Poster: How Bad is it, Really? An Analysis of Severity Scores for Vulnerabilities

Christopher Theisen
North Carolina State University
Raleigh, North Carolina
crtheise@ncsu.edu

Laurie Williams
North Carolina State University
Raleigh, North Carolina
lawilli3@ncsu.edu

ABSTRACT

To date, vulnerability research has focused on the binary classification of code as vulnerable or not vulnerable. To better understand the conditions in which vulnerabilities occur, researchers must consider the severity of these vulnerabilities in addition to a binary classification system. To explore this issue, we mined 2,979 publicly disclosed vulnerabilities from Fedora 24 and 25. We then found severity scores from the Common Vulnerability Scoring System (CVSS) and plotted the distribution of these vulnerabilities. We found that publicly scored vulnerabilities skew high, with few vulnerabilities rated lower than a 5. We then explore other potential issues with the use of CVSS in practice, such as imbalances in Confidentiality, Availability, and Integrity scores.

CCS CONCEPTS

• **General and reference** → **Metrics**; • **Security and privacy** → **Vulnerability management**;

KEYWORDS

vulnerabilities, severity, metrics

ACM Reference Format:
Christopher Theisen and Laurie Williams. 2018. Poster: How Bad is it, Really? An Analysis of Severity Scores for Vulnerabilities. In *Proceedings of 5th Annual Hot Topics in the Science of Security (HoTSoS) Symposium (HotSoS'18).* ACM, New York, NY, USA, Article 4, 1 page. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Research into vulnerability prediction to assist practitioners in finding vulnerabilities relies, as one might expect, on the quality of vulnerability datasets used to validate the approaches. Previous research in this space has focused on predicting vulnerabilities in a binary fashion (something either has a vulnerability or it does not) [1, 2]. However, vulnerabilities come in many different forms, and different vulnerabilities can have drastically different consequences if exposed and exploited by a malicious user. In this paper, we present a distribution of 2,979 vulnerabilities mined for Fedora 24 and 25 and describe the distribution of the scores and subscores.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
HotSoS'18, April 2018, Raleigh, North Carolina USA
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

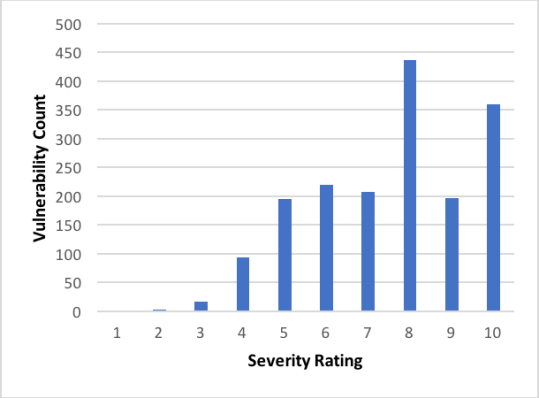


Figure 1: Distribution of severity of vulnerabilities occurring in Fedora 24 and 25.

2 POSTER CONTENT

In our poster, we describe the distribution of vulnerability severity for the Fedora operating system. We also manually verify the security vulnerabilities reported publicly as actual vulnerabilities, and use keyword searches to identify bugs that should also be included in vulnerability datasets. Our results can be found in Figure 1.

The distribution of subscores contributes to the overall skew of CVSS. As an example, three of the subcomponents of the base score for CVSS v3 are Confidentiality, Integrity, and Availability. For each of these measures, users can mark their vulnerability as having LOW impact or HIGH impact, if there is an impact to consider. For our dataset, 87% , 87%, and 98% of the entries were marked as HIGH impact for Confidentiality, Integrity, and Availability, respectively. While there are guidelines on how to apply each of these measures, a clearer distinction between LOW and HIGH, or a wider range of options, may make vulnerability data easier to study. In addition, making scores contextual to the program in question is also a possibility. For example, a vulnerability that is a "nine" for one system may only be a "five" for another. Based on these results, we recommend further work on the curation of validated vulnerability datasets for researchers to evaluate prediction models with.

REFERENCES

[1] Riccardo Scandariato, James Walden, Aram Hovsepyan, and Wouter Joosen. 2014. Predicting vulnerable software components via text mining. *IEEE Transactions on Software Engineering* 40, 10 (2014), 993–1006.
[2] Christopher Theisen, Kim Herzig, Brendan Murphy, and Laurie Williams. 2017. Risk-based attack surface approximation: how much data is enough?. In *Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), 2017 IEEE/ACM 39th International Conference on.* IEEE, 273–282.