

# VULZoo: A Comprehensive Vulnerability Intelligence Dataset

Bonan Ruan<sup>†</sup> Jiahao Liu<sup>†</sup> Weibo Zhao<sup>†</sup> Zhenkai Liang<sup>†</sup>

<sup>†</sup> National University of Singapore  
{r-bonan,jiahao99,weibo,liangzk}@comp.nus.edu.sg

## ABSTRACT

Software vulnerabilities pose critical security and risk concerns for many software systems. Many techniques have been proposed to effectively assess and prioritize these vulnerabilities before they cause serious consequences. To evaluate their performance, these solutions often craft their own experimental datasets from limited information sources, such as MITRE CVE and NVD, lacking a global overview of broad vulnerability intelligence. The repetitive data preparation process further complicates the verification and comparison of new solutions. To resolve this issue, in this paper, we propose VULZoo, a comprehensive vulnerability intelligence dataset that covers 17 popular vulnerability information sources. We also construct connections among these sources, enabling more straightforward configuration and adaptation for different vulnerability assessment tasks (e.g., vulnerability type prediction). Additionally, VULZoo provides utility scripts for automatic data synchronization and cleaning, relationship mining, and statistics generation. We make VULZoo publicly available and maintain it with incremental updates to facilitate future research. We believe that VULZoo serves as a valuable input to vulnerability assessment and prioritization studies. The dataset with utility scripts is available at <https://github.com/NUS-Curiosity/VulZoo>.

## 1 INTRODUCTION

Software vulnerabilities (SVs) have been growing in both scale and complexity, posing substantial security and economic risks in recent years. In 2023 alone, the National Vulnerability Database (NVD) [20] cataloged 28,831 vulnerabilities, marking a 15% increase from the previous year [11]. Cybersecurity Ventures also estimated that cyberattacks cost the global economy approximately 8 trillion USD in the same year [12]. Swift response and remediation of reported SVs before exploitation are crucial for mitigating potential losses. However, due to the large volume of vulnerabilities, it is infeasible to address all of them simultaneously. Therefore, it is essential to accurately assess and prioritize exposed vulnerabilities, enabling developers and researchers to address the most critical SVs as promptly as possible.

Existing vulnerability assessment and prioritization techniques can be classified into two categories: (1) *Improving SV information quality*. A software vulnerability can be profiled from multi-channel information, such as government vulnerability repositories, vulnerability-disclosure email lists, and forums. These solutions aim to enhance the quality of information to better profile SVs. For example, they address issues like inconsistency [29], incorrectness [26], and incompleteness [9], as well as align different sources of information [24]. (2) *Exploring SV intrinsic property*. These methods aim to examine SVs from multiple perspectives, such as assessing the exploitability [30], predicting their vulnerable type [23], their impact [4], and their inherent weaknesses [28].

To evaluate these proposed approaches, researchers often craft specific datasets to check whether they achieve the design objectives. However, there are two shortcomings of the current data preparation process. Firstly, this process is sophisticated and time-consuming, which impedes the rapid verification and comparison of new approaches or tools for vulnerability assessment and prioritization. Secondly, the data sources adopted in existing studies are mainly Common Vulnerabilities and Exposures (CVE) databases and software projects, which are limited compared to the large-scale open vulnerability intelligence available on the Internet. Intuitively, a well-structured and well-maintained dataset can better profile vulnerabilities and aid in assessing and prioritizing them. Unfortunately, to the best of our knowledge, no such dataset currently satisfies these requirements.

To bridge this gap, we introduce VULZoo, a comprehensive vulnerability intelligence dataset, which is readily usable and adaptable for broad vulnerability investigation tasks. VULZoo is derived from 17 heterogeneous data sources, including structural and non-structural data. The structural data contains 604,943 **CVE records** from 4 popular online databases (MITRE CVE [16], NVD [20], ZDI Advisory [31], and GitHub Advisory [7]), **assessment** related metrics from 8 vulnerability related catalogs (CPE [21], CWE [17], CVSS [5], KEV [3], CAPEC [15], AttackerKB [25], ATT&CK [18], and D3FEND [19]). The non-structural data involves the text of 46,882 vulnerability-related **mails**, 46,540 Proof of Concepts (**PoCs**), and 12,626 **patches**. In addition, we mine 11 different **relationships** describing the connectives across data in different categories. Detailed information is provided in Table 2 and Table 3.

In summary, we make the following contributions:

- **Comprehensive Dataset**. We build the VULZoo dataset (~6 GB) based on 17 heterogeneous vulnerability information sources with a well-designed structure.
- **Utility scripts**. We develop a set of scripts for data synchronization and cleaning, relationship mining, and statistics generation. These scripts are leveraged to keep VULZoo up-to-date.

## 2 DATASET CONSTRUCTION

To ensure that VULZoo can comprehensively describe the broad intelligence of software vulnerabilities and support various related tasks, we perform a thorough review of research and industry communities to include as many information channels as possible for profiling vulnerabilities. In the subsequent sections, we first present the data sources included in VULZoo and then detail the data collection process.

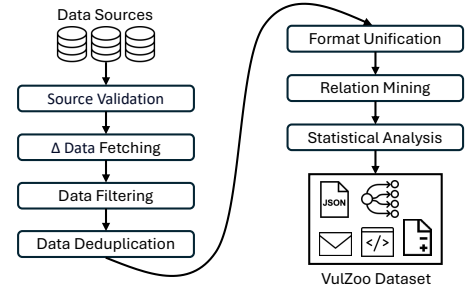
### 2.1 Data Sources

VULZoo adopts 17 data sources for the dataset construction, their names along with the available data of interest presented in Table 1. We give a brief introduction to each data source as follows:

**Table 1: Data sources considered when constructing VulZoo**

Source	Data of Interest
MITRE CVE	CVE records
NVD	Enhanced CVE records and CPE dictionary
ZDI Advisory	Enhanced CVE records
GitHub Advisory	Enhanced CVE records
CISA KEV	Records of known exploited CVEs
MITRE CWE	Weakness catalog
MITRE CAPEC	Attack pattern catalog
MITRE ATT&CK	Offensive technique catalog
MITRE D3FEND	Defensive technique catalog
Rapid7 AttackerKB	Crowdsourcing CVE assessments
OffSec Exploit-DB	Proof of Concepts (PoCs)
Bugtraq	Mails on CVEs
Full-Disclosure	Mails on CVEs
OSS-Security	Mails on CVEs in open-source projects
Linux-CVE-Announce	Mails on CVEs in Linux kernel
GitHub	Patches of CVEs
git.kernel.org	Patches of CVEs in Linux kernel

- MITRE CVE [16] is the official CVE database for cataloging publicly disclosed vulnerabilities in cybersecurity. Each entry in this database includes a unique identifier for a specific vulnerability, along with textual descriptions and corresponding links.
- NVD is another critical vulnerability database maintained by the U.S. government. NVD performs analysis for published CVEs in MITRE CVE database and provides more detailed vulnerability intelligence, such as impact metrics (Common Vulnerability Scoring System, CVSS [5]), vulnerability types (Common Weakness Enumeration, CWE [17]), applicability statements (Common Platform Enumeration, CPE [21]), and other pertinent metadata. NVD also maintains the CPE dictionary for reference.
- ZDI Advisory [31] maintains a list of disclosed vulnerabilities acquired by Zero Day Initiative [31] (one of the world's largest vendor-agnostic bug bounty programs), which provides CVE information and CVSS assessment.
- GitHub Advisory [7] is a new vulnerability database inclusive of CVEs and GitHub-originated security advisories beginning in 2017. This database provides CVE information, CVSS and CWE assessment, and affected packages with version ranges specified.
- CISA KEV [3] is a catalog of Known Exploited Vulnerabilities maintained by CISA. CVEs in this list are authoritatively confirmed to be exploited in the wild. Hence, KEV serves as a valuable input to vulnerability prioritization tasks.
- MITRE CWE [17] is a hierarchically organized list specialized for the root cause mapping of vulnerabilities. Records in vulnerability databases (e.g., NVD and GitHub Advisory) are correlated CWE entries via CWE identifiers.
- MITRE CAPEC (Common Attack Pattern Enumeration and Classification) [15] provides a dictionary of known attack patterns employed by adversaries to exploit known cyber weaknesses.
- MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) [18] is a knowledge base of adversary tactics and techniques based on real-world observations. ATT&CK has been widely used in threat modeling [1, 10]. Many techniques in ATT&CK have relationships with CAPEC entries.
- MITRE D3FEND [19] serves as the defensive counterpart framework against ATT&CK, providing a countermeasure knowledge base derived from research and development literature.

**Figure 1: Dataset Construction Process**

- Rapid7 AttackerKB [25] is a crowdsourcing knowledge base, where security researchers contribute detailed assessment analysis or attach tags (such as *Easy to weaponize*, *Gives privileged access*, and *Vulnerable in default configuration*) to vulnerabilities.
- OffSec Exploit-DB [27] is a CVE-compliant repository for exploits and proof of concepts<sup>1</sup>.
- Bugtraq [2] was a mailing list dedicated to cybersecurity issues, including vulnerabilities, vendor announcements, exploitations, and patches. It was shut down on January 31, 2021.
- Full-Disclosure [6] is another cybersecurity-related mailing list for discussions and vulnerability disclosures. It got a temporary interruption but resumed in 2014.
- OSS-Security [22] is a mailing list focusing on security issues in open-source projects.
- Linux-CVE-Announce [14] is a mailing list specialized for announcements of CVEs in Linux kernel.
- GitHub [8] hosts numerous open-source projects, as well as the patches for vulnerabilities discovered in these projects.
- git.kernel.org [13] hosts the source code repositories (including common commits and patches) of the Linux kernel.

## 2.2 Data Collection

Figure 1 illustrates the high-level overview of VulZoo's data collection process, demonstrating how raw data is transformed into a well-structured dataset. We introduce the process step by step.

**Source validation and data fetching.** The data sources for VulZoo are diverse and come in different formats. To handle them, we have customized four crawling strategies. (1) For sources with *Git-based repositories*, such as MITRE CVE, NVD, ZDI Advisory, GitHub Advisory, MITRE ATT&CK, OffSec Exploit-DB, and Linux-CVE-Announce, we configure them as *git submodules* within VulZoo repository, benefiting from the natural incremental data synchronization feature of Git. (2) For sources releasing files via *stable hyperlinks*, i.e., NVD (for CPE dictionary), CISA KEV, CWE, CAPEC, and D3FEND, we directly download the target files and decompress them if necessary. (3) For sources containing *discrete data with respective links*, i.e., Bugtraq, Full-Disclosure, and OSS-Security, we develop a mail crawler to first retrieve and parse mail links for each mail, and then download them individually. (4) For patches on *GitHub and git.kernel.org*, we develop a program to identify patch links in NVD by tags and then fetch the contents.

During the preliminary analysis, we observed many links in mailing lists and NVD are invalid. Additionally, there are still 1,605

<sup>1</sup>In this paper, we do not distinguish between exploits and proof of concepts, but refer to them as PoCs.

**Table 2: Descriptive Statistics of Data in VULZoo**

Format	Category	Measurement	Value
Structural	CVE Record	MITRE-recorded CVEs	320,861
		NVD-recorded CVEs	253,722
		ZDI-recorded CVEs	13,291
		GitHub-recorded CVEs	17,069
	Assessment	CPE Names	1,271,275
		CWE Weaknesses	963
		CVSS Metrics	N/A
		KEV	1,120
		AttackerKB Assessments	1,665
		CAPEC Attack Patterns	615
		ATT&CK Techniques	1062
		D3FEND Techniques	183
	PoC	Exploit-DB PoCs	46,540
	Mail	CVE-related Bugtraq Mails	17,404
		CVE-related Full-Disclosure Mails	12,448
		CVE-related OSS-Security Mails	14,976
		CVE-related Linux-CVE-Announce Mails	2,054
Non-structural	Patch	Patch Files	12,626

links tagged as *Patch* in NVD pointing to [www.securityfocus.com](http://www.securityfocus.com), which has been inactive for a long time. Therefore, our crawlers first validate the link status before proceeding with further operations. **Data filtering, deduplication, and format unification.** After fetching the raw data, we filter out irrelevant files (e.g., READMEs and CHANGELOGs) and retain only the vulnerability-related data to improve dataset readiness for programming usage. Following data filtering, we conduct SHA256-based file comparison for deduplication. We observe that there are respectively 71, 24, and 6 duplication cases among mails from Bugtraq, Full-Disclosure, and OSS-Security mailing lists. After deduplication, we convert all structural data from various formats into JSON to provide a unified data interface for future usage.

**Relationship mining and statistical analysis.** After the aforementioned processing, we get data in 4 types: structural JSON files, non-structural mails, patches, and PoC code snippets. To profile vulnerabilities from different aspects, we mine the relationships within the data to unveil and enhance data interconnection. Note that we deal with real-world vulnerabilities, not synthetic ones. As a result, not all CVEs have the same number of relationships to the other data. For example, PoCs do not exist for the majority of existing CVEs. We mine 11 types of relationships in all, covering all the edges in the topology graph shown in Figure 2. Once the dataset is ready, we generate the statistics for inspection.

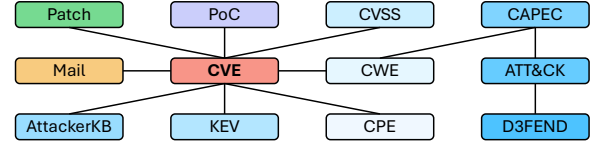
### 3 DATASET DESCRIPTION

In this section, we first provide a global overview of VULZoo’s contents. Following that, we take CVE-2020-7247 as an example to demonstrate how the assembled intelligence in VULZoo enables comprehensive profiling for vulnerabilities.

#### 3.1 Dataset Overview

VULZoo organizes the structural and non-structural data into 5 categories: *CVE Record*, *Assessment*, *PoC*, *Mail*, and *Patch*. The statistics for each category as of June 13, 2024, are presented in Table 2. We give a brief introduction to each category as follows:

- **CVE Record.** Each CVE record contains basic information regarding a CVE ID, such as descriptions, names, and versions of affected products. VULZoo integrates 4 different CVE databases

**Figure 2: Topology of VULZoo Dataset****Table 3: Descriptive Statistics of Relationships in VULZoo**

Relationship	Number
CVEs with CPE Names	224,998
CVEs with CWE Weaknesses	179,950
CVEs with CVSS Metrics	234,260
CVEs mentioned in KEV	1,120
CVEs with AttackerKB Assessments	1,108
CVEs with Exploit-DB PoCs	24,587
CVEs with Mails	42,030
CVEs with Patch Files	10,548
CWE – CAPEC	336 / 450
CAPEC – ATT&CK	177 / 36
ATT&CK – D3FEND	301 / 121

as data sources. Thus, there could be one or more pieces of information describing the same CVE. This benefits future research from two aspects. First, CVE instances from different databases contribute more or less unique information, such as CPE metrics in NVD, and independent CVSS assessment in GitHub Advisory. The composition works better than any single database. Second, since the information describing a CVE comes from various sources, VULZoo inherently provides a valuable resource for CVE differential analysis and alignment research.

- **Assessment.** This category includes 8 different vulnerability related assessment metrics. Specially, CPE is a huge structured naming scheme, commonly used to delineate the vulnerable products and version ranges regarding a CVE ID. Thanks to the rapid development of modern software engineering, CPE dictionary contains more than one million today and still grows very fast. CVSS is a special assessment metric used to capture the principal characteristics of a vulnerability as a vector string and produce a numerical score to reflect its severity. The vector string and the score are usually attached to CVE records. Hence, there is no value for CVSS in Table 2. For clarity, we omit the introductions to CWE, KEV, CAPEC, AttackerKB, ATT&CK, and D3FEND, as they have been introduced in § 2.1.
- **PoC.** Most of the PoC files are code snippets that could be executed to trigger a specific vulnerability. Some others may describe the manual process of reproducing a vulnerability.
- **Mail.** Each mail is stored as a text file recording the communications related to one or more CVEs.
- **Patch.** Each patch file is the code difference generated by one commit operation, with optional description messages.

Table 3 presents the 11 relationships mined from VULZoo. Specifically, the first 8 relationships center on CVE records and the last 3 ones are bipartite relationships among CWE, CAPEC, ATT&CK, and D3FEND. For example, there are 336 entries in the CWE catalog referring to CAPEC and 450 entries in the CAPEC catalog referring to CWE. For a better understanding of the relationship of data in

<b>CVE</b>	smtp_mailaddr in smtp_session.c in OpenSMTPD r6.6, as used in OpenBSD 6.6 ..., allows remote attackers to execute arbitrary commands as root via a crafted SMTP session...	
<b>CPE</b>	cpe:2.3:a:openbsd:opensmtpd:6.6:*:*:*:*:* (5 more CPE strings omitted)	
<b>CVSS</b>	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H (CRITICAL) CVSS:2.0/AV:N/AC:L/Au:N/C:C/I:C/A:C (HIGH)	
<b>CWE</b>	755: Improper Handling of Exceptional Conditions 78: Improper Neutralization of Special Elements used in an OS Command	
<b>KEV</b>	Added Time: 2022-03-25 Action Due Date: 2022-04-15 Name: OpenSMTPD Remote Code Execution Vulnerability Required Action: Apply updates per vendor instructions	
<b>AttackerKB</b>	Attacker-value: 5 (Very High) Exploitability: 5 (Very High) Tags: Easy to weaponize, Gives privileged access, Vulnerable in default configuration	
<b>Patch</b>	commit: 9dcfda045474d8903224d175907bfc29761dcb45 diff --git a/usr.sbin/smtpd/smtp_session.c b/usr.sbin/smtpd/smtp_session.c	
<b>PoC</b>	exploits/linux/remote/{48038.rb,47984.py} exploits/openbsd/remote/48051.pl	
<b>Mail</b>	2020-01-28 LPE and RCE in OpenSMTPD (CVE-2020-7247) 2020-01-30 [SECURITY] [DSA 4611-1] opensmtpd security update	
<b>CAPEC</b>	6: Argument Injection 15: Command Delimiters 88: OS Command Injection 43: Exploiting Multiple Input Interpretation Layers 108: Command Line Execution through SQL Injection	

Figure 3: CVE-2020-7247 Intelligence in VulZoo

VulZoo, we draw their connectivity as a topology graph, as shown in Figure 2. Here, nodes are the data listed in Table 2 and edges denote the relationship in Table 3.

### 3.2 Example: CVE-2020-7247

We employ CVE-2020-7247 to demonstrate VulZoo’s vulnerability profiling capability. Starting from the *CVE* node in Figure 2, we traverse the topology graph to collect all information in VulZoo describing it. Figure 3 shows the summarized result for this investigation, covering all nodes in Figure 2 except for ATT&CK and D3FEND. From the assembled intelligence, we know that CVE-2020-7247 is a critical command injection vulnerability in OpenSMTPD, with one patch file (modifying `smtp_session.c`) and three PoCs (written in Ruby, Python, and Perl) available. Researchers mark this CVE as having very high value and exploitability because it exists in the default configuration, is easy to exploit, and grants privileged access once exploited. CISA added this CVE into KEV and required remediation. Two mails contain the vulnerability analysis and remediation details. We argue that such an information composition profiles CVE-2020-7247 better than any single database.

## 4 APPLICATION SCENARIOS

Thanks to VulZoo’s large-scale data and heterogeneous characteristics, this dataset can support a wide range of vulnerability assessment and prioritization research. In this section, we present three application scenarios of VulZoo.

**Severity and Type Assessment.** Researchers can train statistical or deep learning models based on the extensive data in VulZoo to predict the CVSS score and vector for a new vulnerability. This goal can also be achieved by proposing and applying similarity (distance) based algorithms to measure the similarity (distance) between vulnerabilities in VulZoo and future ones. These approaches also can be applied to predict CWEs of unseen vulnerabilities.

**Intelligence Alignment.** As VulZoo has already integrated vulnerability information from multiple sources, researchers can directly conduct alignment studies on this dataset. For example, they can mine deep relationships and resolve inconsistencies among both structural and non-structural entities in VulZoo.

**Information Augmentation.** To facilitate the vulnerability management process, researchers can augment the vulnerability descriptions by deriving key aspects, such as affected files and exploitation status, from VulZoo, if they are missing by human analysts.

## 5 CONCLUSION

This paper presents VulZoo, a comprehensive vulnerability intelligence dataset containing heterogeneous vulnerability information derived from 17 data sources. For easier updates and usage, we have developed multiple utility scripts to automate data synchronization and cleaning, relationship mining, and statistics generation processes. We believe that future vulnerability assessment and prioritization research will benefit from our dataset.

## REFERENCES

- [1] Mohamed Ahmed, Sakshyam Panda, Christos Xenakis, and Emmanouil Panaousis. 2022. MITRE ATT&CK-driven cyber risk assessment. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*. 1–10.
- [2] Bugtraq. 2024. bugtraq — lists.openwall.net. <https://lists.openwall.net/bugtraq/>.
- [3] CISA. 2024. Known Exploited Vulnerabilities Catalog. <https://bit.ly/45QJJ33>.
- [4] Clément Elbaz, Louis Rilling, and Christine Morin. 2020. Fighting N-day vulnerabilities with automated CVSS vector prediction at disclosure. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*. 1–10.
- [5] FIRST. 2024. CVSS — first.org. <https://www.first.org/cvss/>.
- [6] Full-Disclosure. 2024. full-disclosure. <https://lists.openwall.net/full-disclosure/>.
- [7] GitHub. 2024. GitHub Advisory — github.com. <https://github.com/advisories>.
- [8] GitHub. 2024. GitHub: Let’s build from here — github.com. <https://github.com/>.
- [9] Hao Guo, Sen Chen, Zhenchang Xing, Xiaohong Li, Yude Bai, and Jiamou Sun. 2022. Detecting and augmenting missing key aspects in vulnerability descriptions. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 31, 3 (2022), 1–27.
- [10] Yi-Ting Huang, Chi Yu Lin, Ying-Ren Guo, Kai-Chieh Lo, Yeali S Sun, and Meng Chang Chen. 2021. Open source intelligence for malicious behavior discovery and interpretation. *IEEE Transactions on Dependable and Secure Computing* 19, 2 (2021), 776–789.
- [11] Intel471. 2023. Vulnerabilities Year-in-Review: 2023. <https://bit.ly/4evl9fX>.
- [12] Atika Lim. 2024. The true cost of cyber attacks in 2024 and beyond. <https://www.expressvpn.com/blog/the-true-cost-of-cyber-attacks-in-2024-and-beyond/>.
- [13] Linux. 2024. Kernel.org git repositories — git.kernel.org. <https://git.kernel.org/>.
- [14] Linux. 2024. linux-cve-announce.vger.kernel.org. <https://bit.ly/3zfgeC3>.
- [15] MITRE. 2024. CAPEC — capec.mitre.org. <https://capec.mitre.org/>.
- [16] MITRE. 2024. CVE Website — cve.org. <https://www.cve.org/>.
- [17] MITRE. 2024. CWE — cwe.mitre.org. <https://cwe.mitre.org/>.
- [18] MITRE. 2024. MITRE ATT&CK — attack.mitre.org. <https://attack.mitre.org/>.
- [19] MITRE. 2024. MITRE D3FEND — d3fend.mitre.org. <https://d3fend.mitre.org/>.
- [20] NIST. 2024. NVD - Home — nvd.nist.gov. <https://nvd.nist.gov/>.
- [21] NVD. 2024. NVD - CPE — nvd.nist.gov. <https://nvd.nist.gov/products/cpe>.
- [22] OSS-Security. 2024. oss-security. <https://www.openwall.com/lists/oss-security/>.
- [23] Shengyi Pan, Lingfeng Bao, Xin Xia, David Lo, and Shanping Li. 2023. Fine-grained commit-level vulnerability type prediction by CWE tree structure. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 957–969.
- [24] Yue Qin, Yue Xiao, and Xiaojing Liao. 2023. Vulnerability Intelligence Alignment via Masked Graph Attention Networks. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2202–2216.
- [25] Rapid7. 2024. AttackerKB — attackerkb.com. <https://attackerkb.com/>.
- [26] Bonan Ruan, Jiahao Liu, Chuqi Zhang, and Zhenkai Liang. 2024. KernJC: Automated Vulnerable Environment Generation for Linux Kernel Vulnerabilities. *arXiv preprint arXiv:2404.11107* (2024).
- [27] Offensive Security. 2024. Exploit Database. <https://www.exploit-db.com/>.
- [28] Qiushi Wu, Yang He, Stephen McCamant, and Kangjie Lu. 2020. Precisely characterizing security impact in a flood of patches via symbolic rule comparison. In *The 2020 Annual Network and Distributed System Security Symposium (NDSS’20)*.
- [29] J. Wunder, A. Kurtz, C. Eichenmüller, F. Gassmann, and Z. Benenson. 2024. Shedding Light on CVSS Scoring Inconsistencies: A User-Centric Study on Evaluating Widespread Security Vulnerabilities. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society.
- [30] Wei You, Peiyuan Zong, Kai Chen, Xiaofeng Wang, Xiaojing Liao, Pan Bian, and Bin Liang. 2017. Semfuzz: Semantics-based automatic generation of proof-of-concept exploits. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*. 2139–2154.
- [31] ZDI. 2024. Home — zerodayinitiative.com. <https://www.zerodayinitiative.com/>.