the CVSS scoring system. It evaluated v1 and proposed improvements that contributed to the release of v2. Our study considers both v2 and v3 (but does not try to improve on either). Relative to the statistical evaluation, we consider our paper as a continuation and update of the work in [8]. However, our work uses data from a much longer time period. It also goes one step further by analyzing association rules of vulnerability metrics. It is worth noting that there are similarities between the results of the two studies. For instance, both papers show the predominance of certain types of vulnerabilities. Our historical analysis (which was not performed in [8]) shows that this predominance is maintained over the years.

Reference [11] considers CVSS v1 and v2 and analyzes how effectively v2 addresses the deficiencies found in v1. It also identifies new deficiencies. In contrast, our motivation was to understand the threat landscape.

Reference [13] uses empirical data from an international cyber defense exercise to study how 18 security estimation metrics based on CVSS correlate with the actual *Time-To-Compromised* (TTC) of 34 successful attacks. This study uses TTC as a dependent variable to analyze how well different security estimation models involving CVSS are able to ap-

proximate the actual security of network systems. The results suggest that security modeling with CVSS data alone does not accurately portray the time-to-compromise of a system. This result questions the applicability of the CVSS numerical scoring equation. Our study focused on the raw CVSS vectors, which represent the actual experts' opinions about the vulnerabilities.

Reference [22] uses NVD data to study trends and patterns in software vulnerabilities in order to predict the time to next vulnerability for a given software application. Data mining techniques were used as prediction tools. The vulnerability features used to aid the prediction are the published time of each vulnerability and its version. We believe that these features are not sufficiently informative. Instead, we directly use the eight metrics from the CVSS base scores which constitute the best available information covering large multi-year sets of vulnerabilities.

Reference [23] also carried out a predictive study based on the NVD/CVSS and ExploitDB [24] data. Using the CVSS data, it attempts to answer two questions: (1) *Can we predict the time until a proof of concept exploit is developed based on the CVSS metrics?* and (2) *Are CVSS metrics populated in time to be used meaningfully for exploit delay prediction of CVEs?* The former is answered in the positive, while the latter is answered in the negative. While using the same datasets, our objective differs from that in [23]. We did not attempt to predict the threat landscape; we provide a thorough historical and statistical study of vulnerabilities for the last fifteen years.

The work in [25] is another assessment of CVSS. It evaluates the trustworthiness of CVSS by considering data found in five vulnerability databases: NVD, X-Force, OSVDB (Open Source Vulnerability Database), CERT-VN (Computer Emergency Response Team, Vulnerability Notes Database), and Cisco IntelliShield Alerts. It then uses a Bayesian model to study consistencies and differences. It concluded that CVSS is trustworthy and robust in the sense that most of the databases generally agree. This suggests that our focus on the NVD to study the threat landscape is justified: studies using data from the other databases will likely lead to the same conclusions.

All of the studies cited above are focused on v1 and v2. In our literature survey, we did not find a single study that uses the updated and significantly modified v3 to understand the software vulnerability landscape. We believe that the present paper is the first of this kind in doing so. Furthermore, our study is the first to use association rule mining and co-occurrence of vulnerability metrics' values in an attempt to characterize the software threat landscape.

V. CONCLUSION

Our data indicates that the vulnerability threat landscape for publicly disclosed vulnerabilities has been dominated by a few vulnerability types and has not significantly changed from 2005 to 2019. However, the underlying software flaw types that enable these vulnerabilities change dramatically from year to year (for example, see [26]). This means that many flaw types result in vulnerabilities with the same properties. This is

bad news because it means, as a security community, it will be difficult to eliminate certain vulnerability types because they result from a plethora of underlying software flaw types.

Another concern is that the overwhelming majority of software vulnerabilities are exploitable over the network. When developing software, efforts should be made to reduce unnecessary connections, protect necessary ones, and require more authentication where possible to reduce attack surface area. Another significant issue is that most of the vulnerabilities require no sophistication to be exploited (but again this is hard to improve upon due to the many software flaw types that allow this).

These two factors together combined with the finding that most vulnerabilities require very limited interaction with users facilitates the widespread hacking occurring today. Often in security literature the human is cited as the weakest link. While certainly humans can be exploited, within the set of CVE type vulnerabilities, exploitation of humans plays a very minor role. Hence, although training humans might always help strengthen security, to obtain a better impact in this area, the priority should be shifted to correcting these constant vulnerabilities.

Overall, this study documents the security community's inability to eliminate any types of vulnerabilities through addressing the related software flaw types. In 15 years, the vulnerability landscape has not changed; through the lens of the metrics in this paper we are not making progress. Perhaps we as community need to "stop and think" about the ways we are developing software and/or the methods we use to identify vulnerabilities. The security community needs new approaches. We do not want to write this same paper 15 years from now showing that, once again, nothing has changed.

Overall, this study shows that either we (the community) are incapable of correcting the most common software flaws, or we are focusing on the wrong flaws. In either case, it seems to us that there is a need to "stop and think" about the ways we are developing software and/or the methods we use to identify vulnerabilities.

ACKNOWLEDGEMENT

This work was partially accomplished under NIST (National Institute for Standards and Technology) Cooperative Agreement No.70NANB19H063 with Prometheus Computing, LLC. The authors would like to thank the NVD staff for their review and consideration of this work.

Any mention of commercial products or entities is for information only; it does not imply recommendation or endorsement.

REFERENCES

- [1] "National vulnerability database," 2020, accessed: 2020-01-10. [Online]. Available: <https://nvd.nist.gov>
- [2] "Common vulnerability scoring system special interest group," accessed: 2019-12-10. [Online]. Available: <https://www.first.org/cvss>
- [3] "Forum of incident response and security teams," accessed: 2020-01-10. [Online]. Available: <https://www.first.org/>
- [4] M. Schiffman, A. Wright, D. Ahmad, and G. Eschelbeck, "The common vulnerability scoring system," *National Infrastructure Advisory Council, Vulnerability Disclosure Working Group, Vulnerability Scoring Subgroup*, 2004.

- [5] P. Mell, K. Scarfone, and S. Romanosky, "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, 2007, p. 23.
- [6] "Common vulnerability scoring system v3.0: Specification document," accessed: 2020-2-5. [Online]. Available: <https://www.first.org/cvss/v3.0/specification-document>
- [7] "Common vulnerability scoring system v3.1: Specification document," accessed: 2020-2-5. [Online]. Available: <https://www.first.org/cvss/v3.1/specification-document>
- [8] P. Mell and K. Scarfone, "Improving the common vulnerability scoring system," *IET Information Security*, vol. 1, no. 3, pp. 119–127, 2007.
- [9] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Security & Privacy*, vol. 4, no. 6, pp. 85–89, 2006.
- [10] H. Holm and K. K. Afridi, "An expert-based investigation of the common vulnerability scoring system," *Computers & Security*, vol. 53, pp. 18–30, 2015.
- [11] K. Scarfone and P. Mell, "An analysis of cvss version 2 vulnerability scoring," in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society, 2009, pp. 516–525.
- [12] R. Wang, L. Gao, Q. Sun, and D. Sun, "An improved cvss-based vulnerability scoring mechanism," in *2011 Third International Conference on Multimedia Information Networking and Security*. IEEE, 2011, pp. 352–355.
- [13] H. Holm, M. Ekstedt, and D. Andersson, "Empirical analysis of system-level vulnerability metrics through actual attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 825–837, 2012.
- [14] D. W. Baker, S. M. Christey, W. H. Hill, and D. E. Mann, "The development of a common enumeration of vulnerabilities and exposures," in *Recent Advances in Intrusion Detection*, vol. 7, 1999, p. 9.
- [15] "Common vulnerabilities and exposures," 2020, accessed: 2020-2-5. [Online]. Available: <https://cve.mitre.org>
- [16] A. Gueye and P. Mell, "A historical and statistical study of the software vulnerability landscape," 2021.
- [17] NVD, "private communication," Mar. 2019.
- [18] Symantec, "2019 internet security threat report," 2020, accessed: 2020-02-01. [Online]. Available: <https://www.symantec.com/content/dam/symantec/docs/reports/isr-24-2019-en.pdf>
- [19] McAfee, "McAfee labs 2019 threats predictions report," 2020, accessed: 2020-02-01. [Online]. Available: <https://www.mcafee.com/blogs/other-blogs/mcafee-labs/mcafee-labs-2019-threats-predictions/>
- [20] MITRE, "2019 cwe top 25 most dangerous software errors," 2020, accessed: 2020-02-01. [Online]. Available: https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html
- [21] SANS, "2020 sans cyber threat intelligence (cti) survey," 2020, accessed: 2020-02-01. [Online]. Available: <https://www.sans.org/reading-room/whitepapers/analyst/2020-cyber-threat-intelligence-cti-survey-39395>
- [22] S. Zhang, D. Caragea, and X. Ou, "An empirical study on using the national vulnerability database to predict software vulnerabilities," in *International Conference on Database and Expert Systems Applications*. Springer, 2011, pp. 217–231.
- [23] Y. Y. A. Feutrill, D. Ranathunga and M. Roughan, "The effect of common vulnerability scoring system metrics on vulnerability exploit delay," in *2018 Sixth International Symposium on Computing and Networking (CANDAR), Takayama*, 2018, pp. 1–10.
- [24] "Exploit database," 2020, accessed: 2020-02-01. [Online]. Available: <https://www.exploit-db.com/>
- [25] P. Johnson, R. Lagerstrom, M. Ekstedt, and U. Franke, "Can the common vulnerability scoring system be trusted? a bayesian analysis," *IEEE Transactions on Dependable and Secure Computing*, 2016.
- [26] "National vulnerability database, cwe over time," 2019, accessed: 2019-12-10. [Online]. Available: <https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cwe-over-time>