University of Amsterdam

Research Project I

Designing structured metadata for CVE reports

February 7, 2021

Student:
Bart van Dongen
13438441

Supervisor:
Armijn Hemel

Course:
RP1

Course code:
53841REP6Y

1 Abstract

Common Vulnerabilities and Exposures (CVE) is an industry standard and used in numerous cybersecurity products en services. The lack of structure results in difficulty retrieving information without spending a lot of time. The research project attempt to determine what metadata or structure could be added to the CVE to get data more easily. Reviewing the information available within the CVE and interviewing users of CVE a new structure is created. CPE, CWE, Fixed version and CVE version are proposed to be added to the standard structure because these data apply on all products. The optional data: SDK, Manufacturer information and files & lines can be used for specific products.

2 Introduction

Common Vulnerabilities and Exposures (CVE) [3]is the standard for information security vulnerability names which was launched in 1999. CVE an industry standard and is used in numerous cybersecurity products and services. CVE's are currently in free text format with the lack of structure. The lack of structure results in difficulty retrieving information without spending a lot of time. Nowadays many products are using the same underlaying libraries and/or are created in the same factories. There are a limited number of suppliers within the same branch. This may result in having the same vulnerability or exposure in multiple products. However, it does not mean that all products are reflected in the CVE. The current CVE structure does not allow for any link between multiple CVE numbers for the same vulnerability or exposure. The research project attempt to determine what metadata or structure could be added to the CVE to get data more easily.

The main research question is: "What metadata or structure could be added to the CVE to get information more easily?"

To answer the main research question sub-questions are defined:

- What is the current status of CVE sources, databases and structure?
- Which data is not properly accessed in the CVE?

- What tools are available for converting CVE reports for easier processing by machines?
- What additional data should the CVE contain?

3 Related work

Peter Mell and Tim Grance [8] wrote three guidelines for federal organizations on the use of CVE vulnerability naming scheme. The guidelines advise on the use of security-related IT products and services that are compatible with CVE naming scheme to periodically monitor their systems for applicable vulnerabilities listed in the CVE vulnerability page 1 of 3 Proposal Research Project Proposal naming scheme. This should be used in descriptions and communications of vulnerabilities.

David A. Waltermire and Karen Scarfone [17] updated the "Guide to using vulnerability naming schemes". The guidelines addresses the different naming schemes CVE and Common Configuration Emumeration (CCE). The recommendations for the end users of organizations that use CVE and CCE are about product and services selection and design. It contains communications and reporting of vulnerability using the CVE, CCE and Common Platform Enumeration (CPE). For software developers and service providers, the guidelines gives recommendations on how to take advantage of the CVE and CCE capabilities and how to use the CPE.

H.S. Venter, J.H.P. Eloff and Y.L. Li. [15] used artificial intelligence to generate a standard set of vulnerability categories. This paper shows the importance of having vulnerability category and a way of achieving this by generating a standard set of vulnerability categories. This is shown with a prototype, that makes it possible to automatically categorize CVE based on their description.

4 Methods

There are two methods used in the study. The first method is a literature study. This is used to gather information about the CVE data structure, sources and databases. The second method is a semi-structured interviews. The semi-structured interviews provide more dialog and the possibility to get into detail. The interviews are one-on-one and based on pre-structured questions. The questions are related to their work field, experience, goals and issues with CVE's. Four people, working with CVE's from a different client perspective, have participated. The different data are analyzed and structured to make more data accessible.

5 CVE status

The CVE status contains a description of the standard, metadata and sources. Common Vulnerabilities and Exposures (CVE) is the standard for information security vulnerability names. The CVE list is copyrighted by MITRE. This is done for the benefit of the community, to ensure it remains a free and opensource standard. However, it is sponsored by U.S. Department of Homeland Security (DHS) Cybersecurity and Infrastructure Security Agency (CISA) [6].

CVE is an industry standard. Numerous organizations include CVE records in their products, services, vendor alerts and security advisories. To assign CVE ID's there are CVE numbering Authorities (CNA) [4]. These are organizations from around the world that are authorized to assign CVE ID to vulnerabilities affecting products. Each CNA has their own scope for assign CVE ID's. The main source for the CVE is Mitre. Table 1 shows the CVE structure of Mitre. Below table 1 is an example CVE.

CVE-ID	
Description	References
Assigning CNA	Data record Created

Table 1: CVE structure

```
CVE-ID: CVE-2020-29016

Description: 'A stack-based buffer overflow vulnerability in FortiWeb
6.3.0 through 6.3.5 and version before 6.2.4 may allow an unauthenticated,
remote attacker to overwrite the content of the stack and potentially
execute arbitrary code by sending a crafted request with a large certname.'

References:
    CONFIRM:https://www.fortiguard.com/psirt/FG-IR-20-125
    URL:https://www.fortiguard.com/psirt/FG-IR-20-125

Assigning CNA: Fortinet, Inc.

Date Record Created: 2020/11/24
```

There are also other databases containing CVE information. These databases add extra information: scores, categorization, affected products and more.

5.1 CVE metadata

Common Weakness Enumeration (CWE) is a common language for communicating software security vulnerabilities where the CWE represents a single vulnerability type. This helps developers and security to describe and discuss weaknesses in a common language.

Common Vulnerability Scoring System Version (CVSS) [5] is a framework for scoring vulnerability using three metric groups: Base, Temporal and Environmental. The score ranges from 0 till 10 (table 2). The CVSS framework is owned by first.org a U.S. based non-profit organization.

CVSS v2.0 Ratings		**CVSS v3.0 Ratings**	
Severity	**Base Score Range**	**Severity**	**Base Score Range**
		None	0.0
Low	0.0-3.9	Low	0.1-3.9
Medium	4.0-6.9	Medium	4.0-6.9
High	7.0-10.0	High	7.0-8.9
		Critical	9.0-10.0

Table 2: CVSS Metrics.
Source: https://nvd.nist.gov/vuln-metrics/cvss

Common Platform Enumeration (CPE) [1] is a method of describing and identifying classes of applications, operating systems, and hardware devices (figure 1). CPE is used in CVE to link product to a CVE or a list of products. This can be useful when trying to find a product or version is affected by a CVE. The most important layer within the CPE structure is the naming layer.

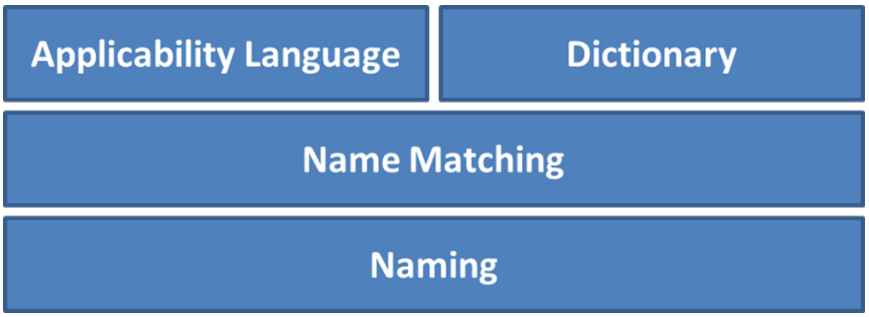

Figure 1: CPE 2.3 stack.
Source: https://csrc.nist.gov/projects/security-content-automation-protocol/specifications/cpe

6 CVE databases

There are five databases giving out CVE information. NVD nist this is a U.S. government repository of standards-based vulnerability management data. This includes databases of security checklist references, security-related software flaws, misconfigurations, product names, and impact metrics [10]. Other countries are starting to host their own CVE databases. Table 3 shows the CVE structure of NVD nist.

CVE-ID	
Description	References
Source	Data record Created
NVD Published Date	NVD Last Modified
CVSS v2/v3	CWE
CPE	

Table 3: NVD structure

The NVD nist added multiple metadata information CWE, CVSS and CPE. NVD NIST analysts are using CWE with their scoring. CWE is also used to categories CVEs. [11]

VULDB is a commercial CVE source [14]. They added extra information around the CVE for their customers to have extra information. The data structure VULDB is shown in table 4.

CVE-ID	
Description	References
Sources	Timeline
Cyber Threat Intelligence (CTI) Interest Score	Threat Intelligence
Exploiting	CWE
CVSS	CPE
Countermeasures	

Table 4: VULDB structure

Whitesources vulnerability database is a database for open-source security vulnerabilities [12] (table 5).

CVE-ID	
Description	Related resources
Date	Top fix
CVSS	CWE

Table 5: Whitesources vulnerability database Structure

The vulnerable code database (Vulncode-DB) is an open source vulnerabilities database that contains the corresponding source code for the vulnerability. The Vulncode-DB is a new vulnerability database that is still in the development stages [16] (table 6).

CVE-ID	
Description	Related resources
Date	Products
Type	First patch
Relevant files	Links
Type	First patch
Detailed repository view	

Table 6: The vulnerable code database structure

Response center Luxembourg CIRL [9] host their own CVE databases using an opensource software cve-search [13]. This software populates a database with CVE information from mitre and nist (table 7).

CVE-ID	
Description	References
Source	Published
Last Modified	Last major update
CVSS	CWE
CPE	

Table 7: CIRL structure

6.1 CVE description

The default CVE information is from Mitre, which only has information about the Vulnerability within the description or linked in a reference to a different source. There is a lot of information within the CVE description, that isn't properly accessible for automation. Because there is no strict format used within the text format of the description. It is difficult to do automation using programs to get information out of the CVE, because of the free text format. When analysing the information of a CVE description it lacks structure, because some descriptions are more detailed than others.
The descriptions usually contain the same characteristics. The standard characteristics are:

- Product;
- Type of vulnerability;

- What happens (sometimes part of the vulnerability type).

Extra data that is not in every description:

- Privileges level;
- Authentication level needed;
- Why it is possible;
- Result of the vulnerability;
- Affected version(s);
- Fix (which patch it is fixed).

For example the CVE-2020-0002 description:

```
"In ih264d_init_decoder of ih264d_api.c, there is a possible out of bounds
write due to a use after free. This could lead to remote code execution
with no additional execution privileges needed. User interaction is needed
for exploitation Product: Android Versions: Android-8.0, Android-8.1,
Android-9, and Android-10 Android ID: A-142602711"
```

Within the description of CVE-2020-0002 there is a lot of information available. Here is a summary of the information:

- Product: Android;
- Type of vulnerability: remote code execution;
- What happens: remote attacker to overwrite the content of the stack and potentially execute arbitrary code;
- Authentication: with no additional execution privileges needed;
- Affected versions: Android-8.0, Android-8.1, Android-9 and Android-10;
- How: There is a possible out of bounds write due to a use after free;
- File name: `ih264d_api.c`;
- Function name: `ih264d_init_decoder`;
- Android identifier: A-142602711 - Android identifier is used within Android's own bug tracker and usually mentioned in Git commit messages.

7 Convert CVE reports

The description of CVE reports are made in free text. This means that the description does not have a structure, therefor it is harder for machines to automatically obtain data of the CVE reports. It is difficult to convert CVE reports. Due to the difficulty of converting CVE reports, there are not many tools available. A few papers describe how they applied machine learning to get information out of the description of the CVE report. So is it possible with machine learning to do characterization of software vulnerabilities, categorizing CVE exploits and CVSS score prediction. This can help companies to get information faster out of the CVE reports without spending too much time on each CVE report.

H.S. Venter, J.H.P. Eloff and Y.L. Li. describe why it is important to have standardized vulnerability categories [15]. It gives the benefit of doing comparison, interoperability, abstract reports and simplifying the responsibilities of vulnerability databases. There is a prototype vulnerability analyzer (VA) made. This is used to identified vulnerability categories based on CVE information.

Research of Atefeh Khazaei, Mohammad Ghasemzadeh and Vali Derhami [7] shows how to use language description of vulnerabilitites for the CVSS calculation. They use text mining tools and techniques to extract information from the text format and examined different algorithms to predict the CVSS scores.

8 CVE data

Data is gathered that should be added to the CVE structure by four IT fields: Software auditing, security research, project management and development. These fields uses the CVE-data for different purposes.

8.1 Software auditing

Software auditing misses a data structure that contains the location where the vulnerabilities are within the code. The data contains information where to find vulnerable snippets. The snippets can be used to compare against other source codes, to see if the source code contains vulnerabilities. Whenever there is new CVE, it is easier and faster possible to add snippets of code that are vulnerable.

When using close source this is not possible. Then it would be useful to have information in which patch the vulnerability is fixed. Some CVE databases do already provide information about the patch. Sometimes this can be found in the references of the CVE. Because it is hard to find automatically patches in references, it is easier to have a field within the CVE structure. This would make it possible for companies to automate this process. Whenever the vulnerability is found they can do checks on daily bases to see when there is a patch rolled out including the fix for the vulnerability.

For the software auditing CPE's are used to link CVE's to products or modules to find vulnerabilities. The CPE standard isn't perfect. It depends on the person registering the CPE. It can happen that multiple CPE names are used for the same product, because a vendor or product name is written differently. Within the CPE structure the version of the product is not always set. A manually check is required to find out if a product is affected by CVE. This can cost a lot of time.

The CPE structure is also missing data for modules or plugins. This can make it very hard to get a general overview to see how many plugins or modules are vulnerable within a product.

8.2 Security research

The most import information for security research is knowing that there was an issue (vulnerability) in a product and that this information is public. This information can be used to inform affected clients. It is important to know which versions of the product are affected by the CVE.

Missing is the data when the vulnerability is fixed. This information can be used to exactly know when a issue is still present within the product.

8.3 Development

The data of the CVE en CPE that is missing for development with CVE and CPE is the Release URL. This URL can indicate the locations of the source code release, the release

page or even the release binary. This data can be helpful when trying to figure out if the product is vulnerable. The data that could be added is a project identifier, for example an ID number, an combination with a version number. The current CPE is not always correctly filled in. Especially when the CVE affects multiple product versions and when it is between, above or below specific version. Sometimes projects change the version scheme while using it. Then it is difficult to determine which versions of a project are Affected. Within the CPE their might be too much structure that makes it difficult to use.

8.4 Project management

For project management it is important to link multiple CVE's together. This can be done by having extra information about:

- Vendor information;
- Software Development Kit (SDK);
- Manufacturer;
- Plugins;
- Type of vulnerability.

With this information it would be easier to identify vulnerable products that are not included in the CVE product list. For example multiple products could be using the same SDK. When a researcher finds a vulnerability in a SDK, he only links the product that was researched. Products using the same SDK could contain the same vulnerability.

metadata is information of the person that reports the CVE. It could be useful to submit this to the CVE data. So it is possible get in touch or see in which countries vulnerabilities are reported.

Product identifiers are useful so that it is possible to get a notification when there is a new CVE released. Using some kind of version management makes it easier to update data and look back to the different versions.

9 New CVE structure

Based on the interviews and study a new CVE structure is proposed. To ensure backwards compatibility with older CVEs, the new structure is designed as an extension of CVE rather than a completely new format. This new format will be referred to as CVE+. The most important piece of information that is missing in the CVE is the affected product data. Currently CPE is used for linking CVE with products. This CPE information is stored in external databases requiring extra steps. This is why CPE should be an integral part of CVE+. The product information can already be found in the CVE description. CWE data is already included in the CVE structure of third party databases. CWE shows the type of exploit and can be used to categorize CVE's. CWE should therefore be part of CVE+.

A field "Fixed version" should be added to the CVE+. Initially this field will be empty. When a fix is released this field can be updated to contain the fixed version. Certain fields in CVE+ are static, other fields could be updated or changed in the future. For example the assigned CVE number will always be the same, but others like 'Fixed version' will be changed. This is why CVE+ should be versioned, so programs processing CVE+ reports can more easily find out that a CVE+ report was changed. By including the full history with all the changes it becomes easier to find out what has been changed. Currently CVEs only contain a publication date and a modification date, but not what was changed, when and why it was changed.

The CVE+ should contain optional data space for specific product types. It is optional because this doesn't affect every product. SDK data in the CVE can be useful when there is a vulnerability in the SDK. Researchers can use the SDK data to see if other products are also effect by the same CVE. Within the CPE there is no information about the Manufacturer. Vendors might use multiple Manufacturers. The same product from different Manufacturers does not necessarily contain the same vulnerability because of different hardware usage. By adding the Manufacturer data it makes it possible to see differences. The Manufacturer data can be added in multiple ways: in own field or integrated in the CPE structure. Since not all products have a Manufacturer, the use of a separate data field is recommended. Open source projects might be re-used by other parties. They can create the same vulnerability in their product. The snipped of the code can be used to check if there project has the same vulnerability.

To create a link between CVEs, the related CVEs field is introduced in the CVE+. Related CVEs is usefull when there are multiple CVEs that are almost the same. Like the following CVEs: CVE-2006-2559, CVE-2006-2560, CVE-2006-2561, CVE-2006-2562, CVE-2011-4501 and CVE-2011-4502. Those CVEs are all taking advantage of the same vulnerability with UPnP request in different products. Most CVE databases also added a CVSS score. The reliability of the score depends on the experience of a person. Therefore the interpretation of the score is difficult and will not be include in the CVE+.
The CVE+ contains default data fields and also optional data field (Table 8).

CVE-ID	
Description	References
Assigning CNA	Data record Created
CWE	CPE (AFFECTED version)
Fixed version	CVE version
*SDK	*Manufacturer
*File : lines	*Related CVEs

Table 8: CVE+ structure
'*' = optional data
Gray = new data

An example of the CVE+ is given below.

```
CVE-ID: CVE-2020-0003

Description:
    'In onCreate of InstallStart.java, there is a possible package
    validationbypass due to a time-of-check time-of-use vulnerability.
    This could lead to local escalation of privilege with no additional
    execution privileges needed. User interaction is needed for
    exploitation. Product: Android Versions: Android-8.0 Android
    ID: A-140195904'

References:
    CONFIRM:https://source.android.com/security/bulletin/2020-01-01

Assigning CNA:
```

```
    Android (associated with Google Inc. or Open Handset Alliance)

Date Record Created: 20191017

CWE:
    CWE-367
CPE:
    cpe:2.3:o:google:android:8.0:*:*:*:*:*:*:*

Fixed version:
    'android security patch level January 1, 2020'


File:
    'android/platform/packages/apps/PackageInstaller/a422e8c84983
    c91f804881d36f31a88851400927/./src/com/android/
    packageinstaller/InstallStart.java'
    Lines: 55-137
```

10 Conclusion

The goal of the paper was to look into the data structure of the CVE and to answer the main research question: "What metadata or structure could be added to the CVE to get information more easily"?

From research we concluded that not all information within the CVE description is properly accessible for automation because of the lack of structure. CPE is used to link product to a CVE of list of products. The most important layer within de CPE structure is the naming layer. And there are four databases for giving out CVE information. It is difficult to convert CVE reports and there aren't many tools available. Based on research there is a CVE+ structure created. This CVE+ contains CPE, CWE, Fixed version and CVE version. These data applies on all products. The new structure also includes optional data. SDK, manufacturer information, files and line (reference to affected code) and related CVEs, can be used for specific products.

11 Future work

To get the CVE+ structure to be used, the CNA need to change the form and tools they used to add a CVE to the database. So for future work there is a need to create a tool that can be applied to use the new CVE data structure. If it's not possible to integrate the CVE+ then the data must be converted in such a way that it is structured within the CVE description. So it will be possible to already get more information out of the CVE description. To do more automation there can be looked into adding more optional data in a standard way for specific use cases. It creates the possibility for the community to easily add extra information for specific type of products.

References

[1] Brant Cheikes, David Waltermire, and Karen Scarfone. "NIST Interagency Report 7695, Common Platform Enumeration: Naming Specification Version 2.3". In: (Aug. 2011).

[2] Q. Chen et al. "Categorizing and Predicting Invalid Vulnerabilities on Common Vulnerabilities and Exposures". In: *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*. 2018, pp. 345–354. DOI: 10.1109/APSEC.2018.00049.

[3] *Common Vulnerabilities and Exposures (CVE)*. URL: https://cve.mitre.org/.

[4] *CVE Numbering Authorities*. URL: https://cve.mitre.org/cve/cna.html.

[5] *CVSS v3.1 Specification Document*. URL: https://www.first.org/cvss/v3.1/specification-document.

[6] *Frequently Asked Questions*. URL: https://cve.mitre.org/about/faqs.html.

[7] Atefeh Khazaei, Mohammad Ghasemzadeh, and Vali Derhami. "An automatic method for CVSS score prediction using vulnerabilities description". In: *Journal of Intelligent & Fuzzy Systems* 30 (Aug. 2015), pp. 89–96. DOI: 10.3233/IFS-151733.

[8] Peter Mell and Tim Grance. "Use of the Common Vulnerabilities and Exposures (CVE) Vulnerability Naming Scheme". In: (Sept. 2002), p. 6.

[9] *Most recent entries - CVE-Search*. URL: https://cve.circl.lu/.

[10] *National Vulnerability Database*. URL: https://nvd.nist.gov/.

[11] *NVD CWE Slice*. URL: https://nvd.nist.gov/vuln/categories.

[12] *Open Source Vulnerability Database*. Jan. 2021. URL: https://www.whitesourcesoftware.com/vulnerability-database/.

[13] *search main page: cve-search - tool-set to perform local searches for known vulnerabilities*. URL: https://www.cve-search.org/.

[14] *the community-driven vulnerability database*. URL: https://vuldb.com/.

[15] H.S. Venter, J.H.P. Eloff, and Y.L. Li. "Standardising vulnerability categories". In: *Computers and Security* 27.3 (2008), pp. 71–83. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2008.04.002. URL: http://www.sciencedirect.com/science/article/pii/S0167404808000096.

[16] *Vulncode-DB*. URL: https://www.vulncode-db.com/about.

[17] David Waltermire and Karen Scarfone. "NIST Special Publication 800-51 Revision 1, Guide to Using Vulnerability Naming Schemes". In: (Feb. 2011).

12 Appendix

12.1 Interviews Questions

Questions:

1. What field do you work?
2. For what do you use CVE?
3. How often do you use CVE?
4. Which source do you use for your CVE information?
   (a) Do you use any tools that use CVE?
   (b) Have you investigated different CVE source?
5. What data form of CVE do you use?
6. What for information are you missing within the CVE?
   (a) What technical information could be useful to add?

(b) What data is available in the Cve but not easy accessible (in the Description)?

(c) To add the missing data. Where would you like to see it? Something created like CPE and CWE as meta data. Or directly to the CVE as a new entry?

(d) What kind of scheme or naming structure would you want for the new added data?

7. Have you ever encountered a product with a vulnerability that was already in the CVE database? But the product or the CVE did not show any information of being vulnerable