Security Vulnerability and Patch Management in Electric Utilities: A Data-Driven Analysis

Fengli Zhang
University of Arkansas
fz002@email.uark.edu

Qinghua Li
University of Arkansas
qinghual@uark.edu

ABSTRACT

This paper explores a real security vulnerability and patch management dataset from an electric utility in order to shed light on characteristics of the vulnerabilities that electric utility assets have and how they are remediated in practice. Specifically, it first analyzes the distribution of vulnerabilities over software, assets, and other metric. Then it analyzes how vulnerability features affect remediate actions.

1 INTRODUCTION

Security vulnerabilities are a big concern for electric utilities. Every month, thousands of new security vulnerabilities and even more might emerge with the assets of an electric utility. Electric utilities have to address each and every vulnerability so that vulnerabilities do not cause security breaches. The NERC CIP regulation [1] also requires vulnerabilities to be timely addressed.

Although much work has been done on vulnerability data analysis [2-4, 7, 9-11], vulnerability and patch management in electric utilities has received little attention. In [2], Stefan et al. explored discovery, disclosure, exploit, and patch dates for about 8000 public vulnerabilities. Shahzad et al. [3] studied vulnerability life cycles. Another work [4] studied the vulnerability disclosure and patch release behavior. Frank Li [7] studied vulnerability characteristics and patch development process. In [9], Treetippayaruk et al. evaluated vulnerabilities of the installed version and the new release of software based on the Common Vulnerability Scoring System (CVSS) and decided whether installing the update is necessary. Most of these analyzed datasets are retrieved from publicly available vulnerability databases, such as the National Vulnerability Database (NVD) [5] and the Open Source Vulnerability Database (OSVDB) [6], but not real-world data from a company. Also, their focus is not electric utilities data. Different from them, this paper explores a vulnerability and patch management dataset from an electric utility[1]. This dataset not only contains vulnerability information, but also information on how vulnerabilities are remediated in practice. This paper analyzes the statistical characteristics of those vulnerabilities over software, assets and other metric, and also studies how vulnerability characteristics affect human operators' decisions of remediating vulnerabilities such as patching and applying mitigation plans.

To the best of our knowledge, this is the first analysis of vulnerability and patch management data for electric utilities, and also the first analysis based on the real operation data of a utility. It sheds light on the current vulnerability and patch management practice in electric utilities. Insights are also obtained as to the security risks with the energy sector and how they are remediated.

This paper is organized as follows. Section 2 describes the dataset from the utility company. Section 3 analyzes the statistical features of vulnerabilities. Section 4 analyzes how vulnerability features affect patching decisions. Section 5 concludes this paper.

2 DATASET DESCRIPTION

The vulnerability and patch management dataset is collected from the utility company for one year from June 2016 to May 2017. It records the security vulnerabilities occurred within their assets during that time window and the remediation actions that human operators took to remediate those vulnerabilities. After removing some incomplete and missing data records, this dataset has around 3500 records. Each record is for one vulnerability, and it contains the software with the vulnerability, vulnerability features, asset features, and the remediation decision for this vulnerability. Sensitive information (i.e. software name and asset name) in the dataset is anonymized to preserve confidentiality, but this will not affect the analysis.

Vulnerability features are described with CVSS metrics [8]. The features and their possible values are shown in Table 1. The CVSS score is a number between 0 and 10 to describe a vulnerability's overall severity. Attacker Vector shows how a vulnerability can be exploited, e.g., through the network or local access. Exploitability indicates the likelihood of a vulnerability being exploited. High as the highest level means exploit code has been widely available, and Unproven as the lowest level means no exploit code is available, with two other levels in between. More detailed explanations for vulnerability characteristics can be found in [8].

Asset features are also considered when making remediation decisions. They are shown in Table 2. Workstation User Login means whether the asset allows users to login, External Accessibility means whether this asset can be accessed externally from the network, Confidentiality Requirement means the asset's requirement for confidentiality, and so on.

Based on vulnerability and asset features, human operators decide the remediate action for each vulnerability. For example, they may decide to patch the vulnerability immediately (Patch Immediately), develop some mitigation plans (Mitigate), or patch in the next scheduled patching cycle (Patch Later).

[1] The company's name is not provided here per the company's requirement.

Table 1. Vulnerability Characteristics

Characteristics	CVSS Score	Attack Vector	User Interaction	Privilege	Confidentiality Impact	Integrity Impact	Availability Impact	Exploitability
Possible Values	Value in 0-10	Network, Adjacent, Local	High, Medium, Low	Multiple, Single, None	Complete, Partial, None	Complete, Partial, None	Complete, Partial, None	High, Functional, Proof-of-Concept, Unproven

Table 2. Asset Characteristics

Characteristics	Workstation User Login	External Accessibility	Confidentiality Requirement	Integrity Requirement	Availability Requirement
Possible values	Yes, No	High, Limited, Authenticated-Only	High, Medium, Low	High, Medium, Low	High, Medium, Low

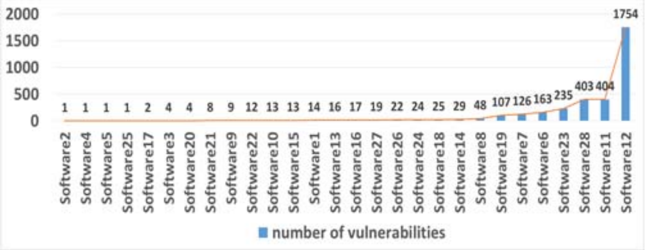

Figure 1: Vulnerability distribution over software

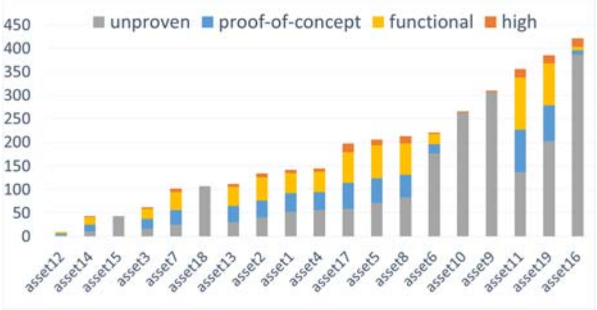

Figure 2: Vulnerability and exploitability distribution over asset

3 VULNERABILITY ANALYSIS

This section explores statistical characteristics of vulnerabilities based on the dataset. Questions to be answered include: How are vulnerabilities distributed over software? What is distribution of vulnerability and its exploitability over assets? How is the CVSS score distributed over asset?

3.1 Vulnerability Distribution over Software

We first explore the vulnerability distributions over software. The results are shown in Fig. 1. Here software names are anonymized in order to preserve confidentiality. The X-axis is software name and the Y-axis is the number of vulnerabilities. From the figure, it can be seen that most vulnerabilities are from a small subset of software. For example, software12 constitutes about 50% of all the vulnerabilities. The top 5 software with the most vulnerabilities (software12, 11, 28, 23 and 6) constitutes about 85% of vulnerabilities. The results indicate that some software are more likely to be attacked due to the many vulnerabilities they have. Thus more attention should typically be paid to them. However, analysis must also include asset characteristics and existing mitigation, which may lower the organizational risk.

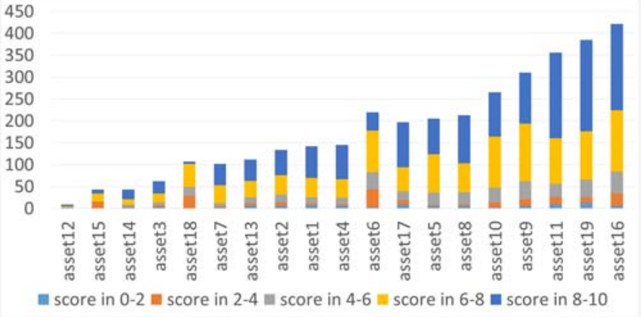

Figure 3: CVSS Score distribution over asset

3.2 Vulnerability and Exploitability Distributions over Asset

In this part, the vulnerability distribution over assets (here each asset represents one device or multiple identical devices) is explored. The results are shown in Figure 2. In order not to leak any sensitive information of the company, all the asset names are replaced with asset1, asset2, …, etc. The dataset consists of 19 assets in total. From the figure, it can be seen that vulnerabilities are not evenly distributed over assets, which is similar to the distribution over software, although not as skewed. Besides, this figure also shows the exploitability distribution over assets, which is a very important vulnerability characteristic. As to be shown, it is a critical factor when deciding how to address a vulnerability. Exploitability levels include Unproven, Proof-of-Concept, Functional, and High in the increasing order. If the exploitability is high, it means the vulnerability has exploit code and can be attacked very easily. It can be seen that some assets (those with higher exploitability levels) are more targeted by attackers and adversaries tend to develop exploit code against the vulnerabilities of those assets. Thus they should be better protected and might deserve more conservative remediation actions.

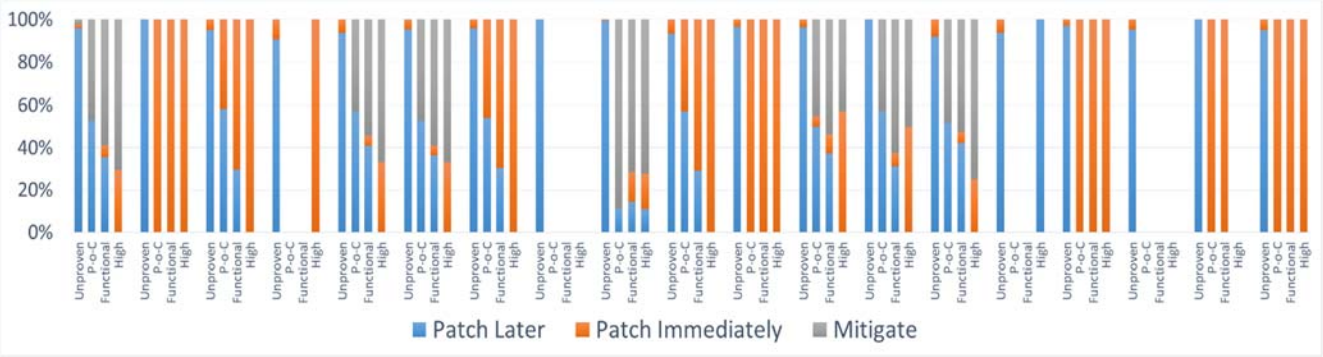

Figure 4: Effects of exploitability on remediation decisions

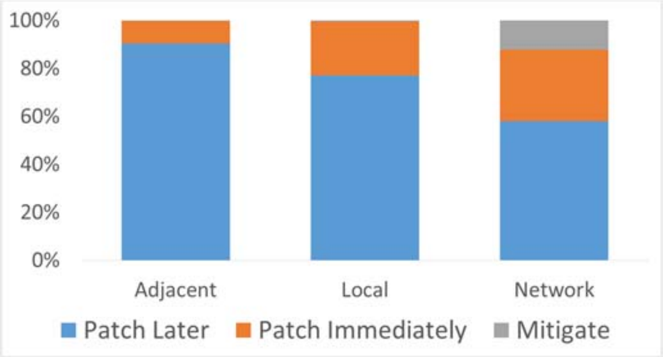

Figure 5: Effects of Attack Vector on remediation decisions

3.3 CVSS Score Distribution over Asset

CVSS score indicates the severity of a vulnerability. It is calculated based on all characteristics of vulnerabilities. It gives an overall assessment of how severe the vulnerability is and how much the vulnerability can affect the system if exploited. Higher value of the score means higher severity. Here, we classify the CVSS score into five groups: group [0, 2), [2, 4), [4, 6), [6, 8) and [8, 10]. As shown in Figure 3, for the vulnerabilities in the dataset, most of the CVSS scores fall into [6, 8) and [8, 10], which implies that most vulnerabilities are of high severity.

4 REMEDIATION DECISION ANALYSIS

When operators decide how to remediate vulnerabilities, many factors (i.e. vulnerability characteristics and asset characteristics) are considered. Some characteristics are usually given more weight in decision making. In this section, we explore how exploitability and attack vector affect decision making.

4.1 Exploitability and Decisions

Figure 4 shows the effects of exploitability on remediation decisions for all assets. For each asset, we analyze the decisions made for the four different exploitability levels. The Y-axis means percentage of vulnerabilities under each remediation decision. For example, for asset1, when exploitability is functional, around 60% vulnerabilities are mitigated and 5% vulnerabilities are patched immediately. From the figure, it can be seen that when the exploitability is Unproven, Patch Later is adopted for almost all the assets because no exploit code is available and thus the risk is low. When the exploitability is High which means the vulnerability can be easily exploited, the vulnerability is patched immediately or mitigated for all the assets. When the exploitability is the other two levels, any of the three decisions is possible, which depends on the asset. In general, the higher the exploitability level is, the more likely the vulnerability is patched immediately or mitigated. Thus exploitability is a critical factor that affects remediation decisions. This calls for more careful assessment of exploitability by software vendors or third parties and better availability of such assessment.

4.2 Attack Vector and Decisions

Attack vector indicates where attacks comes from: network, adjacent network, or local access. It also matters for deciding how to remediate a vulnerability. The effects of attack vector on remediation decisions are shown in Figure 5. The most common attack vector among vulnerabilities is Network. It can be seen that a larger fraction of vulnerabilities is patched immediately or mitigated when the attack vector is the network, than when it is the other two cases. The reason is that exploits or attacks from the network are easier to be launched than those from physically local accesses. Thus the remediation actions for network-connected assets should be more conservative.

5 CONCLUSIONS

This paper explored a real vulnerability and patch management dataset from an electric utility in order to shed light on current practice. Analysis showed that vulnerability distribution over software and asset is skewed, meaning a small subset of software and asset has most of the vulnerabilities, but analysis must include asset characteristics to which the vulnerabilities apply. Also some assets are more likely to be targeted by attacks. In addition, exploitability and attack vector are two important factors affecting remediation decisions for vulnerabilities.

ACKNOWLEDGMENT

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000779.

REFERENCES

[1] http://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx

[2] Frei, Stefan, May, M., Fiedler, U., and Plattner, B. "Large-scale vulnerability analysis." Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense. ACM, 2006.

[3] Shahzad, Muhammad, Muhammad Zubair Shafiq, and Alex X. Liu. "A large scale exploratory analysis of software vulnerability life cycles." Proceedings of the 34th International Conference on Software Engineering. IEEE Press, 2012.

[4] Arora, Ashish, Krishnan, R., Telang, R., and Yang, Y. "An empirical analysis of software vendors' patch release behavior: impact of vulnerability disclosure." Information Systems Research 21.1 (2010): 115-132.

[5] https://nvd.nist.gov/vuln/full-listing

[6] https://blog.osvdb.org/category/vulnerability-databases/

[7] Li, Frank, and Vern Paxson. "A Large-Scale Empirical Study of Security Patches." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017.

[8] https://www.first.org/cvss/v2/guide

[9] Treetippayaruk, Sirikwan, and Twittie Senivongse. "Security vulnerability assessment for software version upgrade." Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2017 18th IEEE/ACIS International Conference on. IEEE, 2017.

[10] Murshed, SM Monzur. An investigation of software vulnerabilities in open source software projects using data from publicly-available online sources. Diss. Carleton University Ottawa, 2017.

[11] Allodi, Luca, and Fabio Massacci. "Attack Potential in Impact and Complexity." Proceedings of the 12th International Conference on Availability, Reliability and Security. ACM, 2017.