Cybersecurity Applications in Software: Data-Driven Software Vulnerability Assessment and Management

Jiao Yin, MingJian Tang, Jinli Cao, Mingshan You, and Hua Wang

1 Software Vulnerability Assessment and Management

With the ongoing adoption of information technology and its impact on national economies and society, software plays a key role in the daily life of both organizations and individuals. However, a growing number of vulnerabilities caused by poor design or overlooked implementation are being disclosed nowadays [1]. The insecurity of information technology is often inevitable, which is a side effect brought by the use of information technology [2].

The scale, type and destructiveness of cyber threats and cyberattacks are increasing year by year, as more and more software vulnerabilities are discovered and publicly disclosed. According to CVE details [3], more than 166,000 software vulnerabilities have been disclosed and archived from 1988 to the end of 2021. More vulnerabilities are available from various channels and venues (e.g., security bulletins, forums, social media and so on). Bilge and Dumitras pointed out that once a vulnerability is disclosed, the chance of being exploited increases by five orders of magnitude[4, 5]. Obviously, unpatched known vulnerabilities impose significant security risks to modern society. Considering the huge number of disclosed

J. Yin · J. Cao (✉)
Department of Computer Science and Information Technology, La Trobe University, Melbourne, VIC, Australia
e-mail: j.yin@latrobe.edu.au; j.cao@latrobe.edu.au

M. Tang
Westpac Banking Corporation, Sydney, NSW, Australia
e-mail: ming.tang@westpac.com.au

M. You · H. Wang
Institute for Sustainable Industries & Liveable Cities, Victoria University, Melbourne, VIC, Australia
e-mail: mingshan.you@live.vu.edu.au; Hua.Wang@vu.edu.au

K. Daimi et al. (eds.), *Emerging Trends in Cybersecurity Applications*,
https://doi.org/10.1007/978-3-031-09640-2_17

vulnerabilities, it is difficult for information system vendors and users to patch each vulnerability in a timely manner. Because of limited budget and resources, vulnerability assessment and management has become critical for both commercial organizations regardless of the size and the entire cybersecurity community to make contingency plans in advance.

This section lays the foundation of software vulnerability assessment and management by introducing the readers to some of the key concepts spanning from vulnerability lifecycle to the entire vulnerability ecosystem.

1.1 Vulnerability and Vulnerability Disclosure

Vulnerability is a term referring to a system flaw that can leave it open to attack. According to the Common Vulnerabilities and Exposures (CVE) consortium, it is formally defined as a weakness in the computational logic (e.g., code) found in software and some hardware components (e.g., firmware) that, when exploited, results in a negative impact on confidentiality, integrity or availability (CIA) [6].

Vulnerability disclosure is the practice of reporting security flaws in computer software or hardware [7]. Vulnerability can be disclosed by multiple parties, including but not limited to third-party or internal software developers, vendors, suppliers, cybersecurity professionals and cybersecurity researchers. Different parties have different preferences for vulnerability disclosure time. Software vendors, suppliers and related developers usually prefer to disclose vulnerabilities after the corresponding patches or remedies are available, while affected end-users, cybersecurity professionals and researchers tend to disclose vulnerabilities as early as possible.

1.2 Exploit and Exploitability

A typical exploit in the cybersecurity domain can be a piece of software, a chunk of data or a sequence of commands, which takes advantage of a bug or vulnerability to cause unintended or unanticipated behaviour [8].

Exploitation is the behaviour of using an exploit to abuse software, hardware or other electronic equipment, including things like gaining control of a computer system, allowing privilege escalation or launching a denial-of-service (DoS) attack [9].

Exploitability is the state or condition of being exploitable. In the cybersecurity domain, a vulnerability is identified as exploitable when the proof-of-concept of the corresponding exploit exists. Exploitability is an important vulnerability assessment metric to reflect the properties of the vulnerability that lead to a successful attack [10].

Table 1 Six vulnerability lifecycle events

Event	Occurred time	Available to public?
Creation	t_{creat}	No
Discovery	t_{disco}	No
Exploit available	t_{explo}	No
Disclosure	t_{discl}	Yes
Patch available	t_{patch}	Yes
Patch installation	t_{insta}	Yes

1.3 Lifecycle of a Vulnerability

Frei et al. described a typical vulnerability lifecycle in [11]. We simplify the main events of the lifecycle in Table 1. A typical vulnerability lifecycle consists of six events, namely creation, discovery, exploit available, disclosure, patch available and patch installation. It should be noted that the order of occurrence of these six events may be slightly different for individual vulnerabilities. For example, vulnerability exploitation may occur before disclosure.

When a vulnerability is disclosed, the vulnerability information has three features, namely free access, independence and validation [11]. Information about disclosed vulnerabilities is available to the public for free. Then, disclosed vulnerability information will be widely accepted and used by the entire cybersecurity community. Finally, the disclosed information undergoes a thorough assessment by a panel of security experts and some assessment results will also be added to the disclosed vulnerability as basic risk ratings.

The time period between vulnerability discovery and disclosure is called the pre-disclosure phase, denoted as $\triangle\, t_{disco}(v)$, where $\triangle t_{disco}(v) = t_{discl}(v) - t_{disco}(v)$; $t_{discl}(v)$ is the disclosure time of v and $t_{disco}(v)$ is the discovery time of v. At this stage, the newly discovered vulnerabilities remain largely private. If they are known by researchers or vendors, they can work to provide patches before they become exploitable or disclosed in public. However, once they are discovered by malicious intruders or cyber-criminals, the potential risk involved can be significantly elevated. However, in this pre-disclosure phase, few things can be done to stop exploitation.

The time period from disclosure to patch available is another important phase, namely the post-disclosure phase, denoted as $\triangle t_{patch}(v)$, where $\triangle t_{patch}(v) = t_{patch}(v) - t_{discl}(v)$ and $t_{patch}(v)$ is the patch available time of v. At this stage, the risks of exploitation soar because more parties, including hackers and cyber-criminals, know of the existence of and have detailed information on the vulnerability. To make matters worse, end-users of the affected products will also be aware of the existence of this vulnerability, which will undoubtedly bring great pressure to vendors and service providers. Therefore, it is crucial for vendors and security information providers (SIPs) to provide a patch or give effective security advice. This research focuses on improving exploitability predictions and analysis to better inform decision-makers to prioritize the most urgent and risky vulnerabilities.

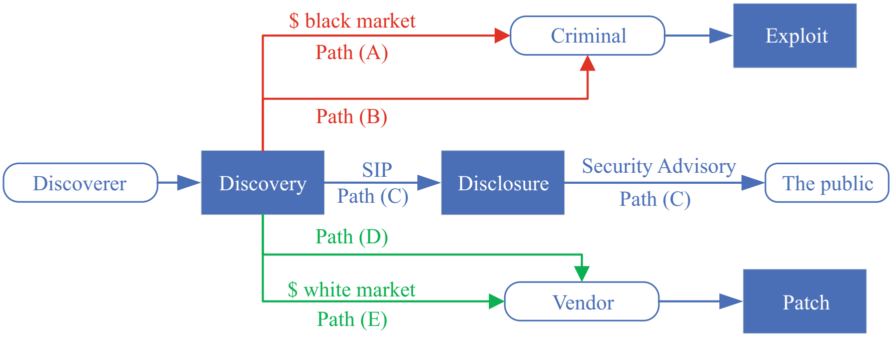


Fig. 1 Cybersecurity ecosystem illustration

Similarly, post-patch phase refers to the time period between vulnerability patch available and patch installation, which is denoted as $\triangle t_{insta}(v)$, where $\triangle t_{insta}(v) = t_{insta}(v) - t_{patch}(v)$ and $t_{insta}(v)$ is the patch instalment time of v. At this stage, if users are able to install the patch of v in a timely manner, the risks of exploitation can be mitigated.

1.4 Cybersecurity Ecosystem

Whenever a new vulnerability is discovered, various parties with different and often conflicting motivations and incentives become involved in a complex way [11]. Participants include but are not limited to discoverers, security advisories, cybercriminals, traders in white or underground black markets, vendors and the public. The so-called security ecosystem consists of these players and their interactions. Figure 1 provides a high-level view of a cybersecurity ecosystem.

As shown in Fig. 1, Path (A) and (B) in red are at high risk while path (D) and (E) in green have fewer security concerns. Most vulnerabilities go through path (C). Once disclosed, the security of a vulnerability is uncertain, depending on whether it is exploited by attackers or patched by vendors. The risk of exploitation soars after being disclosed, as described in [5], ‘after vulnerabilities are disclosed publicly, the volume of targeted attacks increases by five orders of magnitude’.

2 Mainstream Vulnerability and Exploit Databases

Historical vulnerability and exploit records are the most important and valuable digital assets for vulnerability assessment and management. Therefore, many commercial or non-profit organizations are collecting, storing and maintaining their own vulnerabilities and exploit databases. Some of them are available to the public.

This section introduces some well-known and publicly accessible databases and repositories. They are dedicated to comprehensive and credible information on vulnerabilities and potential links to detailed exploits (if exploitable).

2.1 CVE: Common Vulnerabilities and Exposures database

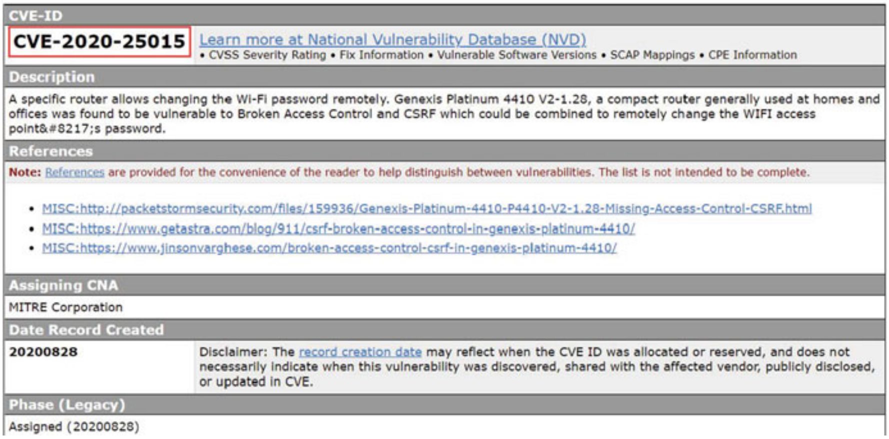
At present, vulnerability disclosure sources mainly include individual vendors, cybersecurity forums and open-source databases. Each disclosed vulnerability will be assigned a unique identification code, CVE-ID. CVE-ID is widely accepted by both local individual information providers/repositories and multiple global vulnerability databases [12]. This unique CVE-ID of each vulnerability facilitates the fast and accurate integration of data across multiple information sources and databases. In other words, it can be used to retrieve and link various information of the same vulnerability from different databases. Apart from CVE-ID, vulnerability disclosure reports may also include disclosure date, the names and corresponding version numbers of affected software products, required permission, the scope of impact and repair suggestions etc. [12].

The CVE database is one of the most well-known vulnerability databases. It stores essential disclosed vulnerability information, such as the CVE-ID, description, one or more public reference links [13]. A vulnerability description is a brief paragraph on each vulnerability, which contains abundant details such as the vulnerability type, names of affected products and vendors, a summary of affected versions, the impact, the access that an attacker requires to exploit the vulnerability and the important code components or inputs that are involved [14]. Depending on the source of disclosure, the description of a software vulnerability is usually written by the party requesting its CVE-ID.

The information in the CVE database serves as the baseline for vulnerability disclosure, and is referenced by many vulnerability databases, security products and services. The vulnerability list in the CVE database organized by year is available for download in several formats, i.e., comma separated format, HTML, text and XML. More than 160,000 vulnerability entries spanning over 20 years from 1999 to the present are included in the CVE database. The CVE database provides multiple attributes of each vulnerability for the public, such as Description, References, Assigning CNA, Date Record Created and Phase. Figure 2 shows a screenshot of vulnerability information listed in the CVE database. For more information, refer to the official website of the CVE database https://cve.mitre.org/index.html.

2.2 NVD: National Vulnerability Database

The NVD database is the U.S. government repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP) [15]. It provides an analysis on CVE entries that have been published to



CVE-ID

CVE-2020-25015 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

A specific router allows changing the Wi-Fi password remotely. Genexis Platinum 4410 V2-1.28, a compact router generally used at homes and offices was found to be vulnerable to Broken Access Control and CSRF which could be combined to remotely change the WIFI access point&#8217;s password.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:http://packetstormsecurity.com/files/159936/Genexis-Platinum-4410-P4410-V2-1.28-Missing-Access-Control-CSRF.html
- MISC:https://www.getastra.com/blog/911/csrf-broken-access-control-in-genexis-platinum-4410/
- MISC:https://www.jinsonvarghese.com/broken-access-control-csrf-in-genexis-platinum-4410/

Assigning CNA

MITRE Corporation

Date Record Created

20200828 Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

Phase (Legacy)

Assigned (20200828)

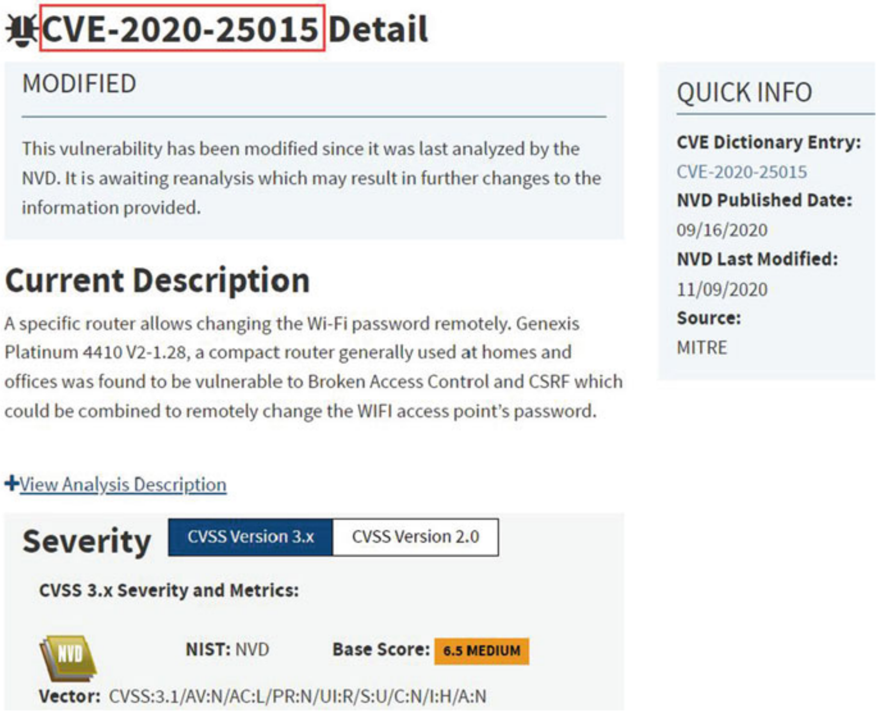
Fig. 2 A vulnerability listed in the CVE database

the CVE database. Based on the descriptions and references provided by the CVE database and other publicly accessed supplemental data, NVD expert panellists conduct an initial vulnerability assessment and give results based on certain standards, such as impact metrics (defined by Common Vulnerability Scoring System (CVSS)), applicability statements (defined by Common Platform Enumeration (CPE)), vulnerability types (defined by Common Weakness Enumeration (CWE)), and also other pertinent metadata [15]. Figure 3 shows the screenshot of information on a vulnerability listed in the NVD database.

Most importantly, the NVD database keeps re-analysing vulnerabilities as time and resources change over time to ensure the information provided by NVD is up to date. The NVD database is updated periodically to maintain the accuracy and real-timeness of vulnerability information and the data feeds in NVD database is available to the public for free [16].

2.3 CVE Details

CVE Details is a website developed by security consultant Serkan Özkan, who wanted to find an easy-to-use list of security vulnerabilities [3]. CVE Details contains information from multiple sources, including NVD XML data feeds, the Exploit Database [17], software vendor statements and additional vendor-supplied data, and Metasploit modules [3]. CVE Details presents each vulnerability entry on a single, easy-to-use web page. Figure 4 shows an example of the vulnerability information listed in CVE Details. Some statistics on vulnerabilities, vendors, products and exploits are also available in tables or figures [3].



CVE-2020-25015 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

A specific router allows changing the Wi-Fi password remotely. Genexis Platinum 4410 V2-1.28, a compact router generally used at homes and offices was found to be vulnerable to Broken Access Control and CSRF which could be combined to remotely change the WIFI access point's password.

+View Analysis Description

Severity

CVSS 3.x Severity and Metrics:

NIST: NVD

Base Score: 6.5 MEDIUM

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:H/A:N

QUICK INFO

CVE Dictionary Entry:
CVE-2020-25015

NVD Published Date:
09/16/2020

NVD Last Modified:
11/09/2020

Source:
MITRE

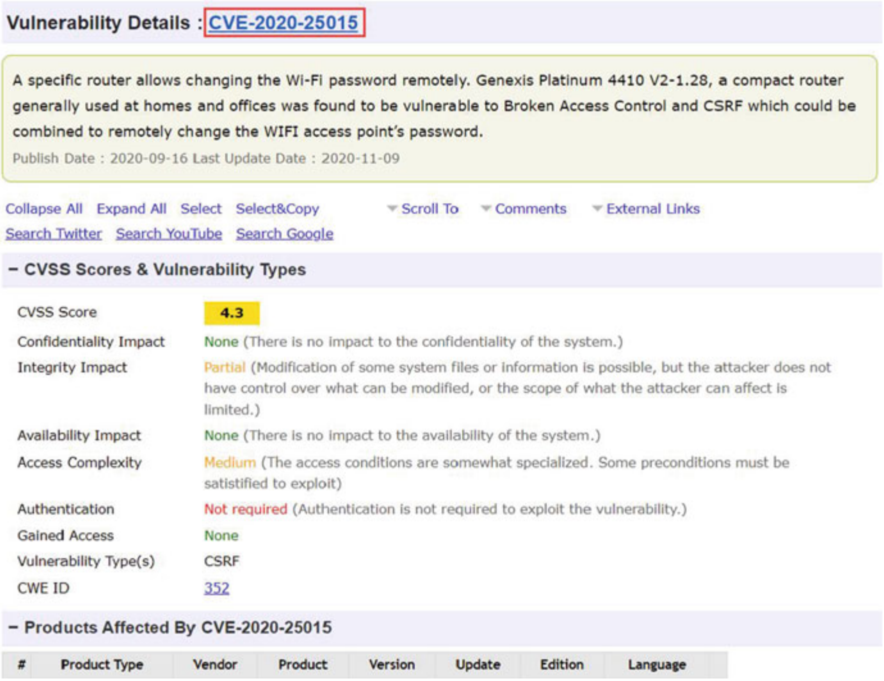
Fig. 3 A vulnerability listed in the NVD database

2.4 EDB: Exploit Database

The Exploit Database is an archive of public exploits and their targeted vulnerabilities, developed for use by penetration testers and vulnerability researchers [17]. The exploits in EDB are gathered from public sources and are freely available to the public. Each exploit in the EDB database has a unique EDB-ID for identification purposes.

The EDB database provides proofs-of-concept rather than advisories for vulnerabilities. Therefore, many researchers use the existence of exploits as the first sign of the exploitability of vulnerabilities [8, 18, 19], although exploitations always appear behind the existence of exploits. Figure 5 shows a screenshot of information on an exploit listed in the EDB database. Apart from the proof-of-concepts' executive codes, other crucial information on an exploit is also provided, such as EDB-ID, CVE-ID, Author, Type and Platform, as shown in Fig. 5. The EDB database is also a CVE-compatible database, making it possible to link the information of vulnerabilities and exploits.

At present, most commercial vulnerability management systems regularly synchronize the vulnerability and exploit information from these mainstream databases.



Vulnerability Details : CVE-2020-25015

A specific router allows changing the Wi-Fi password remotely. Genexis Platinum 4410 V2-1.28, a compact router generally used at homes and offices was found to be vulnerable to Broken Access Control and CSRF which could be combined to remotely change the WIFI access point's password.

Publish Date : 2020-09-16 Last Update Date : 2020-11-09

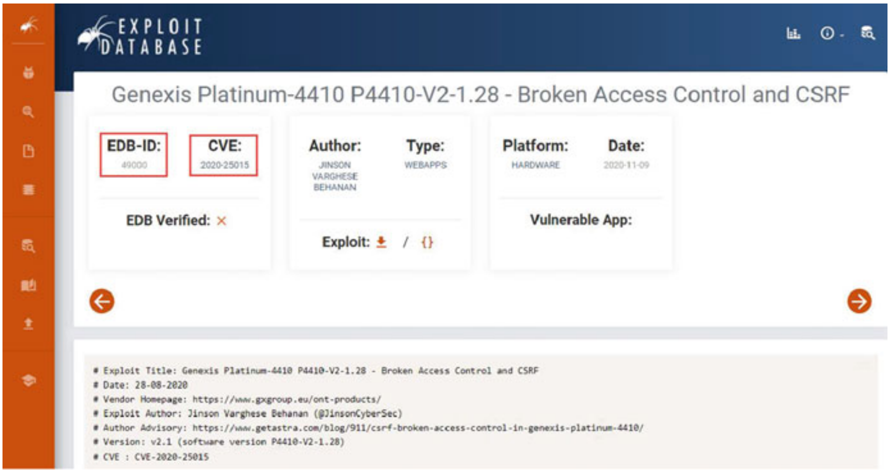
– CVSS Scores & Vulnerability Types

CVSS Score	4.3
Confidentiality Impact	None (There is no impact to the confidentiality of the system.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	None (There is no impact to the availability of the system.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satistified to exploit)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	CSRF
CWE ID	352

– Products Affected By CVE-2020-25015

#	Product Type	Vendor	Product	Version	Update	Edition	Language

Fig. 4 A vulnerability listed in CVE details



Genexis Platinum-4410 P4410-V2-1.28 - Broken Access Control and CSRF

EDB-ID: 49000

CVE: 2020-25015

Author: JINSON VARGHESE BEHANAN

Type: WEBAPPS

Platform: HARDWARE

Date: 2020-11-09

EDB Verified: ✗

Exploit: / {}

Vulnerable App:

```
# Exploit Title: Genexis Platinum-4410 P4410-V2-1.28 - Broken Access Control and CSRF
# Date: 28-08-2020
# Vendor Homepage: https://www.gxgroup.eu/ont-products/
# Exploit Author: Jinson Varghese Behanan (@JinsonCyberSec)
# Author Advisory: https://www.getastra.com/blog/911/csrf-broken-access-control-in-genexis-platinum-4410/
# Version: v2.1 (software version P4410-V2-1.28)
# CVE : CVE-2020-25015
```

Fig. 5 An exploit listed in the EDB database

Furthermore, the experimental data of most research papers on vulnerability risk assessment come from the integration results of these open-sourced mainstream databases [4, 20–23]. To provide more examples for reference, Table 2 lists the

Table 2 Examples of vulnerabilities and their corresponding exploits listed in CVE, NVD, CVE Details and EDB

CVE-ID/EDB-ID	Database	URL
CVE-2020-25015	CVE	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-25015
	NVD	https://nvd.nist.gov/vuln/detail/CVE-2020-25015
	CVE details	https://www.cvedetails.com/cve-details.php?cve_id=CVE-2020-25015
EDB-ID: 49000	EDB	https://www.exploit-db.com/exploits/49000
CVE-2021-24275	CVE	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-24275
	NVD	https://nvd.nist.gov/vuln/detail/CVE-2021-24275
	CVE details	https://www.cvedetails.com/cve-details.php?cve_id=CVE-2021-24275
EDB-ID: 50346	EDB	https://www.exploit-db.com/exploits/50346
CVE-2021-24287	CVE	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-24287
	NVD	https://nvd.nist.gov/vuln/detail/CVE-2021-24287
	CVE details	https://www.cvedetails.com/cve-details.php?cve_id=CVE-2021-24287
EDB-ID: 50349	EDB	https://www.exploit-db.com/exploits/50349
CVE-2021-24286	CVE	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-24286
	NVD	https://nvd.nist.gov/vuln/detail/CVE-2021-24286
	CVE details	https://www.cvedetails.com/cve-details.php?cve_id=CVE-2021-24286
EDB-ID: 50350	EDB	https://www.exploit-db.com/exploits/50350

Uniform Resource Locator (URL) of examples of more vulnerabilities or exploits contained in the aforementioned four databases. The URL is entered into a browser for detailed information corresponding to that vulnerability or exploit. It is worth mentioning that the exploit listed below each vulnerability in Table 2 is the specific exploit attacking that vulnerability.

3 Common Vulnerability Scoring System

Vulnerability management is a crucial measure for both organizations and the entire cybersecurity community to protect their information systems and networks from cyberattacks, intrusions, malware and various types of data breaches [23]. Since the availability of exploits is much more in quantity than the availability of patches [24], it is important for vulnerability management experts to accurately assess the risk level of existing vulnerabilities. For risk management and vulnerability repair of modern information systems, vulnerability assessment and prioritization are the most basic steps in order to allocate budget and resources efficiently and effectively [25].

To date, various methods have been developed and introduced to assess software vulnerabilities and predict the trends of vulnerability outbreaks [23, 26]. Among them, CVSS plays the role of the de facto standard to assess the severity of software vulnerabilities in industry. It originated from a research project which aimed to promote a common understanding of vulnerabilities and their impact through the development of a common vulnerability scoring system by the National Infrastructure Advisory Council (NIAC) in July 2003 [27]. CVSS is currently at version 3.1 and under the custody of the Forum of Incident Response and Security Teams (FIRST). As a premier organization and recognized global leader in incident response, currently, FIRST has more than 400 members ranging from government, commercial and educational organizations, spread over Africa, the Americas, Asia, Europe and Oceania [28, 29]. Nowadays, CVSS is recommended by a large number of hardware and software vendors, such as Cisco, Oracle and Microsoft [30].

3.1 CVSS Metric Groups

CVSS defines three independent metric groups, namely the base metric group, temporal metric group and environmental metric group, whose detailed metric names are shown in Table 3 [10]. Only the base metric group is mandatory for the calculation of a vulnerability CVSS score.

3.2 CVSS Scores

The values of CVSS metrics shown in Table 3 are either a number between 0–10 or a discrete categorical value, which are given by a cybersecurity experts panel

Table 3 Metrics in CVSS metric groups

Metric group	Metric name (and abbreviated form)
Base metric group	Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), User Interaction (UI), Scope (S), Confidentiality (C), Integrity (I), Availability
Temporal metric group	Exploit Code Maturity (E), Remediation Level (RL), Report Confidence (RC)
Environmental metric group	Confidentiality Requirement (CR), Integrity Requirement (IR), Availability Requirement (AR), Modified Attack Vector (MAV), Modified Attack Complexity (MAC), Modified Privileges Required (MPR), Modified User Interaction (MUI), Modified Scope (MS), Modified Confidentiality (MC), Modified Integrity (MI), Modified Availability (MA)

according to the basic information of disclosed vulnerabilities [10]. Based on these metric groups, CVSS then calculates an overall score between 0–10.0 as the final CVSS score of a vulnerability according to a specially designed formula, where 10.0 represents the highest risk [4]. The detailed calculation process can be found in [10].

In particular, CVSS includes a formula to calculate the exploitability score of a vulnerability, as shown in Eq. (1) [10],

$$\text{Exploitability} = 8.22 \times \text{AV} \times \text{AC} \times \text{PR} \times \text{UI}, \tag{1}$$

where 8.22 is the coefficient assigned by a panel of CVSS cybersecurity experts; AV, AC, PR and UI are the abbreviated forms of the four base metrics listed in Table 3.

In addition to an overall score between 0 and 10, CVSS also provides a qualitative evaluation method for vulnerabilities by mapping the overall CVSS score to five risk levels, namely none (0.0), low (0.1–3.9), medium (4.0–6.9), high (7.0–8.9) and critical (9.0–10.0).

CVE Details presents the current vulnerability distribution by CVSS scores based on 162,031 vulnerabilities, which shows the weighted average CVSS score for all disclosed vulnerabilities is 6.5 [3].

3.3 Limitations of CVSS

CVSS is a carefully designed scoring system based on expert knowledge and has been accepted by a wide range of organizations. However, it is widely questioned by researchers that an overall score calculated by combining multiple metric groups with fixed weights, such as a CVSS score, can accurately represent the risk level of different software vulnerabilities [4].

Furthermore, CVSS is widely criticized by the academic community for the inconsistency between CVSS scores and the exploitability of vulnerabilities [8, 12, 19]. The overall CVSS scores of existing disclosed vulnerabilities show that there is no significant correlation between the CVSS score of a vulnerability and the possibility of its exploitability.

To further validate the criticism of CVSS in the academic community, the work in [4] visualizes two CVSS metrics, namely base score and exploitability score, that are most relevant to the exploitability of vulnerabilities from two CVSS versions (CVSS V2.0 and V3.0), as shown in Fig. 6. The data samples in Fig. 6 are from all disclosed vulnerabilities recorded in the NVD database from 1988 to 2019. Figure 6 shows exploited vulnerabilities in dark orange and unexploited vulnerabilities in the navy. The Y-axis represents the comparison of the number of these two types of vulnerabilities with the same CVSS metric score. The X-axis indicates the value of the corresponding CVSS metrics and the larger the value, the greater the risk of the corresponding vulnerabilities. Taking the V2 exploitability score shown in subplot (b) as an example, the blue bar with a score between 8 and 10 is very high, indicating that the number of unexploited vulnerabilities in this interval is very

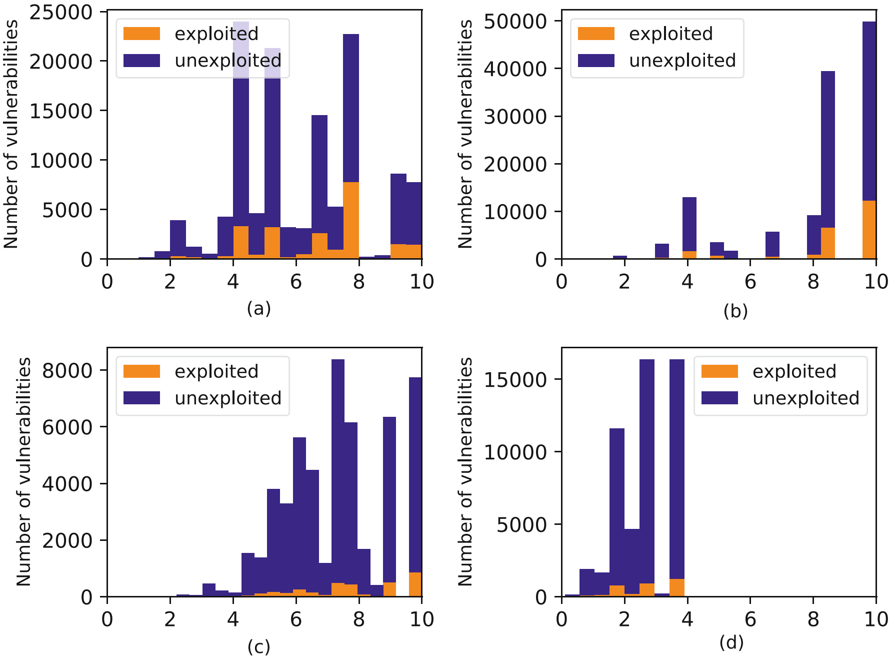


Fig. 6 The CVSS metric score distribution of vulnerabilities disclosed from 1988 to 2019 . (**a**) V2 base score. (**b**) V2 exploitability score. (**c**) V3 base score. (**d**) V3 exploitability score

large. Obviously, this contradicts the low probability of unexploited vulnerabilities. Similarly, the contradiction between the CVSS metric score and the exploitability of vulnerabilities is also reflected in subplots (a), (c) and (d). It is worth noting that V3 is an improved CVSS version of V2. However, as can be seen from subplots (c) and (d), the deficiency that CVSS cannot effectively depict the exploitability of vulnerabilities has not been significantly improved.

Other concerns on the application of CVSS as a vulnerability assessment indicator include the following two points. Firstly, the value assignment of CVSS metrics relies on an expert panel, which is costly in time and money. Furthermore, it is difficult to ensure consistency when the personnel changes.

4 Vulnerability Exploitability Prediction and Analysis

Vulnerability exploitability prediction and analysis is one of the most important tasks in vulnerability assessment and management. Considering the inaccuracy of the CVSS Exploitability score calculated by Eq. (1), researchers in the academic community have done a significant amount of work in vulnerability exploitability prediction. This section introduces some representative work in three aspects, see Sects. 4.1, 4.2 and 4.3 for details.

4.1 Vulnerability Exploitability Prediction

The exploitability of a vulnerability indicates if a vulnerability will be exploited or not. With the accumulation of more and more historical data, researchers adopted a variety of machine learning and deep learning models and algorithms to predict the exploitability of vulnerabilities and very promising results are reported. Vulnerability features can be extracted from publicly available information, including descriptions, CVSS metrics, social media streams, etc. [12, 31, 32].

As one of the most influential early works, the work in [12] extracted text features for vulnerabilities using a kind of one-hot representation. Specifically, a dictionary containing important tokens for vulnerability exploitation prediction was formed and if a token in the dictionary appears in the text fields of disclosed vulnerability information, the corresponding position is set to 1 otherwise 0. The achieved classification accuracy in [12] is nearly 85% with the linear support vector machine (LSVM) classifier.

Sabottke, Suciu and Dumitras [31] proposed a Twitter-based exploit detector, predicting real-world vulnerability exploitations. In this work, they manually extract statistical features from Twitter streams and other open-source databases, like NVD and OSVDB.

Along the same line, the work in [32] proposed an exploitability prediction method based on neural language models. Instead of extracting linguistic features using traditional TF-IDF-based representation, it adopts the neural language models to learn word embeddings based on the corpus collected from multiple sources. The experimental results show that the high-dimensional word embedding features extracted by the deep learning language model have better performance on the vulnerability exploitability prediction problem than features extracted by traditional statistical-based text feature extraction methods [32].

The authors in [19] considered two risk factors: (1) the existence of a public proof-of-concept exploit; (2) the existence of an exploit traded in the cybercrime black markets to evaluate the possibility of exploitation using a case-control study methodology.

According to the historical records in the NVD database and EDB database, the work in [33] established a machine learning model to automatically predict the vulnerability of unseen vulnerabilities. The authors compared the impact of different vulnerability feature sources on predictive performance. Results showed that the features extracted from text information such as vulnerability descriptions and external references are the most effective. On the premise that the above-mentioned features have been extracted, features such as CVSS metrics are redundant.

Jacobs, Romanosky et al. proposed an Exploit Prediction Scoring System (EPSS), which has the capability to predict if a vulnerability will be exploited or not in the wild within one year after disclosure [34]. The authors claimed that their system is simple to implement and therefore can be updated in a timely manner when new data becomes available.

The work in [4] proposed a deep neural language model based framework for vulnerability exploitability prediction. They apply the transfer learning technique and fine-tune a widely used pre-trained NLP model, Bidirectional Encoder Representations from Transformers (BERT), on the corpus consisting of vulnerability descriptions to extract domain-specific semantic features from vulnerability descriptions only. The extracted semantic features are fed into a pooling layer and an LSTM classification layer for the final decision-making. The experiments showed that their method achieved 91% in accuracy on a balanced real-world dataset.

4.2 Online Vulnerability Exploitability Prediction

The aforementioned research work treated vulnerability exploitability prediction as an offline machine learning problem. However, the reality is that the features and patterns of vulnerabilities, exploitation and the latent relationship between them are dynamically changing with the development of technology. In practical vulnerability exploitability prediction systems, if the possible concept drift problem is not considered, the performance of the predictive model will get worse and worse along time. Therefore, online learning models for vulnerability exploitability prediction have become a new trend.

The authors in [35] pointed out that exploitability assessment suffers from a class bias because ‘not exploitable’ labels could be inaccurate over time. Therefore, they proposed a new metric, called Expected Exploitability (EE) to provide a time-varying view of exploitability. In this work, they characterized the noise in exploit prediction as a class- and feature-dependent label noise and developed techniques to incorporate noise robustness into learning EE by capitalizing on domain-specific observations. Furthermore, instead of extracting features by technical analysis on existing metrics, they designed novel feature sets from previously under-utilized artefacts which are published after the disclosure of vulnerabilities, such as technical write-ups, social media discussions and proof-of-concept exploits. Experiment results on a dataset of 103,137 vulnerabilities showed an increase of precision from 49% to 86% was achieved by EE over existing metrics.

The authors in [20] also noticed an ‘actual drift’ problem existing in vulnerability exploitability labels, which means that the exploitability of vulnerabilities is chronologically variable. An ‘unexploitable’ vulnerability can become ‘exploitable’ after several days, months or years. In this work, based on the fact that vulnerability exploitability labels may change from unexploitable to exploitable over time, they proposed an algorithm called class rectification strategy (CRS) to detect the conceptual drift of vulnerability exploitability labels. Furthermore, they improved the real-time performance of the predictive model by updating the model online with vulnerabilities that have experienced label drift.

For the vulnerability exploitability prediction problem, on the whole, unexploitable vulnerabilities are far more than exploitable ones. This class unbalanced state changes dynamically in online learning scenarios. The work in [20] discussed how to improve the performance of vulnerability exploitability prediction under an online learning setting. It proposed a balanced window strategy (BWS) to build a dynamic class-balanced dataset to update the predictive model periodically. The experiment results show that BWS is effective in improving the online exploitability prediction performance of a variety of classifiers.

4.3 Vulnerability Exploitation Time Prediction

A vulnerability can have multiple exploits. The exploitation time of a vulnerability discussed in this section refers to the time difference between the earliest disclosure time of its exploits and the disclosure time of the vulnerability itself [22]. Figure 7 shows the exploitation time distribution of 23,302 vulnerabilities disclosed from 1988 to 2020 [22]. Exploitation time prediction is a more valuable and challenging task than exploitability prediction, which can provide a prediction of how soon a vulnerability will be exploited with an exact exploitation time or an exploitation time range. As shown in Fig. 7, the exploitation time varies in a large range in a biased distribution, which makes it challenging to make a prediction.

As early as 2010, the work in [12] began to study the problem of vulnerability exploitation time prediction. Instead of predicting a specific date, they reported the possible exploitation time in a weekly and monthly manner. An overall cumulative error rate of 15% was reported at the end of online training with a simple linear classier, which is extremely promising.

Sabottke, Suciu and Dumitras [31] proposed a Twitter-based exploit detector, predicting real-world vulnerability exploitations. In this work, they extract features from NVD, OSVDB and Twitter streams and manually select features based on the mutual information between features and labels. One of the contributions of

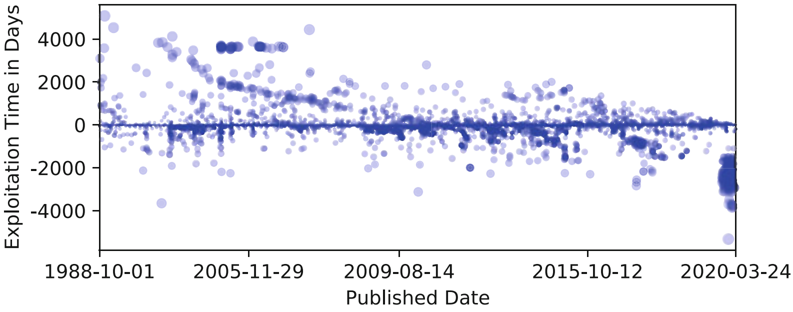


Fig. 7 The distribution of exploitation time

this work is that it not only investigated the vulnerability exploitability prediction problem but can also predict the emergence of exploits on an average of two days in advance.

The work in [21] and [22] investigated the vulnerability exploitation time prediction problem. Specifically, they divided the exploitation time into three classes, Neg, ZeroDay and Pos, based on the time differences between a vulnerability being exploited and being published. Although promising results were reported, more fine-grained exploitation time prediction results are expected.

5 Summary

Rigorous vulnerability evaluation and assessment empowers organizations to make informed and data-driven risk management decisions towards better mitigation and security of their IT environment against malicious exploitations. This chapter touches on cybersecurity applications in software vulnerability assessment and management. Specifically, this chapter includes:

(1) the key concepts, background, significance and foundations of software vulnerability evaluation and assessment;
(2) the valuable digital assets for both industry applications and academic research, the mainstream vulnerability and exploit databases, namely CVE, NVD, CVE Details and EDB;
(3) some of the latest advances as well as open challenges on vulnerability assessment and evaluation in both industry and academia community;
(4) further introduction and review on one of the research hotspots, vulnerability exploitability prediction and analysis.

Previous research works provided some promising solutions for vulnerability assessment and management. However, there are still many unsolved challenges. Some future directions in improving vulnerability assessment and management are listed as follows:

(1) Explore the exploitation time prediction with finer granularity. As the early attempts to predict vulnerability exploitation time, the work in [20–22] divided exploitation time into three classes, Neg, ZeroDay and Pos, based on the time differences between a vulnerability being exploited and being published. Although their prediction results are more detailed than exploitability prediction, a finer-granular exploitation time prediction can be more useful in practice, especially for Pos vulnerabilities. For example, the predicted exploitation time period can be yearly, monthly, weekly or even daily. The main challenge of finer granularity comes from data deficiencies and data imbalance within each granularity. With the increase of available vulnerabilities and exploit data and the development of unsupervised learning techniques, novel solutions will emerge in the future.

(2) Combine the exploitability prediction with other vulnerability assessment metrics to form a more comprehensive vulnerability risk evaluation model. In addition to exploitability, the risk level of a vulnerability is affected by many other aspects, such as the number of devices and users affected, the business process affected and the cost comparison between exploitation and remediation. CVSS is an example of a comprehensive vulnerability risk evaluation model. However, its effectiveness is far from satisfactory. More accurate availability prediction is undoubtedly conducive to vulnerability risk assessment, but how to combine the exploitability prediction results with other risk factors to form a holistic and effective vulnerability risk assessment framework will be grand challenging.

(3) Construct cybersecurity domain specific knowledge graph and explore more knowledge graph powered vulnerability intelligence applications. So far, the major source of vulnerabilities and exploits for research works and industry applications comes from existing well-organized databases, such as NVD, EDB and CVE Details. However, there are still vast amounts of vulnerability raw data from multimodal information sources, such as social media, software vendors, technical forums. This information can be used to build comprehensive cybersecurity knowledge graph. Based on such a domain-specific knowledge graph, novel knowledge-driven applications can be nurtured, including but not limited to subgraph matching to discover multi-stage and highly sophisticated cyberattack tactics and multi-hop question-and-answer systems, which can make highly specialized cyber knowledge more accessible.

References

1. M. Tang, M. Alazab, Y. Luo, Big data for cybersecurity: Vulnerability disclosure trends and dependencies. IEEE Trans. Big Data **5**(3), 317–329 (2017)
2. R. Anderson, T. Moore, The economics of information security. Science **314**(5799), 610–613 (2006)
3. S. Özkan, CVE details, the ultimate security vulnerability database (2021). https://www.cvedetails.com/, [Retrieved: Nov, 2021]
4. J. Yin, M. Tang, J. Cao, H. Wang, Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description. Knowl. Based Syst., 106529 (2020)
5. L. Bilge, T. Dumitraş, Before we knew it: an empirical study of zero-day attacks in the real world, in *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (Raleigh North Carolina, USA, 2012), pp. 833–844
6. The MITRE Corporation, About CVE - terminology. https://cve.mitre.org/about/terminology.html, [Retrieved: Nov, 2021]
7. L. Rosencrance, Vulnerability disclosure (2017). https://searchsecurity.techtarget.com/definition/vulnerability-disclosure, [Retrieved: Nov, 2021]
8. A. Younis, Y.K. Malaiya, I. Ray, Assessing vulnerability exploitability risk using software properties. Softw. Qual. J. **24**(1), 159–202 (2016)
9. Wikipedia, Exploit (computer security). https://en.wikipedia.org/wiki/Exploit_(computer_security), [Retrieved: Nov, 2021]

10. Forum of Incident Response and Security Teams, Common vulnerability scoring system v3.1: Specification document. https://www.first.org/cvss/v3.1/specification-document, [Retrieved: Nov, 2021]
11. S. Frei, D. Schatzmann, B. Plattner, B. Trammell, Modeling the security ecosystem-the dynamics of (in) security, in *Economics of Information Security and Privacy*, London, England, 2010, pp. 79–106
12. M. Bozorgi, L.K. Saul, S. Savage, G.M. Voelker, Beyond heuristics: learning to classify vulnerabilities and predict exploits, in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 2010, pp. 105–114
13. The MITRE Corporation, The mission of the cve program is to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities. https://cve.mitre.org/, [Retrieved: Nov, 2021]
14. The MITRE Corporation, Cve - frequently asked questions (2021). https://cve.mitre.org/about/faqs.html#cve_entry_descriptions_created, [Retrieved: Nov, 2021]
15. National Institute of Standards and Technology, U.S. Department of Commerce, General information. https://nvd.nist.gov/general, [Retrieved: Nov, 2021]
16. National Institute of Standards and Technology, U.S. Department of Commerce, NVD data feeds. https://nvd.nist.gov/vuln/data-feeds, [Retrieved: Nov, 2021]
17. Offensive Security, Exploit database (2021). https://www.exploit-db.com/, [Retrieved: Nov, 2021]
18. B.L. Bullough, A.K. Yanchenko, C.L. Smith, J.R. Zipkin, Predicting exploitation of disclosed software vulnerabilities using open-source data, in *Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics* (Scottsdale, USA, 2017), pp. 45–53
19. L. Allodi, F. Massacci, Comparing vulnerability severity and exploits using case-control studies. ACM Trans. Inf. Syst. Secur. (TISSEC) **17**(1), 1–20 (2014)
20. J. Yin, M. Tang, J. Cao, H. Wang, M. You, A real-time dynamic concept adaptive learning algorithm for exploitability prediction. Neurocomputing, 1–36 (2021)
21. J. Yin, M. Tang, J. Cao, H. Wang, M. You, Y. Lin, Vulnerability exploitation time prediction: an integrated framework for dynamic imbalanced learning. World Wide Web, 1–23 (2021)
22. J. Yin, M. Tang, J. Cao, H. Wang, M. You, Y. Lin, Adaptive online learning for vulnerability exploitation time prediction, in *Web Information Systems Engineering – WISE 2020*, Amsterdam, Netherlands, 2020, pp. 252–266
23. M. Tang, J. Yin, M. Alazab, J.C. Cao, Y. Luo, Modelling of extreme vulnerability disclosure in smart city industrial environments. IEEE Trans. Ind. Inf., 4150–4158 (2020)
24. S. Frei, M. May, U. Fiedler, B. Plattner, Large-scale vulnerability analysis, in *Proceedings of the 2006 SIGCOMM Workshop on Large-Scale Attack Defense*, 2006, pp. 131–138
25. L. Allodi, M. Cremonini, F. Massacci, W. Shim, The effect of security education and expertise on security assessments: The case of software vulnerabilities. Preprint (2018). arXiv:1808.06547
26. M. Alazab, M. Tang, *Deep Learning Applications for Cyber Security* (Springer Nature Switzerland AG, Cham, Switzerland, 2019)
27. M. Schiffman, A. Wright, D. Ahmad, G. Eschelbeck, The common vulnerability scoring system, in *National Infrastructure Advisory Council, Vulnerability Disclosure Working Group, Vulnerability Scoring Subgroup*, San Francisco, USA, 2004
28. Forum of Incident Response and Security Teams, Forum of incident response and security teams (first) (2021). https://www.cybersecurityintelligence.com/forum-of-incident-response-and-security-teams-first-5620.html, [Retrieved: Nov, 2021]
29. Forum of Incident Response and Security Teams, FIRST is the global forum of incident response and security teams (2021). https://www.first.org/, [Retrieved: Nov, 2021]
30. Oracle, Use of common vulnerability scoring system (CVSS) by oracle. https://www.oracle.com/technetwork/topics/security/cvssscoringsystem-091884.html, [Retriveved: Nov, 2021].
31. C. Sabottke, O. Suciu, T. Dumitras, Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits, in *24th {USENIX} Security Symposium ({USENIX} Security 15)*, 2015, pp. 1041–1056

32. N. Tavabi, P. Goyal, M. Almukaynizi, P. Shakarian, K. Lerman, Darkembed: Exploit prediction with neural language models, in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 7849–7854
33. M. Edkrantz, A. Said, Predicting cyber vulnerability exploits with machine learning, in *SCAI*, 2015, pp. 48–57
34. J. Jacobs, S. Romanosky, B. Edwards, M. Roytman, I. Adjerid, Exploit prediction scoring system (epss). Preprint (2019). arXiv:1908.04856
35. O. Suciu, C. Nelson, Z. Lyu, T. Bao, T. Dumitras, Expected exploitability: Predicting the development of functional vulnerability exploits. Preprint (2021). arXiv:2102.07869