Common Vulnerabilities and Exposures (CVE)

An Overview of CVE and the Security Databases Ecosystem

Seminararbeit

Ausgewählte Kapitel der IT-Security

Vorgelegt von:
Christopher Skallak

Personenkennzeichen
1710475053

Abgabe am:
08.01.2020

Abstract

This thesis deals with the ecosystem of cyber security databases provided by mitre. This survey focuses on the Common Vulnerabilities and Exposures (CVE) database and the Common Vulnerability Scoring System (CVSS) applied by the National Vulnerability Database (NVD). The CVE developed a standard enumeration system for vulnerabilities, which is an industry standard today. The CVE and its enumeration system were developed to interconnect all proletary security databases and make their entries publicly accessible.

Abkürzungsverzeichnis

ATT&CK	Adversarial Tactics, Techniques & Common Knowledge
CNA	CVE Numbering Authority
CAPEC	Common Attack Pattern Enumeration and Classification
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
NIST	National Institute of Standards and Technology
NVD	National Vulnerability Database

Schlüsselbegriffe

Common Attack Pattern Enumeration and Classification
Common Weakness Enumeration
Common Vulnerabilities and Exposures
CVE Numbering Authority
Common Vulnerability Scoring System
MITRE
NIST

Contents

Chapter 1

Introduction

Security databases categorize information of vulnerabilities, weaknesses or other cybersecurity related Topics. The security databases ecosystem provided by mitre is free and publicly available. The mitre security databases ecosystem helps programmers, testers and cybersecurity experts all over the world. Common vulnerabilities and exposures, which is also a part of the mitre security databases ecosystem, set an industry standard enumeration system for vulnerabilities. Even the National Vulnerability Database (NVD) powered by the U.S. government uses the CVE enumeration system.

Chapter 2 gives an introduction to CVE and the mitre cooperation. Chapter 3 presents the current ecosystem of mitre and the relationship of the CVD and NVD database. Chapter 4 focuses on the vulnerability scoring system used by the NVD database. In chapter 5 I give the conclusion.

Chapter 2

Common Vulnerabilities and Exposures (CVE)

2.1 History and background

Common Vulnerabilities and Exposures (CVE) was developed by the MITRE Employees David E. Mann and Steven M. Christey. The original concept was presented as a paper with the title *"Towards a Common Enumeration of Vulnerabilities at the 2nd Workshop on Research with Security Vulnerability Databases on January 21-22, 1999 at Purdue University in West Lafayette, Indiana, USA"* [cve].

MITRE MITRE is a non-profit-organization, which works in the duty of the US government. MITRE is active in the fields of: "defense and intelligence, aviation, civil systems, homeland security, judiciary, healthcare and cybersecurity" [mit].

History of CVE Back in 1999 most of the Cybersecurity had their own vulnerabilities databases and identification methods. The incompatible databases caused the MITRE Employees to specify a common, open and standardized vulnerability identification system for this kind of databases. The first CVE list was published in September 1999 and contained 321 CVE entries. Nowadays CVE has set the industry standard for vulnerability identifiers and is used as a reference point in conversations about vulnerabilities. [cve].

The process of a CVE After the Discovery of the security vulnerability, it gets passed over to a CVE Numbering Authority (CNA). The CNA checks the Information and assign a CVE ID and Description to it. Afterward the CVE Entry is added to the CVE Database and gets published on the CVE website [cve].

CVE Numbering Authorities As of November 12, 2019, 106 organizations in 19 countries all over the world are authorized to assign CVE IDs. There are two Root CNA's, which regulate the voluntary sub-CNAs [cve].

2.2 CVE Entry

An CAn CVE Entry consists of a CVE ID, a brief description of the vulnerability or exposure and references to publicly available papers [cve].

CVE ID The CVE ID is a unique identifier of the CVE. 2.1 shows the Syntax of the CVE ID. The YYYY part defines the year the CVE was assigned by a CVA or was published. The NNNN portion is a unique identifier that gets set back to zero if the YYYY part gets incremented. The restriction of the NNNN portion to 4 digits was changed on January 1 of 2014 to an unlimited number of digits. [cve].

$$CVE - YYYY - NNNN \tag{2.1}$$

CVE Description The description of the CVE is usually *"written by CVE Numbering Authorities (CNAs), the CVE Team, or individuals requesting a CVE ID"* [cve]. The description should contain details to ease the searching process for a specific exposure. *"Ideally, Descriptions include details such as the name of the affected product and vendor, a summary of affected versions, the vulnerability type, the impact, the access that an attacker requires to exploit the vulnerability, and the important code components or inputs that are involved"* [cve].

States of CVE entries The CVE-Entry can have different states depending of the acceptance by the CNA [cve]:

1. Reserved
   A CVE Entry gets marked as "RESERVED", if it gets hold back for further investigation by the CNA. Detailed information's doesn't get published until investigation are completed.

2. Disrupted
   A CVE Entry is marked as "DISPUTED", if two parties disagree with another about this particular vulnerability and the CNA wants to maintain an impartial standpoint.

3. Reject
   If the CNA doesn't accept the CVE request, it gets marked as "REJECT". The CNA states the reason of the rejection in the description of the CVE Entry.

2.3 Reduced complexity of Database mapping with CVE

The main purpose of introducing a vulnerability database based on a standardized Enumeration was create an interoperability between the numerous databases. Figure 2.1 shows that the CVE Database was designed to act as middle man between the ones of the companies, which reduces the amount of mappings [MC99]. [MC99].

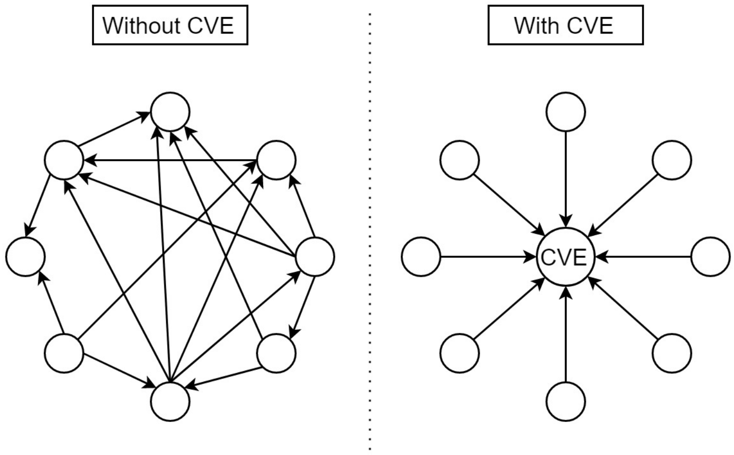

Figure 2.1: Database mapping [cve]

> *" Given two particular databases, the mapping between them must specify which entries in one database are related to entries in the other. For example, if database A enumerates a set of NFS vulnerabilities, and B enumerates its own set, then for each of the vulnerabilities in A, the mapping must specify which of the vulnerabilities in B are related to it. To achieve full interoperability between all databases, we would need to maintain* ${}_nC_2 = n(n-1)/2$ *different mappings.*
>
> *An alternative to this approach is to introduce an additional database - a CVE - and use it as a standard to which all others are mapped. There are two advantages to this approach. The first and most obvious is that this reduces the number of mappings to* $O(n)$ *instead of* $O(n^2)$*. "* [MC99]

Roadblocks to interoperability The CVE was designed to overcome some roadblocks to deliver a high interoperability between databases [MC99]:

1. Inconsistent Naming Conventions
2. Managing Similar Information from Diverse Sources
3. Managing Multiple Evolving Perspectives of the Same Vulnerability
4. Complexity of mapping between databases
5. Political Shareability
   The problem is, that security organizations often tied to commercial interests. Due to this they are copyrighting vulnerability information about their software and hold back the distribution of this information.
6. Potential loss of Precision
   The loss of precision results of the composite database mapping via CVE. A direct mapping would better sustain the precision. The loss of precision is a trade of for the better mapping complexity.

Chapter 3

Security Databases Ecosystem

There is a whole ecosystem of databases with vulnerabilities, weaknesses, event and attack patterns enumeration designed for and by cyber security experts to share the information about exploits and related topics.

3.1 CWE

Common Weakness Enumeration (CWE) is also hosted by mitre since 2008 and holds a list of common weaknesses. The difference to the CVE is that CWE is system and operating system independent. The CWE focuses the principle of the weakness like CWE326: Inadequate Encryption Strength[1]. In contrast CVE enumerates a specific instance of this weakness for example CVE-2019-9506, where the key negotiation protocol gets exploited to lower the key length to one byte[2]. A weakness is an error that leads to a vulnerability. CWE has fixed IDs and is structured in a tree like taxonomy [cwe][3].

CWE Entry Every CWE entry contains an ID, a Name of the weakness and a description of it. There are also Descriptions for the behaviour, exploit and consequences of this weakness. The relationship information about parent and child nodes is also included. The description may also include Codes samples and references to the CVE database via ID's of related exposures [cwe][4].

Structure of the CWE taxonomy The Structure of the CWE taxonomy is a 3-level tree structure. the highest level consists of 5-15 nodes which defines classes of weaknesses like CWE-693: Protection Mechanism Failure which is parent of the mid-tier CWE-311: Missing Encryption of Sensitive Data and others. The middle-layer contains 25 to 60 CWE nodes of descriptive groupings. The lowest layer consists of the CWEs which are related to a specific weakness. In our case is CWE-311 parent of

[1] https://cwe.mitre.org/data/definitions/326.html

[2] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9506

[3] https://cwe.mitre.org/about/process.html

[4] https://cwe.mitre.org/about/faq.html

CWE-312: Cleartext Storage of Sensitive Information and 319: Cleartext Transmission of Sensitive Information [cwe][5]. figure 3.1 shows a little portion of the CWE tree structure.

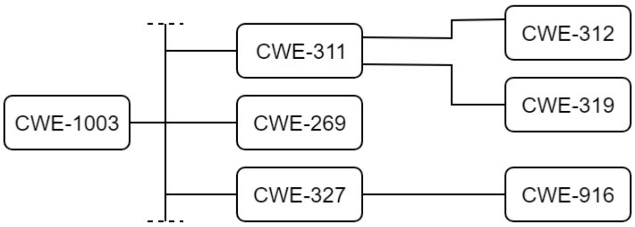

Figure 3.1: Portion of CWE hierarchical structure [nvd]

3.2 Common Attack Pattern Enumeration and Classification (CAPEC)

The Common Attack Pattern Enumeration and Classification (CAPEC) is a database specialized on common attack patterns. CAPEC was released by mitre in 2007 and gets hosted by the organization until today. The purpose of CAPEC is to understand how hackers exploit weaknesses in software. The benefits of CAPEC are cybersecurity training and attacking tools for penetration testing [CAP] [6].

CAPEC Entries CAPEC entries contain [Bar08]:

1. Identifying Information which consists of the attack pattern ID and it's name
2. Description
3. Severity and Likelihood of exploit
4. Relationships of the tree structure
5. Execution Flow
6. Prerequisites
7. Required knowledge, skills and resources
8. Indicators, consequences and mitigations
9. Example Instances
10. Related Weakness and related Vulnerabilities with the CVE / CWE ID's

[5] https://cwe.mitre.org/about/process.html

[6] https://capec.mitre.org/about/index.html

Structure of the CAPEC taxonomy CAPEC also follows a tree-like structure which at different levels of abstraction. At highest layer are the roots CAPEC-3000: Domains of Attack or CAPEC-10000: Mechanisms of Attack which contain the category nodes. The category nodes contain nodes of three abstraction levels[CAP][7][8]:

1. Meta level
   Meta level attack patterns is an abstract description of a technique or pattern used in an attack. It's the high-level approach of a pattern.
2. Standard level
   The Standard level attack patterns are focused on a specific attack processes.
3. Detailed level
   Detailed CAPECS contain the complete execution flow of an attack and are the mist specific ones

Contrast to ATT&CK There is another Database for attack patterns the Adversarial Tactics, Techniques & Common Knowledge (ATT&CK) but they have different use cases. In contrast to CAPEC, which focuses on application security, ATT&CK put their interest into network defense. Many CAPECs are also described by ATT&CK so their entries overlap [CAP][9]

The CVE and NVD Relationship The U.S. National Vulnerability Database (NVD) is also a Vulnerability Database launched by the National Institute of Standards and Technology (NIST). The NVD is synchronized with the CVE Database which means that CVE entries get uploaded to the NVD as soon as they get published. The Team of the NVD add enhanced information to the entries "such as fix information, severity scores, and impact ratings" [cve]. The severity scores also named Common Vulnerability Scoring System (CVSS) are described in chapter 4 [cve].

[7] https://capec.mitre.org/documents/schema/index.html
[8] https://capec.mitre.org/data/definitions/3000.html
[9] https://capec.mitre.org/about/attack_comparison.html

Chapter 4

Common Vulnerability Scoring System (CVSS)

Common Vulnerability Scoring System (CVSS) is an open framework owned and managed by. FIRST.Org.Inc a US-based non-profit organization. CVSS version 3.1 got published in June 2019 and got fully integrated into the NVD database at September 10th,2019. Figure 4.1 shows the arrangement into the three main metric groups: base metric group, temporal metric group and environmental metric group [nvd, fir].

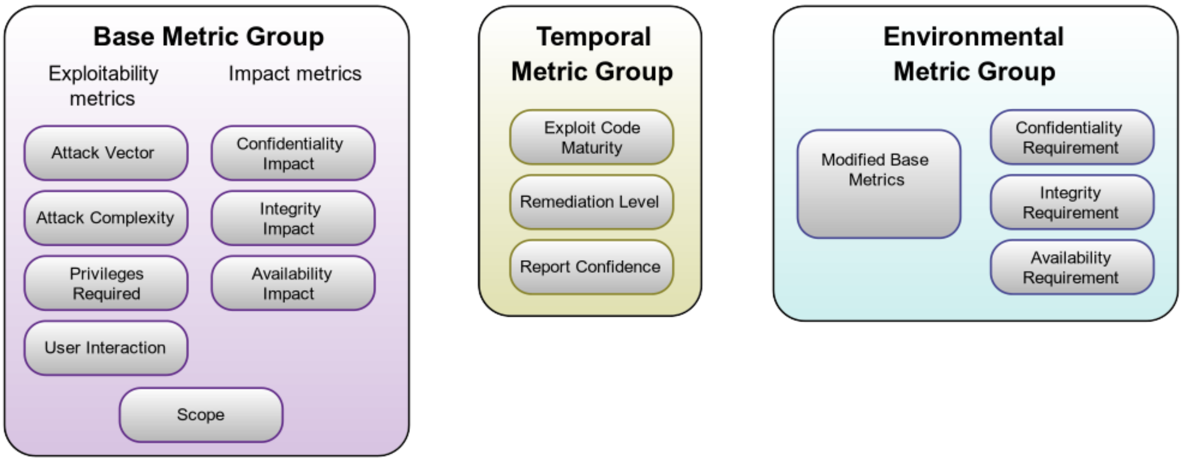

Figure 4.1: CVSS Metric Groups [fir]

4.1 Base Metrics

The CVSS base score is independent of the specific configuration and gets differed into exploitability metrics and impact metrics [fir].

4.1.1 Exploitability Metrics

The base metric group refers to the vulnerable component and scores the following metrics relative to it [fir].

Attack Vector (AV) The Attack Vector describes the remoteness of the attack. The value of this metric is directly proportional to the remoteness of the attack. The Attack vector can be differed into four values [fir]:

1. **Network (N)** The vulnerable component can be accessed through the TCP/IP stack and it is possible to execute the attack anywhere in the internet like a denial of service (DOS) attack via a specially crafted TCP packet.
2. **Adjacent (A)** The vulnerable component can be accessed through the TCP/IP stack but to the attack must be executed in the local area network like a DoS attack via ARP requests.
3. **Local (L)** The vulnerable component has to be attacked remotely by SSH or by using engineering techniques to trick a user into opening a malicious file, because the component is not bound to the network stack.
4. **Physical (P)** This attack vector depends on physical manipulation of the component.

Attack Complexity (AC) The attack complexity metric describes the conditions to exploit the vulnerability and are not in the attacker's control. This score is indirect proportional to the complexity of the attack. The value of this metric can be grouped into high or low [fir]:

1. **Low (L)** There are not any specialized access conditions. The attacker can expect repeatable success of this attack.
2. **High (H)** The success of this attack depends on conditions that are beyond the attacker's control. The attacker must gather knowledge about the environment to be successful.

Privileges Required (PR) This metric describes the level of privileges that is required to perform the attack. This metric gets divided into none, low and high required privileges. CWEs without needed privileges are rated highest score of this metric[fir]:

1. **None (N)** There is no privilege required to perform this attack.
2. **Low (L)** The attacker requires basic user capabilities to perform this attack. The attack can only perform by files owned by the attacker.
3. **High (H)** For this metric value are administrative privileges are needed.

User Interaction (UI) This metric describes the interaction needed of users other than the attacker. If none user interaction is needed the score is the highest. This metric gets differed into two values [fir]:

1. **None (N)** This attack doesn't require no user interaction.
2. **Required (R)** There is user interaction required to perform this attack.

4.1.2 Scope (s)

The scope metric describes if the vulnerability impacts resources of other components. This metrics can be grouped into unchanged (U) and changed (C), where change gathers a higher score.

4.1.3 Impact Metrics

The impact metrics define the effect of an exploited vulnerability on the component. The three impact metrics are confidentiality, integrity and availability.

Confidentiality (C) This metric describes the impact to the confidentiality of the information resources. The value is directly proportional to the metric and can be differentiated into three values [fir]:

1. **High (H)** The vulnerability results in a total loss of confidentiality for all resources of the component.
2. **Low (L)** The vulnerability results in some loss of confidentiality and access to some information is obtained.
3. **None (N)** There is no loss of confidentiality.

Integrity (I) This metric describes the impact to the integrity of the information resources. The value is directly proportional to the metric and can be differentiated into three values [fir]:

1. **High (H)** The vulnerability results in a total loss of integrity or a complete loss of protection.
2. **Low (L)** The vulnerability results in some loss of integrity.
3. **None (N)** There is no loss of integrity.

Availability (A) This metric measures the impact to the availability of the impacted component such as a networked service. The value is directly proportional to the metric and can be differentiated into three values [fir]:

1. **High (H)** The vulnerability results in a total loss of availability. The vulnerability deny any access of legitimate users.
2. **Low (L)** The vulnerability results in some loss of availability. Legitimate users can still access the component or networked service but there is a significant loss in performance.
3. **None (N)** There is no loss of availability.

4.2 Temporal Metrics

> *"The Temporal metrics measure the current state of exploit techniques or code availability, the existence of any patches or workarounds, or the confidence in the description of a vulnerability."* [fir]

Exploit Code Maturity (E) This metric measures the exploit code availability. The value of the metric is directly proportional to the values it gets differentiated in [fir]:

1. **High (H)** There exists autonomous code or tools to exploit this vulnerability.
2. **Functional (F)** There exists code that works in most situations.
3. **Proof-of-Concept (P)** There is Proof-of-concept exploit code is available, or an attack demonstration that only may work on some systems.
4. **Unproven (U)** The exploit is only theoretical, or no exploit code exists
5. **Not Defined (X)** There is insufficient information to choose one of the other values. This value doesn't impact the overall temporal score.

Remediation Level (RL) The remediation level describes if there is a workaround or official fix. An official fix gives this metric the lowest value. The metric is classified into [fir]:

1. **Unavailable (U)** There is no solution available for this vulnerability.
2. **Workaround (W)** There is an unofficial solution available, which got programmed by users.
3. **Temporary Fix (T)** There is a temporary fix or workaround, which gets distributed by the vendor.
4. **Official Fix (O)** There is an official fix which gets distributed by the vendor.

5. **Not Defined (X)** There is insufficient information to choose one of the other values. This value doesn't impact the overall temporal score.

Report Confidence (RC)

> *"This metric measures the degree of confidence in the existence of the vulnerability and the credibility of the known technical details. Sometimes only the existence of vulnerabilities is publicized, but without specific details."* [fir]

This metric gets differentiated in [fir]:

1. **Confirmed (C)** There are existing reports of this vulnerability or a functional reproduction is possible.
2. **Reasonable (R)** Some details are already published but the root cause may still be unknown.
3. **Unknown (U)** There are no details are already published but reports indicate the existence of the vulnerability.
4. **Not Defined (X)** There is insufficient information to choose one of the other values. This value doesn't impact the overall temporal score.

4.2.1 Environmental Metrics

Environmental Metrics are used to adapt the CVSS to a specific configuration in terms of complementary/alternative security controls. The security requirements metrics of confidentiality requirement (CR), integrity requirement (IR) and availability requirement (AR) can be differentiated into the following values [fir]:

1. **High (H)** Loss of confidentiality, integrity or availability is having a catastrophic effect.
2. **Medium (M)** Loss of confidentiality, integrity or availability is having a serious effect.
3. **Low (L)** Loss of confidentiality, integrity or availability is having a limited effect.
4. **Not Defined (X)** There is insufficient information to choose one of the other values. This value doesn't impact the overall temporal score.

Modified Base Metrics

> *These metrics enable the analyst to override individual Base metrics based on specific characteristics of a user's environment. Characteristics that affect Exploitability, Scope, or Impact can be reflected via an appropriately modified Environmental Score.* [fir]

4.3 Vulnerability Severity Ratings

After calculating all this metrics, the calculated value defines the severity of the vulnerability. Table 4.1 shows the severity rating of CVSS version 3.1 [fir].

Rating 1	CVSS Score
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Table 4.1: Vulnerability Severity Ratings [fir]

Chapter 5

Conclusion

CVE, the security databases of mitre ecosystem and other databases which are open to the public like NVD are generating a huge free accessible distributed knowledge about current vulnerabilities and weaknesses and categorize them for improved searching. Databases like this help developers, IT designers, testers, and cybersecurity specialists to gain high quality resources to secure their work and secure the data of us individuals. Furthermore, CVE offers a database to interconnect them all with a minimal mapping complexity. A special Protocol allows databases to access the CVE entries and get a trigger message as soon as a new entry gets published in real time.

For future work it is interesting to analyze the Security Content Automation Protocol (SCAP) which is an a management protocol for Security Content and builds the relationship of CVD and NVD. Furthermore I am interested in analyzing the statistics of the number of entries per year, the distribution of the vulnerability / weaknesses per vulnerability / weakness category or the distribution of CVSS scores per category.

Appendix A

Verzeichnisse

List of Figures

List of Tables

Bibliography

[Bar08] Sean Barnum. Common attack pattern enumeration and classification (capec) schema description. *Cigital Inc, http://capec. mitre. org/documents/documentation/CAPEC_Schema_Descr iption_v1*, 3, 2008. 6

[CAP] Common attack pattern enumeration and classification (capec). [Online] Available: https://capec.mitre.org/ [Accessed: Jan.4,2020]. 6, 7

[cve] Common vulnerabilities and exposures (cve). [Online] Available: https://cve.mitre.org/ [Accessed: Jan.4,2020]. 2, 3, 4, 7, 16

[cwe] Common weaknesses enumeration (cwe). [Online] Available: https://cwe.mitre.org/ [Accessed: Jan.4,2020]. 5, 6

[fir] Cvss v3.1 specification document. [Online] Available: https://www.first.org/cvss/v3.1/specification-document [Accessed: Jan.5,2020]. 8, 9, 10, 11, 12, 13, 16, 17

[MC99] David Mann and Steven Christey. Towards a common enumeration of vulnerabilities. 01 1999. 3, 4

[mit] The mitre corporation. [Online] Available: https://www.mitre.org/ [Accessed: Jan.3,2020]. 2

[nvd] National vulnerability database nvd. [Online] Available: https://nvd.nist.gov/general [Accessed: Jan.5,2020]. 6, 8, 16