CVE (NVD) Ontology

Vladimir Dimitrov [1]

[1] *University of Sofia, 5 James Bourchier Blvd., Sofia, 1164, Bulgaria*

Abstract

CVE (Common Vulnerabilities and Exposures) is community-supported database of known vulnerabilities. It is under permanent update and development. The database is hosted by MITRE Corporation.

NVD (National Vulnerability Database) is an extract of CVE developed by NIST. It contains only "stable" vulnerabilities and exposures. After careful investigations, NIST augments this database with useful for exploitation information.

CVE (NVD) ontology is based on NVD database. It redefines vulnerabilities in terms of the Semantic Web in Manchester syntax.

This is the next ontology after CPE ontology in the ontology series developed to capture cybersecurity knowledge on vulnerabilities, weaknesses and attacks.

CVE (NVD) ontology is very useful for further investigations on new vulnerabilities using reasoners. It is useful for cybersecurity incident forensic investigations too.

Keywords

CVE, ontology, vulnerabilities, Semantic Web

1. Introduction

IoT has a complex multilayered architecture. These layers are different by their nature starting from classical built-in devices and ending to the Business Intelligence solutions.

IoT security has to be investigated with different approaches and knowledge on these architecture layers and combined in one solution.

In this situation (i.e. IoT), the most flexible approach for security knowledge presentation are the ontologies. This is the main motivation for research presented in this paper.

Information Systems & Grid Technologies: Fifteenth International Conference ISGT'2022, May 27–28, 2022, Sofia, Bulgaria

EMAIL: cht@fmi.uni-sofia.bg (V. Dimitrov)
ORCID: 0000-0002-7441-253X (V. Dimitrov)

2. CVE repositories

The MITRE Corporation's Common Vulnerabilities and Exposures (CVE) [1] referred as "vulnerability list" below, and the NIST's National Vulnerability Database (NVD) [2] – "vulnerability database". The difference between these repositories is explained in the next section.

The initiative for the list of vulnerabilities is to "identify, define and catalog publicly disclosed vulnerabilities in cybersecurity" [1]. It is implemented in partnership with business, academia and government institutions. Participation in the CVE program is voluntary and free. Any organization that wishes and has an interest in participating in the initiative can register itself as a CNA (CVE Numbering Authority). The task of the CNA is to assign identifiers (CVE IDs) of emerging (newly discovered) vulnerabilities in products that fall within CNA scope. As of May 2021, there are 167 registered organizations from 27 countries as CNAs. Among them are vendors, projects, vulnerability researchers, national and industrial CERTs (Computer Emergency Response Teams), bug detection programs in the software and the hardware, and CNA organizational structures.

The CNA can reserve a CVE ID (Allocated) to which an Assigned vulnerability can later be added. The identifier with the associated vulnerability is then subjected to public discussion (Public). An identifier can be rejected.

The life cycle of identifiers and vulnerabilities is reflected in the repository records as states.

Initially, an empty CVE ID is created by the CNA.

This is followed by filling the record with information about the associated vulnerability (Populated). The state of filling with information is considered as a constant state, i.e. it is assumed that something can always be added to the vulnerability description.

A vulnerability can be removed along with its ID, but the record continues to appear as Rejected in the repository.

There is another situation with the records where the identifier is public and the associated record is reserved (Reserved but Public – RBP). This happens when a CNA has published its recommendation for the vulnerability, but has not yet completed the record.

More details on the states of the identifiers, the repository entries and the operation of the CNA can be found in [1].

The list of vulnerabilities maintained by MITRE Corporation aims to register and reconcile community identifiers and vulnerabilities by CNA organizations. The process is described in more details in [1].

The CVE program of MITRE Corporation is sponsored by the US Department of Homeland Security (DHS) and the Cybersecurity and Infrastructure Security Agency (CISA).

The CVE repository of MITRE Corporation is a list of records, each containing an identification number, a description, and at least one public reference to the known vulnerabilities.

The list of vulnerabilities is used by NIST for the organization and maintenance of NVD. This is a database with quite detailed information about each vulnerability, which has been added by NIST after a subsequent analysis. NVD is regularly updated by NIST using the CVE list. Each new vulnerability is analyzed and enriched with the analysis results. Some of the vulnerabilities in the CVE list may be ruled out by NIST.

NVD and CVE can be used freely by anyone.

The focus of the current study is NVD, as the vulnerability records contain richer information than the list.

3. NVD

NVD is a product of the NIST Computer Security Division, Information Technology Laboratory and is sponsored by CISA.

The framework in which NVD is supported by NIST is defined by SCAP (Security Content Automation Protocol).

NVD includes references to security checklists, software security vulnerabilities, incorrect configurations, and impact metrics.

The vulnerability records in the NVD partially contain the information from the CVE list – the essential information has been left, the procedural one has been removed and a new one has been added by the NIST analysis.

The vulnerability analysis performed by NIST is based only on publicly available sources. This is the beginning of the analysis of the new vulnerabilities – it is checked whether the referenced materials in the CVE record are publicly available.

The next analytical task is to associate vulnerability with CWEs (Common Weakness Enumeration) [3]. Weaknesses are considered as types of vulnerabilities, i.e. vulnerabilities are categorized by CWE weaknesses.

In [1], vulnerability is defined as weakness in computational logic (e.g. code) found in software or hardware components that, when exploited, can adversely affect privacy, integrity or accessibility. Mitigating vulnerabilities in this context usually involves modifying the code, but may involve changing the code specification and even removing it (for example, removing affected protocols or even functionality in particular and entirely).

The following two definitions of [4] link weakness and vulnerability:

- Vulnerability is "*an occurrence of a weakness (or multiple weaknesses) within a product, in which the weakness can be used by a party to cause the product to modify or access unintended data, interrupt proper execution, or* *perform incorrect actions that were not specifically granted to the party who uses the weakness.*"

- Weakness is "*a type of mistake that, in proper conditions, could contribute to the introduction of vulnerabilities within that product. This term applies to mistakes regardless of whether they occur in implementation, design, or other phases of a product lifecycle.*"

Some weakness may be available in the product, but may not be accessible for exploit – this is due to "appropriate conditions".

The definitions of weakness and vulnerability overlap in the sense that they are present in the product, but unlike weakness, vulnerability is exploitable.

The applied classification practice of MITRE Corporation and NIST deepens the differences:

- CVE vulnerability is always related to a specific product (or products) and version (or versions).
- CWE weakness can be linked to specific products and versions through the CVE vulnerabilities it categorizes. Moreover, CWE weaknesses are most often organized into taxonomies and may not be directly related to products and their versions.

A CWE weakness can categorize a whole group of weaknesses for which the weakness detection and mitigation methods (described in the weakness) are the same. The manifestation of weakness as a vulnerability is the same for the various vulnerabilities associated with it. Weakness as a vulnerability can be manifested in different environments, platforms, configurations, etc., i.e. to be expressed in different ways in its manifestations.

This is not the end of addressing weaknesses, but it is enough for vulnerabilities, i.e. vulnerabilities are categorized by weaknesses.

NIST uses weaknesses organized in CWE-1003 view to categorize vulnerabilities. The categorization of vulnerabilities in NVD is coordinated with the CWE team of MITRE Corporation.

In the next step of the vulnerability analysis, NIST uses the Common Vulnerability Scoring System (CVSS) exploitation and impact metrics, a common vulnerability scoring system: CVSS V 2.0 and CVSS V 3.1 [5].

Furthermore, the vulnerability is characterized by an applicability statement under the CPE [6], i.e. indicates for which products, platform versions, etc. is available.

An independent verification of the analytical results is performed before the new analyzed CVE is published in NVD.

This is a general description of the process and the contents of the NVD vulnerability database.

4. NVD structure

Originally, NVD was distributed in XML format, but due to attacks on the site, it was decided to switch to JSON format. Accordingly, the XSD schema and the JSON data schema are available. The first schema is nvdcve.xsd, and the second – nvd_cve_feed_json_1.1.schema (and CVE_JSON_4.0_min_1.1.schema). There is another schema for the configuration names used – nvd_cpematch_feed_json_1.0.schema

The schemas are available from the NIST website. In the discussions below is used the current format – JSON.

It should be noted that despite the mentioned schemas in NVD, as well as in the other databases (CPE, CWE and CAPEC), control over the distribution is not done, and i.e. it is possible that part of the distributed dictionary material does not correspond to the respective schema.

NVD information is available through the REST service and in the form of files. The latter are in JSON format – one for each year starting from CVE-2002 to the current year. Two more CVE-Modified and CVE-Recent files are updated every two hours.

CVE-Recent includes all new CVEs, and CVE-Modified includes all modified CVEs.

All files reflect the state of the database at a specific date and time.

In order not to constantly publish the entire database, the last two files are updated and published more often. Their content is related to the already published CVEs by years.

As the name implies, CVE-Recent contains the recently added CVEs (new), and CVE-Modified – the recently modified CVEs in the files published over the years. In fact, CVE-Modified also includes newly added vulnerabilities from CVE-Recent.

Another important file contains the link between CPE applicability expressions and base CPE names from the CPE dictionary.

The description of the NVD files uses the JSON format following the nvd_cve_feed_json_1.1.schema.

Each file represents one JSON object. More details for their structure is available in the corresponding schemas.

The unfortunate point with the platform name matches is that the CPE dictionary lacks a number of names set with the “feeds” from the NVD dictionary.

The method of setting the “feeds” for CPE name matches is not described anywhere in any reasonable way.

Every CVE is scored with CVSS V3.0 and CVSS V2.0. Although CVSS V3.0 is very similar to CVSS V2.0, there are differences in the values the listed types and the set of properties.

For more details on the applicability expressions of CPE [6], CVSS V2.0 [7] and CVSS V3.0 [7], see the relevant referenced sources.

The NVD database uses applicability operators for vulnerabilities to the CPE platform names. In fact, this is a form of the applicability language of CPE (matches). Accordingly, an array of "base" CPE names from the CPE platform dictionary should be matched by these matches.

The CPE "feed" file is in JSON format and is controlled by the nvd_cpematch_feed_json_1.0.schema. The file is an array of objects. Every element in this array must represent the "match-array of CPE names" match. The concept of setting matches is to present a configuration of platform names for which the vulnerability is applicable. The vulnerability can be applicable in several configurations.

5. CVE ontology

A central class in the ontology is `CVE`, which describes the vulnerabilities. Its data properties are:

- `publishedDate`, date of publication;
- `lastModifiedDate`, date of last change.
- Object properties of CVE are:
- `basicMetricV2`, functionally, points to an individual describing the vulnerability scoring on CVSS V2;
- `basicMetricV3`, functionally, points to an individual describing the vulnerability scoring on CVSS V3;
- `configuration`, specified in the individual(s) of the `Configuration` class, describing the platforms for which the vulnerability is applicable / inapplicable;
- `cwe`, indicates in the individual(s) of class `CWE` (weakness) by which the vulnerability is categorized;
- `vendor`, indicates in an individual(s) of the `Vendor` class, the platforms are classified by vendors, is obviously a remnant of an older vulnerability categorization and is not currently used, it is left for compatibility.

The description, CNA that assigned the vulnerability (ASSIGNER), references and status (STATE) are presented as annotations with the corresponding multiplicities to each CVE.

The `CVSSV2` class represents the vulnerability scoring on CVSS V2. This class has only data properties that represent the values of the various elements of the CVSS vector (`accessVector`, `accessComplexity`, `authentication`, `confidentialityImpact`, `integrityImpact`, `availabilityImpact`, `baseScore`, `exploitability`, `remediationLevel`, `reportConfidence`, `temporalScore`, `collateralDamagePotential`, `targetDirectionqualityRequirement` and `environmentalScore`).

CVSSV2 also includes NVD vulnerability assessments, namely severity, exploitabilityScore (impactScore, acInsufInfo, obtainAllPrivilege, obtainUserPrivilege, obtainOtherPrivilege and userInteractionRequired).

The version and vectorString fields are not included in the ontology because this is duplication of information. The first field, however, appears in the class annotation, and the scoring vector is deployed in the description fields.

The CVSSV3 class represents the vulnerability scoring on CVSS V3. This class has only data properties that represent the values of the various elements of the CVSS vector (attackVector, attackComplexity, privilegesRequired, userInteraction, scope, confidentialityImpact, integrityImpact, availabilityImpact, baseScore, baseSeverity, exploitCodeMaturity, remediationLevel, reportConfirerality, tempoRefessional, integrityRequirement, availabilityRequirement, modifiedAttackVector, modifiedAttackComplexity, modifiedPrivilegesRequired, modifiedUserInteraction, modifiedScope, modifiedConfidentialityImpact, modifiedIntegrityImpact, modifiedAvailabilityImpact, environmentalScore and environmentalSeverity).

CVSSV3 also includes NVD vulnerability assessments, namely exploitabilityScore and impactScore.

The version and vectorString fields are not included in the ontology because this is duplication of information. The first field, however, appears in the class annotation, and the scoring vector is deployed in the description fields.

The Configuration class represents configurations from platforms that are vulnerable to a given CVE. In one configuration, there are platforms that are vulnerable and platforms that are not vulnerable. The latter, however, are combined with the property vulnerable to manifest the vulnerability.

In this case, an individual in the Configuration class represents only one combination of CPE platforms. In this individual CPE, the names are only on basic platforms in the dictionary. This is represented by the vulnerableCPE and nonVulnerableCPE object properties.

The applicability language expression is calculated to a variety of configurations. Each configuration is a set of CPEs – vulnerable and non-vulnerable. It is possible that the expression gives only invulnerable configurations, i.e. everyone else is vulnerable.

Each configuration (Configuration individual) is the most basic element. There are no logical expressions and combinations here – this is a conjunction. In fact, the applicability expression gives a disjunction of conjunctions, and each conjunction is separated into a configuration. The negation operator sinks into the vulnerable attribute.

6. Conclusion

CVE ontology with NVD source is useful for security investigations of complex solutions and particularly of IoT solutions. This ontology can be further loaded with MITRE Corporation CVEs to extend its usability for research and investigation of newly established vulnerabilities. The information for last ones usually is contradictive, partial and under classification. Reasoners on ontologies are very useful in that situation.

Presented here CVE ontology is relatively simple following the NVD database structure. The problem with this ontology is its size. The ontology is very huge and special structuring has been used ontology to be used by humans. Further research and investigations have to be done in that direction.

7. Acknowledgements

This paper is prepared with the support of MIRACle: Mechatronics, Innovation, Robotics, Automation, Clean technologies – Establishment and development of a Center for Competence in Mechatronics and Clean Technologies – Laboratory Intelligent Urban Environment, funded by the Operational Program Science and Education for smart growth 2014-2020, Project BG 05M2OP001-1.002-0011.

8. References

[1] MITRE Corporation, Common Vulnerabilities and Exposures (CVE), 2022, URL: https://cve.mitre.org.

[2] NIST, Information Technology Laboratory, National Vulnerability Database (NVD), 2022, URL: https://nvd.nist.gov.

[3] NIST, Information Technology Laboratory, National Vulnerability Database (NVD), NVD CWE Slice, 2022. URL: https://nvd.nist.gov/vuln/categories.

[4] MITRE Corporation, Common Weaknesses Enumeration (CWE), Glossary, 2022, URL: https://cwe.mitre.org/documents/glossary.

[5] NIST, Information Technology Laboratory, National Vulnerability Database (NVD), CVSS, 2022. URK: https://nvd.nist.gov/vuln-metrics/cvss.

[6] NIST, NISTIR 7698, Common Platform Enumeration: Applicability Language Specification Version 2.3, 2022: URL: https://csrc.nist.gov/publications/detail/nistir/7698/final.

[7] FIRST, Common Vulnerability Scoring System SIG, 2022. URL: https://www.first.org/cvss.