A Systematic Analysis of Debated Vulnerabilities in the National Vulnerability Database

Sarah Elder
Computer Science)
North Carolina State University
Raleigh, NC
seelder@ncsu.edu

Laurie Williams
Computer Science)
North Carolina State University
Raleigh, NC
laurie_williams@ncsu.edu

I. Introduction

Determining whether a vulnerability report represents a true vulnerability is not as simple as it may sound. Many of the problems associated with vulnerability analysis, e.g. determining whether a security vulnerability exists within a product or determining whether a fault in a product is actually a security vulnerability, are computationally undecidable for the general case [5]. In spite of their theoretical limits, tools that assist analysts in identifying and classifying security vulnerabilities are a valuable asset for practitioners and an area of active research. Evaluation of security tools, particularly automated tools, relies on measures that assume that it can be conclusively determined whether software contains vulnerabilities [2], [3].

Debates over whether an issue is a vulnerability have implications well beyond the theoretical. Gary McGraw [4] considers these "catfights" a necessary and beneficial part of risk assessment. However, disagreement over vulnerabilities is not always positive. For example, September 30, 2018 a vulnerability was reported both in Python's virtualenv github[1] and in the National Vulnerability Database (the NVD)[2] on the security features (CWE-254) of virtualenv. Specifically, the vulnerability reporter claimed that virtualenv 'allows a sandbox escape'. In the github issue comments, multiple individuals indicated that virtualenv was never intended to be a secure sandbox, hence this was an invalid vulnerability report. The discussion continued, and on October 22, 2018 one participant noted that the discussion on whether the issue was a true security vulnerability was itself "now reaching the point of wasting significant time that should be spent on real security issues, in other words, we now do have a sort of DoS on our security infrastructure.". The discussion continued into 2019.

In spite of the implications that debates over vulnerabilities have for vulnerability identification, classification, and mitigation efforts; little research has been done examining what these debates are about, how these debates arise, and what the outcomes of the debates are. *The goal of this research is to assist practitioners in identifying, classifying, and mitigating security vulnerabilities through a systematic analysis of debates on whether vulnerability reports in the National Vulnerability Database (NVD) represent actual vulnerabilities.*

Our research questions are as follows:

- RQ1: What are the concerns of individuals who do not think that these reports are true vulnerabilities?
- RQ2: Who are the individuals who do not think that the issues are true vulnerabilities?
- RQ3: How were the debated vulnerabilities addressed?

We perform our analysis on the National Vulnerability Database (NVD). Each entry in the database is referred to as a CVE because each entry is also registered in the Common Vulnerabilities and Exposures (CVE) list. We look at CVEs from 2018 that were publicly marked as Disputed or Rejected in the CVE list. To answer RQ1 we perform qualitative analysis with two raters to identify 5 themes of the debates over these disputed and rejected vulnerabilities. To answer RQ2 we use keyword searching and heuristics combined with manual verification to identify who the individuals debating the issue are. To answer RQ3 we perform another qualitative analysis with two raters to identify whether a code change was made or planned to be made in response to the vulnerability, or whether no code change would be made. To answer RQ4 we use linear regression to determine whether the severity score, as noted by the Common Vulnerability Scoring System (CVSS) measure provided by the NVD, is related to whether an entry in the NVD is debated. We examine how the CVSS score changes in response to the vulnerability, and how those changes impact the relationship.

It should be noted that the CVEs examined represent less than 1% of all CVEs in the NVD. However, given the NVD's status as a key vulnerability repository [], it is an appropriate starting point for examining why vulnerabilities are debated.

The most frequently occurring theme amongst the debates was concern over the conditions that were necessary to exploit a vulnerability, such as the level of access needed. The second most frequently occurring theme was who is responsible for mitigating the vulnerability.Debates over whether the vulnerability was mis-triaged, whether the vulnerability was part of intended functionality, and whether there was sufficient information to determine if the report was a vulnerability were the three remaining themes.

[1]https://github.com/pypa/virtualenv/issues/1207
[2]https://the NVD.nist.gov/vuln/detail/CVE-2018-17793

Ninety percent of debates were instigated by maintainers of the product in which the vulnerability was reported. Unsurprisingly, these debates are also all occurring in products where the maintainer is not the individual responsible for assigning CVEs to their product, which represents only around 1/3 of the CVEs in the NVD. This suggests that the debates identified are a specific subset of all the debates that may occur. Future research in this area may find additional themes that are more applicable in other contexts.

In debates where it was clear whether or not the product maintainers planned to "fix" the issue or make some change, a similar number of maintainers indicated they planned to make a code change (22 out of 53) as indicated that no change would be made (23 out of 53). The remainder indicated they would change documentation, or considered it a feature request. This further suggests that determining whether or not an issue is a vulnerability is far more complex than simply whether or not something should or even could be fixed.

The contributions of this work are:

- Analysis of why vulnerabilities are considered vulnerabilities

II. Methodology

A. Debate identification

Our process for identifying debates about vulnerabilities in the NVD is as follows:

1) *Identify Relevant CVEs:* To begin, we identify a set of relevant CVEs:
   a) We start by identifying the 2018 CVEs that were marked "Disputed" or "Rejected" as of April 2019 for which there is published information available
   b) We remove all CVEs that were disputed or rejected for reasons not related to whether the report is a security Vulnerability. These CVEs were rejected because they were duplicates of another CVE, because the vulnerability report had not been sent to the correct CNA, or because the CVE ID had never been assigned to a vulnerability. In most cases, this was clearly noted in the description of the CVE. For example a CVE that was rejected for being a duplicate of another CVE generally includes the following in the description " This candidate is a reservation duplicate of CVE-XXXX-XXXXXX" where CVE-XXXX-XXXXXX is the CVE ID of the vulnerability that the rejected CVE is a duplicate of.
   c) We remove CVEs where information on the debate, as found through the references attached to each CVE in the NVD, is not in English
   d) We remove Rejected vulnerabilities from the dataset where insufficient information is available to determine why the issue is regected. For example, e.g. the description for CVE-2018-7863 states "This candidate was withdrawn by its CNA.", and the historical record for this CVE provides no additional information or references)
2) *Consolidate CVEs* Once the Relevant CVEs are identified, we determined which CVEs were part of distinct debates. We used the following criteria to determine when debates about two or more CVEs were not distinct. All three criteria must be met:
   - Participants - The individual claiming that the report does not represent an actual vulnerability is the same for all CVEs.
   - Product - The vulnerabilities described are in the same product.
   - Dispute Content - At least one of the following is true:
     – The CVE's References are the same, and the reference document does not clearly delineate which issues are associated with which CVE (as noted by both researchers reviewing the documentation), then this criterion does NOT apply.
     – The information for the CVEs and their disputes cross reference each other as overlapping or possibly the same.
     – The documentation about the debate is identical, i.e. it appears to be "copy-pasted", the CVE ID numbers are sequential or nearly sequential. E.g. CVE-2018-5270 through CVE-2018-5279 are all disputed because the vendor has "not been able to reproduce the issue on any Windows operating system version (32-bit or 64-bit). "[3]
3) *Update if appropriate:* We update our list of debates if additional, relevant information is extracted during the in-depth analysis of RQ1 as follows
   a) We remove CVEs if two or more researchers agree that the reason for the debate is one of the reasons described previously that were not relevant to this study, e.g. a CVE is a duplicate of another CVE.
   b) We consolidate additional debates if we determine that additional CVEs met the criteria above. If codes have already been applied to the debates for the individual CVEs, the codes already identified for each CVE should be jointly applied to the consolidated debate.

We use the list of CVEs

B. RQ1: What are the concerns of individuals who do not think that these reports are true vulnerabilities?

To answer RQ1, we performed a qualitative thematic analysis [1] on the debates of CVEs in the NVD.

The Analyses was performed in two stages. In the first stage, two raters developed codes and themes, which were verified in the second stage. For both the first and second stages, the raters started with the same set of documents, but did NOT agree to specific text segments within those documents. Text segmenting is frequently a separate activity performed prior to coding [1]. However, the raters noted that when comparing text segments prior to formal coding, the choice of how to

[3]https://nvd.nist.gov/vuln/detail/CVE-2018-5279

segment the text depended on how the rater planned to code the document. Consequently, to encourage independent coding, each rater independently segmented the text within the agreed-to documents and coded the segments prior to coming to an agreement on segment length.

1) First Stage: In the First Stage two raters independently reviewed a random subset of the debated vulnerabilities, using open coding [6] to identify an initial set of codes. Once both raters completed the same set of 10 CVEs, the raters discussed and compared codes. This continued iteratively until XX vulnerabilities had been examined. As needed, the raters expanded their random subset to include additional codes until patterns emerged that were used to develop their codebook, and grouped the codes into themes.

2) Second Stage: Using the codes and themes developed during the first stage, the raters independently reviewed the remaining 35 debates and applied the codes to validate whether the agreement on which categories applied. Agreement was computed at the theme level, as there were 75 different codes that could be applied. While multiple codes could apply to the same debate, agreement at the code level would have little meaning since some codes would only be used once or not at all. The raters then met to resolve disagreements.

C. RQ2

To answer RQ2 we perform a keyword search and pattern matching on the descriptions of the CVEs to identify whether the debate came from the maintainer of the software, a third party, or another source. We also manually verified the source for all debates.

D. RQ3

To answer RQ3, one author reviewed the documentation for each debate to determine whether any change was made to the code in response to the CVE. In addition, a second author reviewed a subset of the debates and performed a similar classification of whether a code change was made to the product as a result of the debate to verify the results obtained by the original classification. We compute the agreement between the classifiers using Cohen's Kappa.

REFERENCES

[1] Daniela S Cruzes and Tore Dyba. Recommended steps for thematic synthesis in software engineering. In *2011 International Symposium on Empirical Software Engineering and Measurement*, pages 275–284. IEEE, 2011.

[2] Aurelien M Delaitre, Bertrand C Stivalet, Paul E Black, Vadim Okun, Terry S Cohen, and Athos Ribeiro. Sate v report: Ten years of static analysis tool expositions. Technical report, 2018.

[3] Seyed Mohammad Ghaffarian and Hamid Reza Shahriari. Software vulnerability analysis and discovery using machine-learning and data-mining techniques: A survey. *ACM Computing Surveys (CSUR)*, 50(4):56, 2017.

[4] Gary McGraw. *Software security: building security in*. Addison-Wesley Professional, 2006.

[5] Bernhard Reus. *Limits? What Limits?*, pages 1–9,97–112. Springer International Publishing, Cham, 2016.

[6] Johnny Saldaña. *The coding manual for qualitative researchers*. Sage, 2015.