enisa

EUROPEAN UNION AGENCY
FOR CYBERSECURITY

# STATE OF VULNERABILITIES 2018/2019

## Analysis of Events in the life of Vulnerabilities

DECEMBER 2019

# ABOUT ENISA

The mission of the European Union Agency for Cybersecurity (ENISA) is to achieve a high common level of cybersecurity across the Union, by actively supporting Member States, Union institutions, bodies, offices and agencies in improving cybersecurity. We contribute to policy development and implementation, support capacity building and preparedness, facilitate operational cooperation at Union level, enhance the trustworthiness of ICT products, services and processes by rolling out cybersecurity certification schemes, enable knowledge sharing, research, innovation and awareness building, whilst developing cross-border communities. Our goal is to strengthen trust in the connected economy, boost resilience of the Union's infrastructure and services and keep our society cyber secure. More information about ENISA and its work can be found at www.enisa.europa.eu.

## CONTACT

For contacting the authors please use team@enisa.europa.eu.
For media enquiries about this paper, please use press@enisa.europa.eu.

## PROJECT MANAGER

Cosmin CIOBANU (ENISA)

## AUTHORS

Vasilis Katos, Shahin Rostami, Panagiotis Bellonias, Nigel Davies, Agnieszka Kleszcz & Shamal Faily (Bournemouth University), Arnolnt Spyros, Alexandros Papanikolaou, & Christos Ilioudis (InnoSec), Konstantinos Rantos (International Hellenic University).

## ACKNOWLEDGEMENTS

We wish to thank the following experts for their continuous support and guidance throught the project: Ioannis Psaroudakis (CERT-EU), James Shank & Steve Santorelli (Team-Cymru), Mihai Barbulescu (RoEdu), Michael Mortensen (Risk Based Security), Kristine Hovhannisyan & Andrew Roberts (Tallinn University of Technology), Costantinos Patsakis, (University of Piraeus), Megan Pleva (Limetools), Cagatay Yucel (Yasar University).

## LEGAL NOTICE



## COPYRIGHT NOTICE

ISBN: 978-92-9204-325-4, DOI: 10.2824/139301

# TABLE OF CONTENTS

# EXECUTIVE SUMMARY

The vulnerability ecosystem has matured considerably in the last few years. A significant amount of effort has been invested to systematically capture, curate, taxonomize and communicate the vulnerabilities in terms of severity, impact and complexity of the associated exploit or attack. Standardisation in the description of vulnerabilities contributes not only to effective threat intelligence sharing, but also potentially efficient threat management, provided that organisations, vendors and security researchers actively seek to discover the vulnerabilities and respond in a timely fashion.

As the standardisation of cataloguing and modelling the vulnerabilities reaches the aforementioned maturity, public or private (i.e. commercial) databases containing information of the actual vulnerabilities (and some with their exploits counterparts) have emerged. As there are a number of initiatives within the research community, quite naturally some databases could be considered to be more “authoritative” and/or “reliable” than others. However, due to the nature of the vulnerability ecosystem, it is not a reasonable assumption that the databases will be complete (that is, contain all vulnerabilities), or reliable in the sense that the information captured is correct, in the sense that the samples gathered can be considered to reliably help in drawing conclusions on the whole population. This is influenced by a number of factors, including the quality of analysis and assessment, the assessment framework itself, the economic aspects (such as the value of any available exploit), as well as the business models of the software vendors, threat intelligence services, and the overall security community.

The purpose of this report is to provide an insight on both the opportunities and limitations the vulnerability ecosystem offers. By using the vulnerabilities published during the year of 2018 and Q1-Q2 of 2019 as a vehicle, this report goes beyond the standard exploratory analysis, which is well captured by many industry whitepapers and reports, and attempts to answer questions related to the reliability, accuracy of the vulnerability sources and the widely accepted evaluation metrics.

In addition, the report leverages established vulnerability taxonomies and frameworks to explore and identify more intrinsic relationships and characteristics. Vulnerabilities are explored in terms of the ATT&CK taxonomy[1], revealing non-uniform distribution in the defined tactics:

1. Differences, inconsistencies and discrepancies between the two major versions of the scoring systems (CVSS[2] version 2 and version 3) may influence risk management actions;
2. Vulnerabilities showing affinity to specific industry sectors, form strong clusters; and of course the
3. Position and performance of vendors and products which varies depending on the type of software.

This report is also accompanied by the underlying dataset and software developed (in Jupyter[3]/Python). These are made publicly available to enable further and independent exploration and analysis of the vulnerability domain by the information security community as well as allow researchers to appreciate the degree of intractability surrounding empirical analysis of vulnerabilities.

---

[1] https://attack.mitre.org/
[2] https://www.first.org/cvss/v3.1/specification-document
[3] https://jupyter.org/

The key findings of the analysis are as follows:

- There are significant differences between the two vulnerability measurement systems (CVSS v2 and CVSS v3) regarding the underlying recorded values. This is possibly attributed to the different wording of the categorical variables fuelling subjective bias. In either case, the correlations of the three impact measures (Confidentiality, Integrity and Availability) were surprisingly low, with Integrity and Availability having a correlation coefficient less than 0.4.
- There are inconsistencies and discrepancies between the different sources. Although there is an authoritative database capturing vulnerability details, this does not imply that the information in that database is accurate.
- The developed taxonomies and standards to describe the vulnerabilities are indeed rich and detailed, but only a subset of the categories was present in the 2018-2019 vulnerabilities.
- There are statistically significant differences between the severity level of CVE (officially recorded) and non-CVE vulnerabilities (i.e. those that were not listed or included in the CVE databases), with the latter showing a higher score.
- The exploit publication date of CRITICAL vulnerabilities is attracted near the vulnerability publication date, with the most exploits being published shortly before or after the vulnerability publication date.
- At least 8.65% of the vulnerabilities are exploitable. This number is expected to be higher due to zero-day exploits and the incompleteness of the datasets.
- Defence Evasion, Persistence and Discovery are the preferred tactics for the exploits.
- Most exploits target web and client-side related vulnerabilities.
- The top 10 weaknesses account for almost two thirds (64%) of the vulnerabilities.

# 1. INTRODUCTION

## 1.1 MOTIVATION AND AIMS

Vulnerabilities are weaknesses leveraged by adversaries to compromise the confidentiality, availability or integrity of a resource; such weaknesses may result from design choices[4]. The market for vulnerabilities has become vibrant in recent years, with different stakeholders and threat actors taking advantage of the opportunities afforded by exploits on offer. Once publicly known, structured information about vulnerabilities is curated in public repositories such as the National Vulnerabilities Database[5] (NVD), while unstructured information is stored and discussed on online forums and locations.

Structured information about vulnerabilities facilitates widespread and timely sharing of information. Significant efforts are now being made to standardise this information to reduce communication barriers and complexity, leading to more effective analysis of vulnerabilities and a better understanding of the context within which different vulnerabilities are discovered. These efforts are, however, fraught with difficulty due to the challenges of categorising vulnerabilities. Vulnerability data can be incomplete, inaccessible, or inaccurate, and the quality of the resulting information has an impact on decision making, policies, and practices. Moreover, the vulnerability disclosure is influenced by a variety of factors, including financial incentives, the agenda of the disclosing stakeholder, the interaction of the various actors and this is all performed in a highly dynamic information security market (ENISA, 2015, 2018).

This study aims to address these challenges by completing three objectives:

1. Represent the state of cybersecurity vulnerabilities in a form allowing stakeholders to make informed decisions on cybersecurity investments.
2. Comprehensively analyse and correlate vulnerability data to better contextualise vulnerabilities.
3. Analyse vulnerability data from a quality and reliability perspective.

## 1.2 STANDARDISATION IN VULNERABILITY MODELLING

Disclosed vulnerabilities are usually uniquely identified, similar to books and publications receiving an ISBN number. The most pervasive vulnerability identification and numbering scheme is the Common Vulnerabilities and Exposures (CVE) referencing system instigated by MITRE[6]. Identifiers are assigned by CVE Numbering Authorities[7] (CNAs), i.e. organisations that are authorized to assign CVE IDs to vulnerabilities affecting products, and vulnerability information typically includes a brief description, advisories, mitigation measures and reports. Other numbering schemes include Microsoft's Security Bulletin (MS), Seebug's Vulnerability Database (SSV) and VMWare's Security Advisory (VMSA).

As the de facto standard, other schemes now typically map to CVE entries. Although this report adopts the CVE convention, not all publicly disclosed vulnerabilities have an associated CVE-ID. Vulnerabilities kept private and not publicly disclosed are often referred to as "zero-day vulnerabilities", and the corresponding exploits are referred to as zero-day (0day) exploits.

Certain events in the lifetime of a vulnerability and the time periods between such events can be significant in the sense that they may influence the risk of the underlying system and provide

[4] https://www.enisa.europa.eu/topics/csirts-in-europe/glossary/vulnerabilities-and-exploits
[5] https://nvd.nist.gov/search
[6] https://cve.mitre.org/.
[7] https://cve.mitre.org/cve/cna.html

opportunities to the adversaries. For example, there are normally delays in recording CVE and disclosure. There are also cyclic events associated with the battle between attacker exploitation and defensive code patching. This overall chronology is referred to as the vulnerability lifecycle (see Section 1.3).

The severity of the impact of a vulnerability is defined using the Common Vulnerability Scoring System (CVSS)[8] maintained by the Forum of Incident Response and Security Teams (FIRST). It is often provided as a qualitative value (Low, Medium or High); this is based on a quantitative calculation derived from the characteristics of individual vulnerabilities. The current CVSS version is v3.1, implemented in June 2019 replacing v3.0, but v2.0 values are often quoted for vulnerabilities prior to June 2015 when v3.0 was published.

Vulnerabilities are associated with information technology systems and software packages; these can be categorised using the Common Platform Enumeration (CPE) database[9]. Exploits and Exploit Kits[10] are developed to take advantage of the vulnerabilities. Exploitable vulnerabilities are usually based on software weaknesses. Vulnerabilities can also be captured using CWE™ (Common Weakness Enumeration): *"a community-developed list of common software security weaknesses"* and provides developers with advice on *"prioritizing software weaknesses in a consistent, flexible, open manner"*[11]; CWE entries rely on the Common Weakness Scoring System (CWSS).

A comprehensive understanding of cyber-attacks and the Cyber Kill Chain [12](CKC) requires awareness of the vulnerability lifecycle including development of vulnerabilities into exploits. The CKC also provides aspects of threat intelligence by assigning some attacker behaviours to specific events and uses model descriptions to comprehend those behaviours. This knowledge helps operators of targeted systems determine a successful defence strategy and solutions to certain cyber-attack problems.

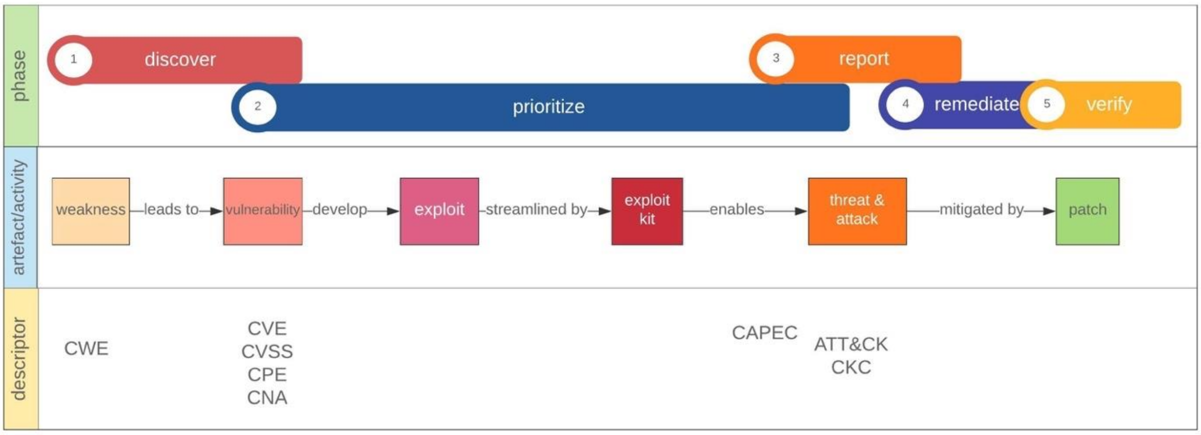
The Cyber Kill Chain (CKC) (Hutchins et. al, 2010) models cyber-attacks as event sequences from reconnaissance through exploitation to command and control of defender systems to achieve attacker actions on their objectives. For example, such a chain might model an attack that also discloses confidential information via malware. By modelling threats, the CKC helps determine their severity and explains how they are enacted. The CKC was used in the ENISA annual threat landscape report 2018 (ENISA, 2019).

MITRE also maintain Common Attack Pattern Enumeration and Classification (CAPEC) which is a *"dictionary of known patterns of attack employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. It can be used by analysts, developers, testers, and educators to advance community understanding and enhance defences."* CAPEC provides an understanding of how adversaries operate thus supporting effective cybersecurity. This can be complemented by ATT&CK™[13], which records attack tactics, techniques and procedures using matrices to map techniques to specific tactics; together this can be used to represent CKCs. The ATT&CK framework's tactics, although not explicitly referred to as a kill chain phase, could be considered as such.

These threat intelligence standardisation activities are summarised in Figure 1.

[8] https://www.first.org/cvss/
[9] https://nvd.nist.gov/products/cpe/search
[10] https://www.trendmicro.com/vinfo/us/security/definition/exploit-kit
[11] https://cwe.mitre.org/
[12] https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
[13] https://attack.mitre.org/

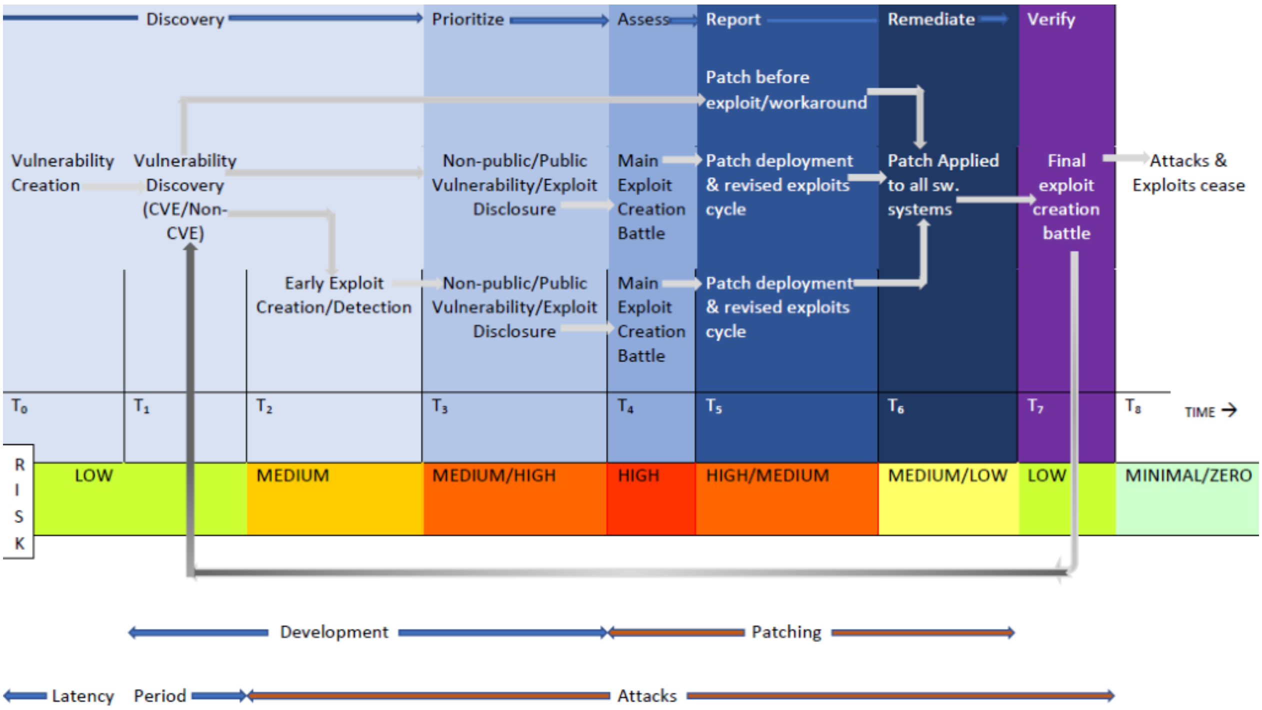
**Figure 1:** Vulnerability model context




## 1.3 VULNERABILITY MANAGEMENT

Early work defining the concept of the modern vulnerability lifecycle was reported by Arbaugh *et al.* (2000). The lifecycle is depicted in Figure 2. Mapping the vulnerability lifecycle identifies significant milestones and events that define risk transitioning boundaries. The significance of risks increases as vulnerabilities trigger the creation of the associated exploits and decrease when the patches become available.

**Figure 2:** Vulnerability lifecycle



# 2. METHODOLOGY

## 2.1 RESEARCH APPROACH

The research approach is depicted in Figure 3. The main steps and phases include the formation of a panel of experts, the formulation of the research questions, the identification of the data sources, the establishment of an appropriate schema, data collection, cleansing and curation, the development of the Jupyter notebooks and the actual analysis. Although the process followed these steps in a linear manner, the involvement and intervention of the experts throughout the duration of the project introduced a degree of iterations in order to refine the quantity and quality of the research questions as well as the creation of an appropriate dataset.

**Figure 3:** Research approach


### 2.1.1 Experts Group Formation

The experts who participated and contributed to the project were from the cyber security industry and academia, as well as from an EU Organisation (CERT-EU). As this project was led by ENISA together with an academic team, it was critical to involve experts from non-academia to validate the work and establish a level of analysis and communication of the results appropriate to the target stakeholders. As such, with regards to the academic experts, Tallinn University of Technology was invited as they also maintain an academic CERT. Including experts from academic CERTs and CERT-EU enabled a better collaboration with the Industry experts.

The experts contributed in two aspects. First, they reviewed the proposed research questions in order to affirm that the research effort would address questions that would be of added value to the industry and organisations. Second, they supported the data collection process either by directing the researchers to the appropriate sources, or by contributing with their own datasets, which improved the speed of collection. It should be highlighted though that all data used in this research are open source.

### 2.1.2 Formulation of research questions

As the aim of this study is to allow the stakeholders to make informed decisions on cybersecurity investments through the exploration and correlation of the vulnerability data, a representative list of questions was established from the outset. However, since this study is accompanied with the analysis scripts and dataset, these questions as well as the attempted answers should be viewed as a starting point to empower organisations to conduct further research by enriching both the questions and the dataset.

The questions were mainly related to the evaluation of the quality and quantity of vulnerabilities, where quality denotes the information provided by vendors and third parties regarding identified characteristics of the vulnerability as well as the information related to identified exploits and the industry's reflection.

The answers provided to these questions are anticipated to provide valuable input to the security industry with regards to the way that identified vulnerabilities are being handled, so that to improve, if necessary, their capabilities to accurate and timely identification and evaluation of exploits and remediation efficiency.

As such, issues related to the vulnerabilities and exploits sources, timings in identification, evaluation and remediation of vulnerabilities and exploits, standardisation and adoption thereof with regards to information formatting and communication will be addressed through this research.

### 2.1.3 Identification of Data Sources

A variety of sources were identified through the course of the study. The following types of sources were considered:

- Vulnerability Databases and taxonomies as listed in Figure 4, and articles about databases. A recent research considering both vulnerability management aspects and data sources by Kritikos *et al*. (2019) was consulted.
- Articles on specific vulnerabilities offering more detail. For example, ThreatConnect are among several threat intelligence aggregators and providers who collect and consolidate information on vulnerabilities' appearance.
- Vendors Bulletins (essentially unstructured data)
- General News Sources (essentially unstructured data)
- Tools and sources for mapping vulnerability information to MITRE's ATT&CK taxonomy.
- Economics Sources (associated with black market exploit and vulnerability prices plus general market volume analysis data)

In order to compile the required datasets, these sources were studied and evaluated, to ensure that the data obtained from them would be appropriate, relevant and of sufficiently high quality. Based on the research questions that were defined in the beginning of the project, certain criteria were set, which were in turn used for performing the selection of the data sources, thus leading to the list presented in **Error! Reference source not found.**5. The main requirement for t he selection of each specific data source was that the data it provides must be free of charge and primarily in the form of structured datasets. The information provided by a data source was also evaluated in terms of accuracy, consistency and completeness by considering additional external references from other well-established sources or standards (e.g. CVE, BID, CWE, etc.). The ease of data extraction from a given source was an additional quality that was taken into consideration.

The selected data sources were divided into three categories: a) The ones that had a CVE ID assigned to the vulnerability data they provided (CVE data), b) the ones that did not (non-CVE data) and c) the ones that provided information relating to CVEs (CVE-related), such as CAPEC IDs, CWEs, etc.

Although locating CVE data and CVE-related sources was relatively easy, the collection of non-CVE data proved to be a more difficult task, primarily because such sources are limited in number and they normally provide their data for a fee. Nevertheless, non-CVE vulnerabilities may represent a significant proportion of all vulnerabilities.

**Figure 4:** Data sources

| Source | Type of data | Description |
|---|---|---|
| **NVD database** | CVE data | https://nvd.nist.gov/ The NVD is the U.S. government repository of standards-based vulnerability management data. The NVD includes databases of security checklist references, security-related software flaws, misconfigurations, product names, and impact metrics. |
| **ATT&CK** | Attacker's patterns (techniques & tactics) | https://attack.mitre.org/ MITRE ATT&CK™ is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. |
| **Shodan*** | Number of exploits | https://www.shodan.io/ Database of internet connected devices (e.g. webcams, routers, servers, etc.) acquiring data from various ports (e.g. HTTP/HTTPS - port 80, 8080, 443, 8443). |
| **Exploit database*** | Non-CVE data | https://www.exploit-db.com/about-exploit-db contains information on public exploits and corresponding vulnerable software. The collection of exploits is acquired from direct submissions, mailing lists and other public sources. |
| **CVE details** | CVE data | https://cve.mitre.org/ CVE® a database containing details of individual publicly known cybersecurity vulnerabilities including an identification number, a description, and at least one public reference. |
| **Zero Day Initiative*** | CVE and non-CVE | https://www.zerodayinitiative.com/ encourages reporting of zero-day vulnerabilities privately to affected vendors by financially rewarding researchers (a vendor-agnostic bug bounty program). No technical details on individual vulnerabilities are made public until after vendor released patches. ZDI do not resell or redistribute the vulnerabilities . |
| **ThreatConnect*** | Number of incidents related to CVE | https://threatconnect.com/ Automated threat intelligence for Intel systems |
| **VulDB*** | Exploit prices and software categories | https://vuldb.com/ vulnerability database documenting and explaining security vulnerabilities and exploits |
| **US CERT** | Industry sector | https://www.us-cert.gov/ The US Department for Homeland Security's Cybersecurity and Infrastructure Security Agency (CISA) aims to enhance the security, resiliency, and reliability of the USA's cybersecurity and communications infrastructure |
| **Zerodium** | Bug bounty exploit prices | https://zerodium.com/ A zero-day acquisition platform. Founded by cyber security experts with experience in advanced vulnerability research. |

*Commercial database

The data sources were combined in order to produce a rich dataset with a variety of features (or dimensions). As many of the sources had feature overlaps it was possible to create a superset contextualising each vulnerability further. For instance, the CVE ID was used to link information between the weaknesses, industries, number of exploits and incidents. The CAPEC that was included in both NVD and ATT&CK allowed mapping of a CVE to different attack techniques and tactics.

Combining different sources to produce one dataset inevitably led to having empty values, as the different data sources do not necessarily overlap horizontally. However, the number of vulnerabilities is adequate (over 27K) to reliably conduct statistical tests and in most cases the sample sizes where adequate to draw safe conclusions.

### 2.1.4 Establishing a schema

The data schema was developed on the basis that the analysis would be conducted through a collection of purpose-built Python Jupyter notebooks. As such, priority was given to having a dataset in a form that is most suitable for a sequential analysis utilising the notebook(s). Moreover, it was possible to estimate the size of the dataset reasonably accurately, both in terms of features/columns as well as number of observations/elements/rows. Furthermore, it was recognised that flat files (that can be easily imported into Python's Pandas data frames) offer advantages over maintaining other and potentially more complex structures such as hierarchical databases.

### 2.1.5 Data collection, cleansing, normalisation and curation

This study covers the period of vulnerabilities published between **January 1st 2018** to **August 31st (Q1 – Q3) 2019**. The vulnerabilities were regularly collected and hosted in the compiled dataset until the cut-off date of **September the 30th**. As such, the NVD "snapshot" of the dataset reflects that date and there are likely to be more vulnerabilities included in the period under examination today. This is due to the lag of vulnerabilities getting officially a CVE ID and entering the system.

In order to further contextualise the vulnerability entries, several dictionaries were also downloaded, see Figure 5: These files were used through mapping and lookup functions to translate the numerical values and IDs to more meaningful information.

**Figure 5:** Dictionaries

| Dictionary | Description |
|---|---|
| **CAPEC** | CAPEC ID to attack pattern. Downloaded from capec.mitre.org |
| **CWE** | CWE IDs to weakness description. Downloaded from cwe.mitre.com |
| **ATT&CK tactics** | ATT&CK tactic ID to description. Compiled. |
| **ATT&CK techniques** | ATT&CK technique ID to description. Compiled. |
| **CPE** | CPE list to product and vendor information downloaded from nvd.nist.gov/products/cpe |

It should be noted that some of the dictionaries did not include complete information leading to further missing values. For instance, the CPE list did not include a complete mapping between the CPE vector and product/vendor information. Although some entries were manually created and added, it is highlighted that the incomplete information is common among such data sources.

### 2.1.6 Development of Jupyter notebooks

The analyses have been conducted within Jupyter notebooks. The open-source Jupyter Notebook technology enables the creation of documents using the literature programming paradigm, where code, narrative, and interactive visualisations can be blended together. These documents can be easily shared or downloaded from the web, and contain live code which enables complete reproducibility.

### 2.1.7 Analysis

The analysis employed - where appropriate - statistical methods and techniques to assess the significance of the findings. Although this report that accompanies the Jupyter notebooks contains fewer details on the analysis, it does however summarise the main and most significant findings or those that the contributors deemed to have the highest potential, added value and impact. Moreover, the statistical approaches presented in the report are explained in a higher level of detail in the Jupyter notebooks.

#### 2.1.7.1 Statistical tools

A basic test for any distribution is to check whether the distribution is **normal**, as in such case it would enable one to run a wide variety of statistical tests, namely **parametric tests**. A caveat however when running normality tests on large (>100) samples, is that most tests fail as they are sensitive (strictly speaking, they are dependent on the standard error which becomes very small as it includes the square of sample size in its denominator). As such, it is also important to visually inspect the distribution and decide whether parametric tests could still be used, accepting of course lower confidence trade-offs. This work employs **Kolmogorov-Smirnov's** test for normality.

Distribution comparison refers to checking whether the means of two distributions (or samples) are equal. **Wilcoxon test** is the most common test and used in this report.

The **independent t-test** is a popular approach to test the hypothesis of two samples having (significant) different means. The t-test is applicable to variables that have two levels (e.g. before and after, or present/absent). If the variables have more than two levels, **analysis of variance** (ANOVA) is performed. One-way ANOVA is performed to compare the means of three or more groups of data over one independent variable, whereas two-way ANOVA is used to compare the means of three or more groups over two independent variables. An example of a one-way ANOVA is to check if the CVSS means of the ATT&CK tactics (groups) are different. Two-way ANOVA can be used to check for instance if there is significance in the interaction between publication delays and severity of a vulnerability.

Both hierarchical clustering and factor analysis aim to explore more intrinsic relations. **Hierarchical clustering** is performed by considering each data sample as a vector and measures the distance from the other data items. The output can be in the form of a **dendrogram**, where the leaves represent the factors or variables and the length of the paths between them show their distance. The dendrogram is structured in a way that the more we travel towards the root, the bigger the clusters that are formed, reducing thus the total number of clusters. Normally, a horizontal line threshold is drawn about halfway through the height of the dendrogram and the number of intersecting branches show the number of clusters. In this work **Ward's** approach is used.

While hierarchical clustering gives a high level and visual representation of clusters, **factor analysis** can provide a more detailed and quantitative description of the potential factors. Factor analysis aims to consolidate and group variables with a view to perform **dimension reduction** over the data variables. A necessary condition to perform factor analysis is first to check if the correlation matrix is not the identity matrix. In this work, this is performed by **Bartlett's sphericity** test.

Following factor analysis, **Chronbach's alpha** measures the internal consistency of variables (items) and indicates how closely related they are as a group. This measure is used to assess whether the variables can be combined to indeed form a factor (also referred to as a latent, unobserved variable). Normally for values over 0.70 it is commonly accepted that the variables can indeed be grouped. For instance, the CVSS formula groups Confidentiality, Integrity and Availability into the Impact subcategory. The Cronbach's alpha value for these three features is 0.83 which is rather high and shows that the choice to have these three under one group was a correct design decision. Moreover, this high value is particularly interesting as it directly relates to the main goals and essence of information security.

#### 2.1.7.2 Economic aspects

Analysis of the economic aspects of vulnerabilities is a critical, complex, and sometimes more esoteric to the topics discussed above. The economic and financial impact is closely coupled to an organisation's risk assessment and risk management approach and is outside the scope of this study. However, this report considers the economic aspects of the development of exploits and includes a high-level analysis of the exploit prices. To this end, bug bounty prices were included (from Zerodium) as well as the exploit price estimates by VulDB who maintain a proprietary estimation algorithm.

In the next Section all findings presented are of statistical significance, when an applicable test was employed. To conduct further and independent data exploration refer to the Jupyter notebooks collection at https://github.com/enisaeu/vuln-report.

# 3. ANALYSIS & FINDINGS

## 3.1 EXPLORATORY DATA ANALYSIS

As mentioned earlier, the vulnerability dataset contains missing values due to the missing data from the source datasets but also due to the joining operation of the different sources. Figure 6 summarises the sample size of the respective columns.

**Figure 6:** Main dataset items population

| Feature: count (%) | | | |
|---|---|---|---|
| vulnerabilities: 27,471 (100.00) | CAPEC: 21,335 (77.66) | 0-day price: 3,390.0(12.34) | sector: 137(0.50) |
| CNA: 27,471(100.00) | CPE: 27,462 (99.97) | current price: 3,390.0(12.34) | incident: 2,169.0(7.90) |
| CVSS v2 score: 27,471(100.00) | ATT&CK technique: 8,077 (29.40) | platform: 2,371(8.63) | End of support: 381(1.39) |
| CVSS v3 score: 27,471(100.00) | ATT&CK tactic: 8,067 (29.37) | vendor: 23,110(84.13) | exploit date: 2,371(8.63) |
| CWE: 27,471(100.00) | CVSS change history: 308(1.12) | product: 23,108(84.12) | exploit verified: 2,371(8.63) |
| Software type: 3,369 (12.26) | | | |


Word cloud output from the main dataset's description field

As mentioned in the previous section, the main source of CVEs is the National Vulnerability Database. As many organisations use this as the authoritative source of CVE entries, it is worth investigating the completeness of this source. It was found that some CVEs that appeared in other databases, did not have up to date entries in NVD. Consider for example, CVE-2018-

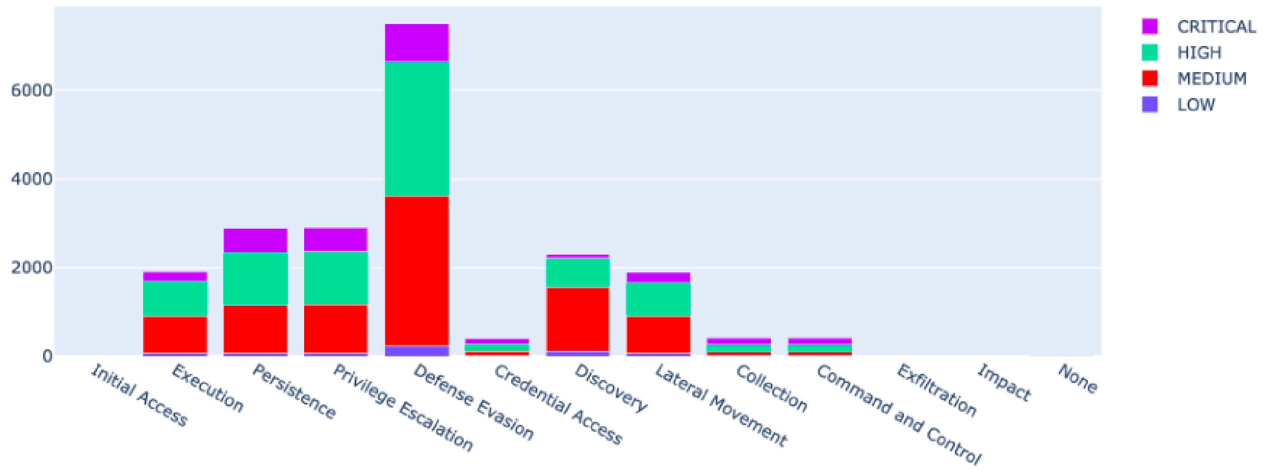
14319 (CVSS score 6.8) which is a buffer overflow / Remote Code Execution vulnerability affecting Samsung Galaxy S8 smartphones. This vulnerability:

- was reported to the vendor in Q1 of 2018, (April 2018, ZDI[14]),
- was reported by the vendor through an initial advisory in August 2018 (SVE-2018-11828), classifying it as a CRITICAL level vulnerability[15],
- did not secure an entry in NVD[16].

From the above the following observations are made. First, organisations that rely solely on one source – no matter how authoritative it may be – will potentially miss vital vulnerability information affecting their systems. Second, the CVSS scoring system, although capable of providing potentially a good reference for assessing, understanding and contrasting the impact and severity of vulnerabilities, does not necessarily forms the “ground truth” as other stakeholders may have a different view on the severity such as Samsung in this case who are the vendor of the vulnerable product.

A non-uniform distribution of vulnerabilities over the ATT&CK tactics is evident in Figure 7. Defense Evasion is clearly by far the most “popular” tactic for vulnerabilities, whereas three tactics do not have any representation in the 2018 dataset. It should be noted that some vulnerabilities are counted more than once, if they appear in multiple tactics.

**Figure 7:** Distribution of vulnerabilities over ATT&CK tactics and CVSS v3 base score (29.37% of vulnerabilities in dataset)




The ATT&CK framework is constantly enriched with techniques and sub-techniques. At the time of writing, the number of techniques recorded to 291. In the 2018 dataset, 52 techniques were associated with vulnerabilities, accounting to the 17% of the complete ATT&CK techniques range. The top 3 techniques associated with vulnerabilities are:

**T1148** – HISTCONTROL (4226 vulnerabilities)[17]
**T1027** – Obfuscated Files or Information (2293 vulnerabilities)[18]
**T1130** – Install Root Certificate (1813 vulnerabilities)[19]

[14] https://www.zerodayinitiative.com/advisories/ZDI-18-1450/
[15] https://security.samsungmobile.com/androidUpdatesSearch.smsb
[16] https://nvd.nist.gov/vuln/detail/CVE-2018-14319
[17] https://attack.mitre.org/techniques/T1148/
[18] https://attack.mitre.org/techniques/T1027/
[19] https://attack.mitre.org/techniques/T1130/

Out of the 52 techniques, 35 have over 1600 appearances in vulnerabilities, following with a large dip (443 vulnerabilities and below) for the remaining 17. The techniques are explored further in Section 3.12.

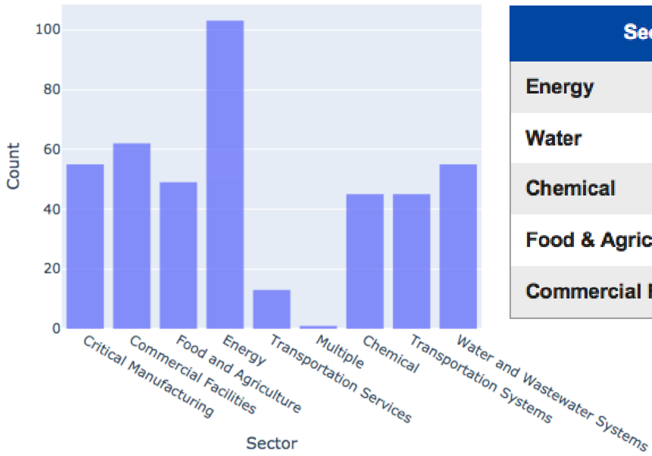
## 3.2 VULNERABILITIES PER SECTOR

Figure 8 summarises the vulnerabilities per sector as well as the number of critical vulnerabilities for each sector.

**Vulnerabilities with the highest cross-sectoral impact:**

**CVE-2019-1639**
**CVE-2019-1638**
**CVE-2019-1636**

**Figure 8:** Vulnerabilities per sector

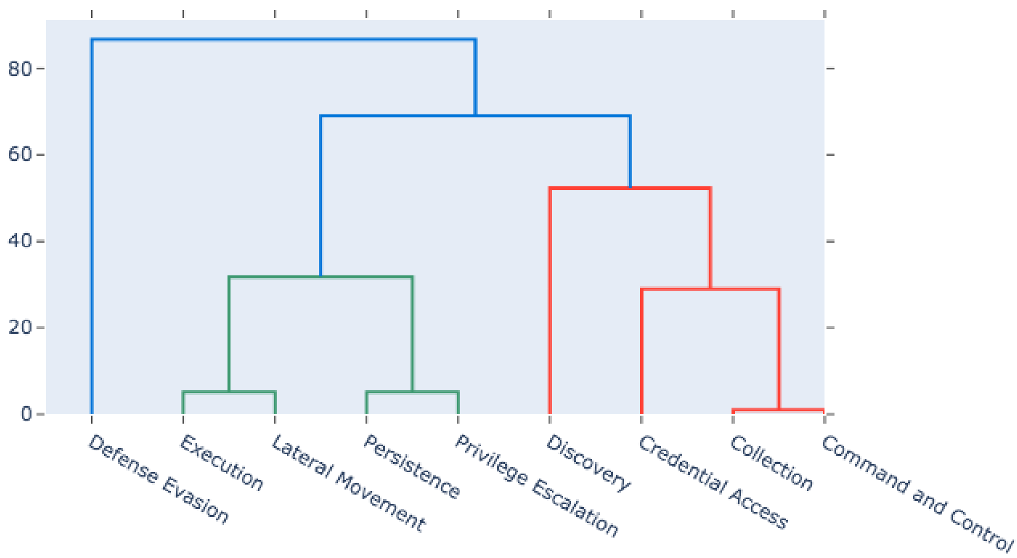



| Sector | Number of CRITICAL CVEs |
|---|---|
| Energy | 5 |
| Water | 2 |
| Chemical | 1 |
| Food & Agriculture | 1 |
| Commercial Facilities | 1 |

## 3.3 HIERARCHICAL CLUSTERING & FACTOR ANALYSIS

Clustering and factor analysis can show more intrinsic relationships existing in the data. Starting with hierarchical clustering, Figure 9 shows the dendrogram after performing Ward's distance approach on the ATT&CK tactics.

**Figure 9:** Hierarchical clustering results on ATT&CK tactics



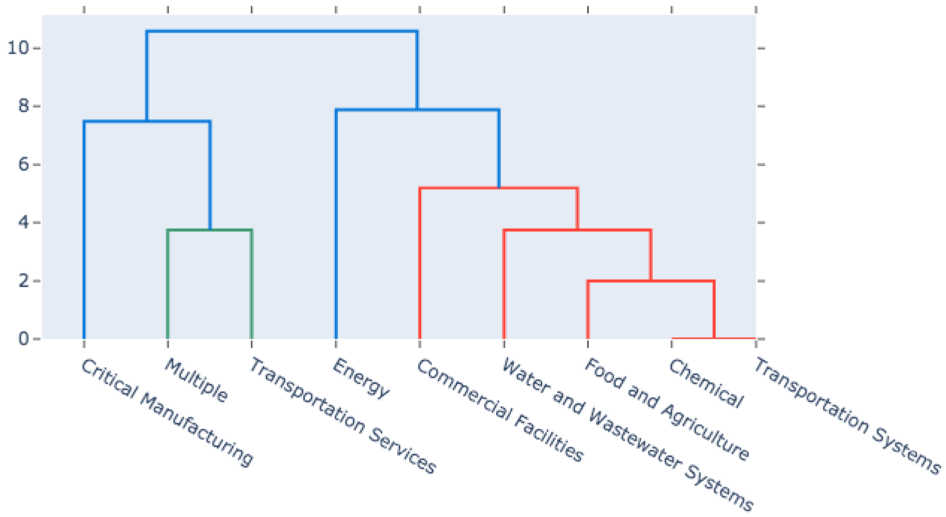
From the above there are some clear clusters emerging: Collection & Command and Control, which then can be grouped with Credential Access, and further with Discovery. On the other side, Privilege Escallation with Persistence form a cohesive cluster which join Execution and Lateral Movement.

However, this approach provides a more qualitative and intuitive representation. A more quantitative and detailed approach is achieve through factor analysis. Following this approach, the results are summarised in Figure 10. It should be noted that the initial analysis suggested 3 factors, but following an internal consistency check (using Cronbach's alpha), 4 factors were confirmed. The loadings column shows the weight of each of tactic in the respective factor.

**Figure 10:** Factor analysis results.

| **ATT&CK tactic factors** | | | |
|---|---|---|---|
| **Groups** | **Loadings** | **Cronbach's alpha** | **Comment** |
| **Factor 1** | | | |
| Collection<br>Command & Control | 0.9937<br>0.9939 | 0.9994 | A strong relationship between these two tactics |
| **Factor 2** | | | |
| Credential Access | (-0.5404) | - | n/a |
| **Factor 3** | | | |
| Defense Evasion | (0.7429) | - | n/a |
| **Factor 4** | | | |
| Discovery<br>Execution<br>Lateral Movement<br>Persistence<br>Privilege Escalation | -0.3767<br>0.9388<br>0.9314<br>0.9192<br>0.9217 | 0.7335 | A good internal consistency, as the result is > 0.70 |

**Figure 11:** Hierarchical clustering results on Sector (Zero Day Initiative sample)




A similar approach was followed for the sector categories, Figure 11. Quite interestingly, Transportation Systems are closely coupled with Chemical and are distant from Transportation Services. A clear cluster is shown for Water, Food and Agriculture, Chemical, Transportation and Commercial Facilities. Vulnerabilities in Energy are quite distinct from those found in the other sectors.

**Vulnerabilities in Energy and Critical Manufacturing are distant from all others**

## 3.4 CVSS

CVSS scores are important metrics as they provide a quantitative measure that can be eventually used to inform risk exposure. Organisations use CVSS to make judgements on their vulnerabilities based on this metric. In the following analysis the CVSS versions are compared and contrasted to establish whether the different measuring systems have differences leading to potentially different decisions.

### 3.4.1 Comparison between CVSS v2 and v3

**Figure 12:** Distribution of CVSS v2 and v3 base scores

| | cvss2_bscore | cvss3_bscore |
|---|---|---|
| count | 27471.000000 | 27471.000000 |
| mean | 5.749459 | 7.339012 |
| std | 1.888787 | 1.612830 |

It is visibly evident from Figure 12 that the distributions using v2 and v3 are significantly different, with CVSS v3 having a considerably higher mean (7.34) than CVSS v2 (5.75). The biggest difference was observed for CVE-2019-12373 (Ivanti LANDESK Management Suite) with a value of 6.3 (CVSS v2: 2.7, CVSS v3: 9.0)

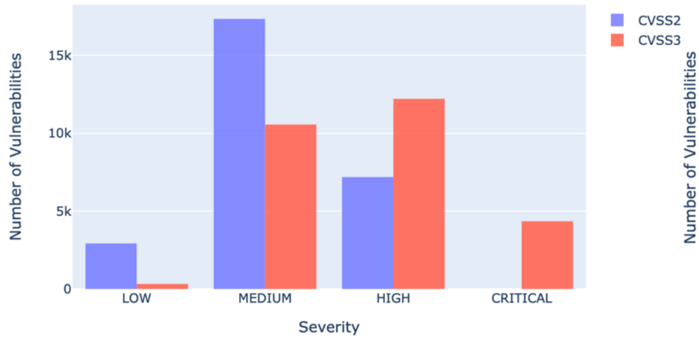
Severity and impact are contrasted in Figure 13. Again, there are visible differences between the two CVSS versions. In this case, the severity for v3 was higher than that of v2, but the opposite was true for impact (CVSS v2 higher impact than v3). Furthermore, the correlation results for the three impact categories were as follows:

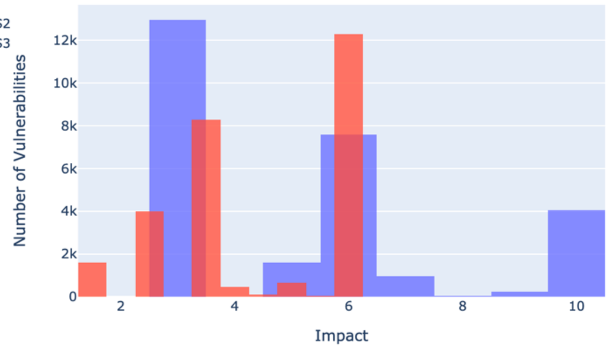
**There are 4 potentially misclassified CRITICAL and exploitable vulnerabilities affecting confidentiality, 42 for integrity, 46 for availability**

- Confidentiality: **0.79**
- Integrity: **0.34**
- Availability: **0.38**

The correlations could arguably be considered low, given that both versions have three levels per impact variable (Confidentiality, Integrity, Availability), but different qualitative definitions, namely for version 2: “none”, “partial”, “complete” and for version 3: “none”, “low”, “high”. The different definitions could be sufficient to be responsible for the associated bias when assigning the values.

**Figure 13:** Severity and Impact



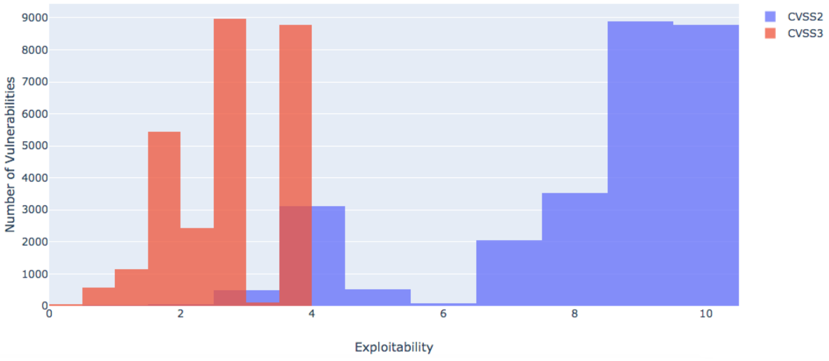

Exploitability attempts to capture the ease (or complexity) by which the vulnerability can be exploited. Although the underpinning metrics slightly differ between versions 2 and 3, exploitability shows the highest contrast between the two CVSS versions (Figure 14). It should be noted that these two versions have different definitions leading to different ranges with version 3 defined in a range of 0-3.9, whereas version 2 operates in the range of 0-10 and over three quarters of vulnerabilities have an exploitability score over 8.

The correlation coefficient between the two exploitability versions is **0.79**, which could be considered marginally acceptable, given that both versions represent the same aspect. Inspecting the different components of exploitability, those that showed the **highest correlation** (between the two versions) were **user interaction** (v3) with **access complexity** (v2) with a correlation of **0.85,** and **attack complexity** (v3) with **access complexity** (v2), with a value of **0.78**.

**CVSS V2 & V3**

There are considerable differences between the two scoring systems leading to different severity classifications, that may in turn affect the risk management, planning and decision-making processes.

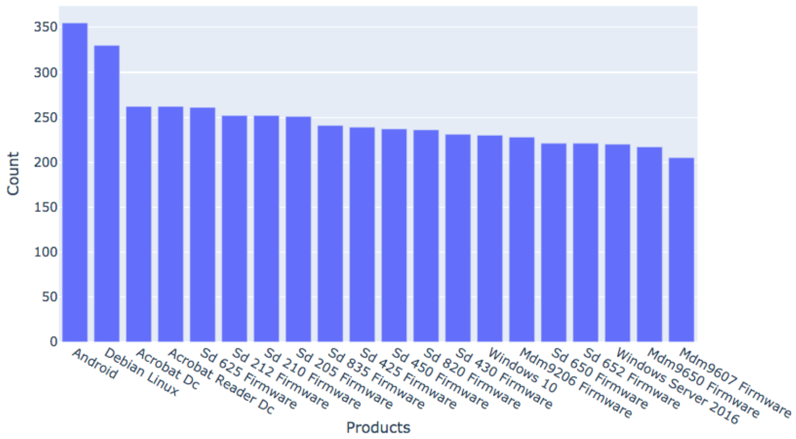
**Figure 14:** Exploitability




## 3.4.2 High CVSS score vulnerabilities

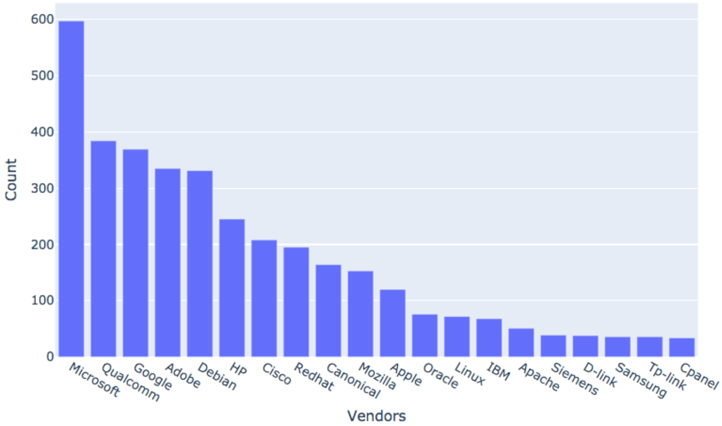
The following series of graphs refer to vulnerabilities with score equal or greater than 7 (High, Critical). Figure 15 shows the top 20 **products** with the most (and high) CVEs and Figure 16 the top 20 **vendors** with the most CVEs. Figure 17 presents the mean scores of the vulnerabilities by vendor and ATT&CK tactic.

**Figure 15:** Top 20 products with the most CVEs (having score >=7)




The product with the highest number of vulnerabilities is the Android OS, followed by Debian Linux and Acrobat reader. The list is dominated by Qualcomm's Sd series Firmware. Further down the list lie more Windows based operating systems whereas Edge is in position 26. Note that this distribution considers vulnerabilities by products; that is, if a particularly vulnerability affects more than one product it will be counted twice (or more times for that matter, depending on the number of products it affects). A different view of the above is presented if the vulnerabilities are grouped by vendor, where in this case Qualcomm is a runner-up to Microsoft (Figure 16).

**Figure 16:** Top 20 vendors with the most CVEs (having score >=7)




Microsoft has the highest number of vulnerabilities (600) which is more than 50% higher than the runner-up, Qualcomm. From an ATT&CK framework perspective (Figure 17), Cisco and Canonical have high scored vulnerabilities across the range of the tactics, whereas Microsoft is ranked among the lowest.

**Figure 17:** Vendor vs. ATT&CK tactic vs. CVSS score (mean) heatmap

| | Collection | Command and Control | Credential Access | Defense Evasion | Discovery | Execution | Lateral Movement | Persistence | Privilege Escalation |
|---|---|---|---|---|---|---|---|---|---|
| Tp-link | 9.8 | 9.8 | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 | 8.9 | 8.9 |
| Siemens | 9.8 | 9.8 | 9.8 | 8.4 | 0.0 | 8.8 | 8.9 | 9.0 | 9.0 |
| Samsung | 0.0 | 0.0 | 0.0 | 7.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Redhat | 9.8 | 9.8 | 8.8 | 9.1 | 8.8 | 8.8 | 8.8 | 9.0 | 9.0 |
| Qualcomm | 8.8 | 8.8 | 7.8 | 9.0 | 8.7 | 8.7 | 8.7 | 8.7 | 8.7 |
| Oracle | 0.0 | 0.0 | 9.8 | 8.4 | 0.0 | 8.4 | 8.4 | 8.4 | 8.4 |
| Mozilla | 0.0 | 0.0 | 0.0 | 9.5 | 0.0 | 9.0 | 9.0 | 9.0 | 9.0 |
| Microsoft | 7.0 | 7.0 | 0.0 | 8.5 | 0.0 | 8.9 | 8.9 | 8.7 | 8.7 |
| Linux | 0.0 | 0.0 | 0.0 | 7.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| IBM | 8.5 | 8.5 | 9.3 | 8.3 | 0.0 | 8.2 | 8.2 | 8.3 | 8.3 |
| HP | 9.1 | 9.1 | 9.1 | 8.9 | 8.1 | 8.9 | 8.9 | 9.0 | 9.0 |
| Google | 0.0 | 0.0 | 0.0 | 8.4 | 8.8 | 8.7 | 8.7 | 8.7 | 8.7 |
| Debian | 9.8 | 9.8 | 0.0 | 9.4 | 0.0 | 9.8 | 9.8 | 9.4 | 9.4 |
| D-link | 9.8 | 9.8 | 9.8 | 9.0 | 9.8 | 8.5 | 8.5 | 8.9 | 8.9 |
| Cpanel | 0.0 | 0.0 | 7.2 | 8.5 | 8.4 | 8.8 | 8.8 | 7.8 | 7.8 |
| Cisco | 9.8 | 9.8 | 9.0 | 8.4 | 8.8 | 9.8 | 9.8 | 9.7 | 9.7 |
| Canonical | 9.8 | 9.8 | 0.0 | 9.2 | 0.0 | 9.8 | 9.8 | 9.8 | 9.8 |
| Apple | 0.0 | 0.0 | 8.2 | 9.0 | 8.2 | 9.2 | 9.2 | 8.8 | 8.8 |
| Apache | 7.2 | 7.2 | 0.0 | 8.9 | 0.0 | 9.8 | 9.8 | 9.2 | 9.2 |
| Adobe | 9.8 | 9.8 | 0.0 | 9.2 | 7.5 | 0.0 | 0.0 | 9.8 | 9.8 |

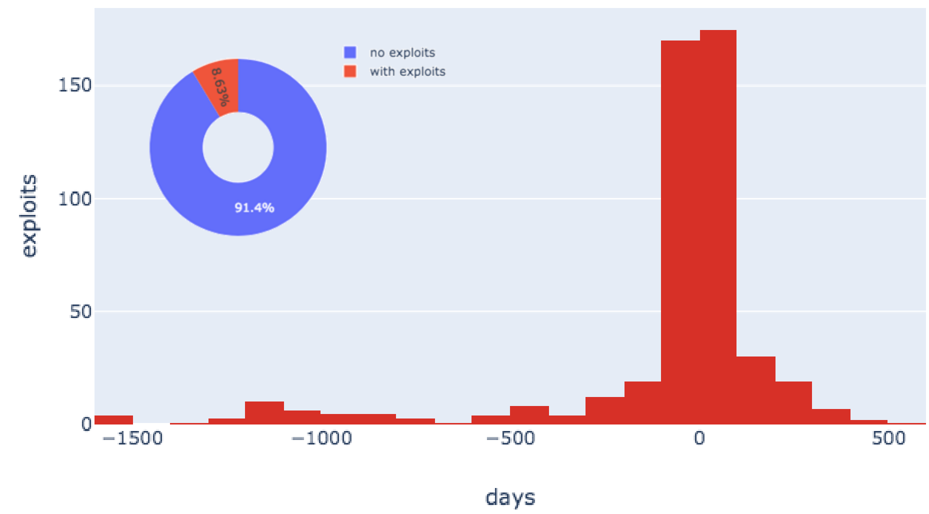
## 3.5 LIFECYCLE ANALYSIS

Among the key milestone dates of a vulnerability is its actual publication date by which the vulnerability becomes widely known, and the publication of the exploit; both these milestones normally put pressure to the vendor to come up with a security update. In a relevant note, End of Support (EOS) date is the point in time where the vendor is not expected to develop a patch. These are explored in the following sections.

### 3.5.1 Exploits

Figure 18 shows the percentage of vulnerabilities with exploits and their publication referencing the vulnerability publication date (t=0).

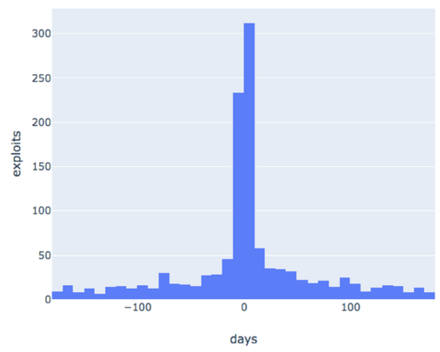
**Figure 18:** Exploit publication date (t=0: vulnerability publication date, sample size 8.63% of dataset)




**EXPLOIT PUBLICATION DATE**
Exploits of vulnerabilities of CRITICAL severity level form a constellation around the vulnerability publication date.

Note that the above results are expected to be skewed due to the upper-bound / cut-off time of the data. In order to proceed with the analysis, we need to ensure that the data are not biased. As such, we select only the vulnerabilities published in 2018 and allow a ±6-month window (±180 days) for the exploits. The distribution and descriptive statistics are presented in Figure 19. The descriptive statistics show the mean of difference per severity.

**Figure 19:** Exploits for 2018 vulnerabilities published within 6 months (before or after) of the publication date of the vulnerability.




| cvss3_severity | N | Mean | SD | SE | 95% Conf. | Interval |
|---|---|---|---|---|---|---|
| CRITICAL | 278 | 24.834532 | 39.475368 | 2.367575 | 20.194085 | 29.474980 |
| HIGH | 465 | 47.546237 | 53.245553 | 2.469203 | 42.706598 | 52.385875 |
| LOW | 7 | 52.714286 | 63.394832 | 23.960994 | 5.750737 | 99.677835 |
| MEDIUM | 463 | 46.313175 | 51.841324 | 2.409271 | 41.591004 | 51.035345 |

```
  Multiple Comparison of Means - Tukey HSD, FWER=0.05
=======================================================
 group1  group2 meandiff p-adj   lower    upper  reject
-------------------------------------------------------
CRITICAL   HIGH  22.7117  0.001  12.9753 32.4481   True
CRITICAL    LOW  27.8798 0.4638 -21.2682 77.0277  False
CRITICAL MEDIUM  21.4786  0.001  11.7343 31.2229   True
    HIGH    LOW    5.168    0.9 -43.7366 54.0726  False
    HIGH MEDIUM  -1.2331    0.9  -9.6647  7.1986  False
     LOW MEDIUM  -6.4011    0.9 -55.3073 42.5051  False
-------------------------------------------------------
```

Note that for CRITICAL severity levels, the mean is smaller than all other severity levels (24.83). A 2-way analysis of variance (ANOVA) shows that there is indeed significant difference between CRITICAL and all other severity levels, with the former clustered around the vulnerability publication date.

Figure 20 shows the outlier products with the highest (and fewest) exploits published before or after the publication of a vulnerability. The results suggest operating systems attract more exploits after the publication of a vulnerability.

**Figure 20:** Publication of exploits surrounding vulnerability publication date.

| **Exploits before and after the vulnerability publication date** | | | |
|---|---|---|---|
| **Product** | **# of exploits before vuln. pub. date** | **# of exploits after vuln. pub. date** | **Δ** |
| **Top 5 – least number of exploits post publication** | | | |
| Rational Quality Manager | 27 | 0 | -27 |
| Firmware | 117 | 93 | -24 |
| Rational Collaborative Lifecycle Management | 20 | 0 | -20 |
| Firefox | 23 | 5 | -18 |
| Thunderbird | 19 | 2 | -17 |
| **Top 5 – most number of exploits post publication** | | | |
| Domainmod | 0 | 11 | 11 |
| Windows Server 2016 | 13 | 28 | 15 |
| Windows Server | 18 | 33 | 15 |
| Windows 10 | 13 | 29 | 16 |
| Ubuntu Linux | 20 | 39 | 19 |

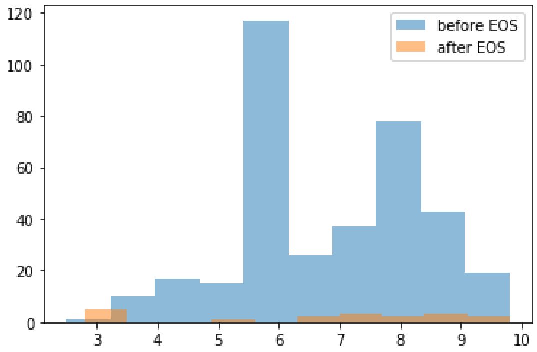
### 3.5.2 End of support (End of life)

A pivotal moment in the life of a software product is when the vendor ceases to support it. From a security perspective this is critical as it stops issuing patch updates. One of the reasons malwares like WannaCry created havoc in a short timeframe was because it exploited vulnerabilities of systems that were impossible to patch, simply because the vendor stopped supporting them.

Figure 21 shows the distribution of exploits published before or after the End of Support (EoS) date. The two groups (before and after EoS exploits) do not have equal variances, but there are no significant differences between their means.

**Figure 21:** End of support exploits.

| | EOS_date | date_exploit | vendor | product | EOS_version | cvss3_severity | cvss3_bscore |
|---|---|---|---|---|---|---|---|
| 17016 | 2019-06-30 | 2019-08-21 | Oracle | Vm Virtualbox | 5.0.38 | HIGH | 8.8 |
| 18827 | 2019-03-20 | 2019-04-19 | Atlassian | Confluence | 6.1 | CRITICAL | 9.8 |
| 21230 | 2019-01-12 | 2019-05-08 | Postgresql | Postgresql | 9.4 | HIGH | 7.2 |




**CVSS v3 base score comparisons:**

Levene's test p-value=0.002175 (significant)

t-test p-value=0.83262 (not-significant)

## 3.5.3 Disclosure after vendor notification grace period

When vulnerabilities are reported to the vendor, it is expected that they would issue a patch within a certain timeframe. This timeframe should be subject to the severity of the vulnerability, however this is not observed. Zero Day Initiative (ZDI) for example have a 120-day grace period. Following that, they disclose the vulnerability to the public, if the vendor does not issue any updates. The ZDI dataset contains 6 vulnerabilities (out of 2228) that fall into this category, 3 of which have a CVSS v3 score of 10 and only 2 have a CVE ID, whereas the other 4 are non-CVE (0-days). The high non-CVE vulnerability refers to Belkin's Wemo Link (ZDI-CAN-5206).

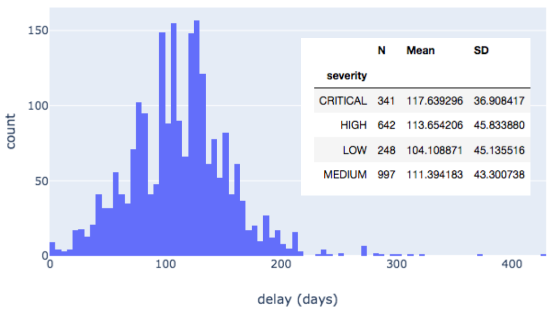
## 3.5.4 Publication delays

The following results refer to the ZDI dataset. On average, a vulnerability is published after 112 days after it has been reported (Figure 22). Also, it takes longer for CRITICAL and HIGH vulnerabilities to be published, than those with a LOW severity score.

**The top 10 CWEs account for the 64.20% of the vulnerabilities**

**Operating systems are at the top of unique CWEs, with the greatest variety of weaknesses**

**Figure 22:** Publication delays for vulnerabilities




ANOVA results:

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=======================================================
 group1  group2 meandiff p-adj   lower    upper  reject
-------------------------------------------------------
CRITICAL   HIGH  -3.9851 0.5153 -11.4542  3.4841  False
CRITICAL    LOW -13.5304 0.0011 -22.8328 -4.2281   True
CRITICAL MEDIUM  -6.2451 0.0992 -13.2378  0.7475  False
    HIGH    LOW  -9.5453 0.0172 -17.8791 -1.2116   True
    HIGH MEDIUM    -2.26  0.707  -7.9005  3.3804  False
     LOW MEDIUM   7.2853 0.0837  -0.6242 15.1948  False
-------------------------------------------------------
```

## 3.6 WEAKNESSES

Weaknesses are the causes leading to vulnerabilities. In the following set of graphs, the CWE data are analysed and presented.

Figure 23 presents the top 10 CWEs. Note that CWE with ID 264 refers to a category rather than a specific weakness, namely "Permissions, Privileges and Access Controls".

**Figure 23:** Top 10 weaknesses (CWEs)

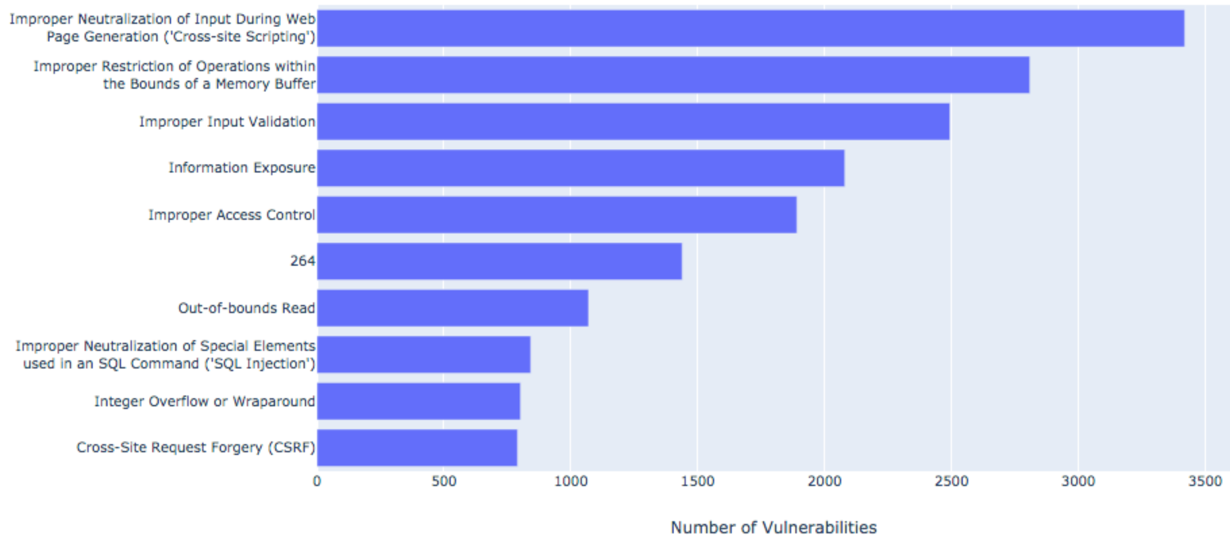



**Figure 24: Average CVSS score for top 10 weaknesses**

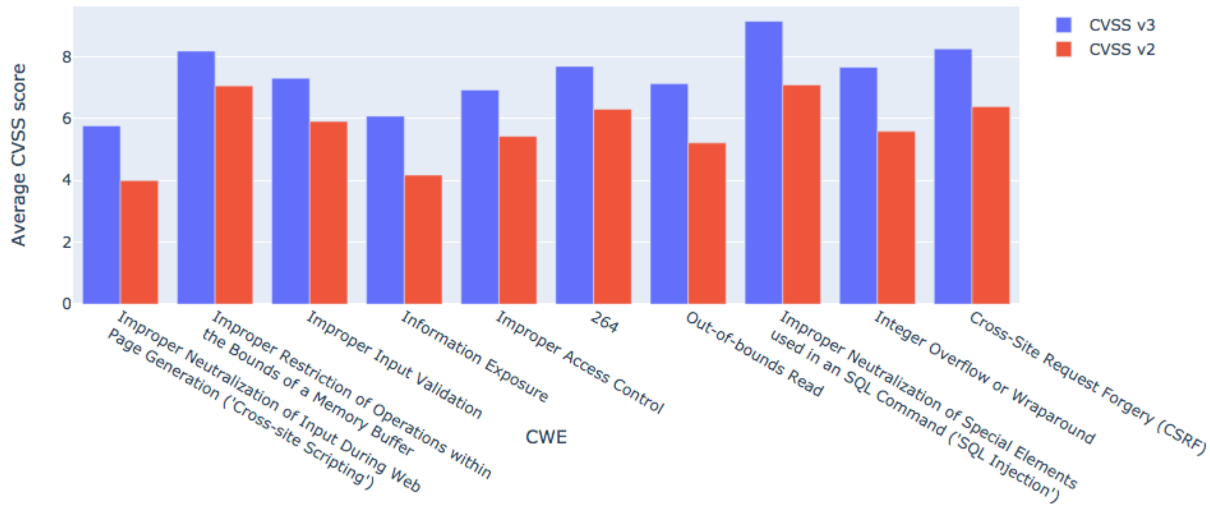
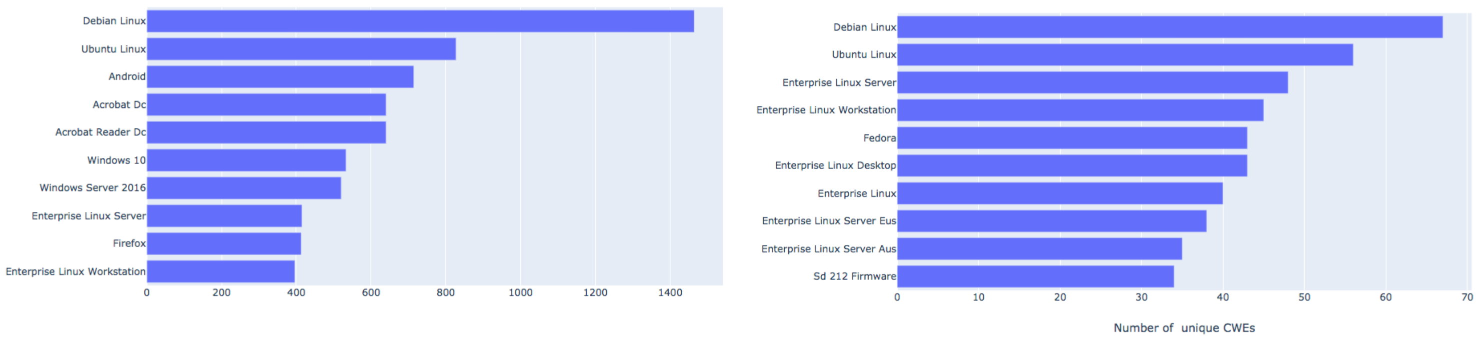


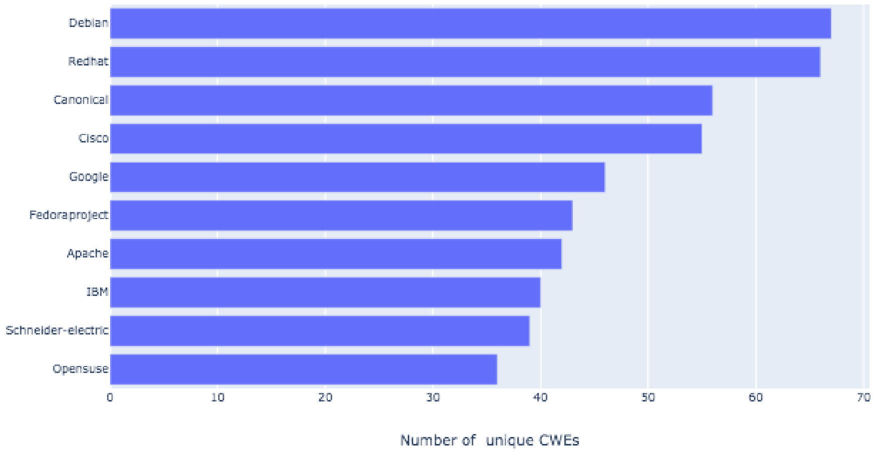

Figure 25 shows weaknesses per product, both in total and unique. Interestingly, Operating Systems dominate both lists, with Open Source OSs leading the charts.

**Figure 25:** CWEs per product



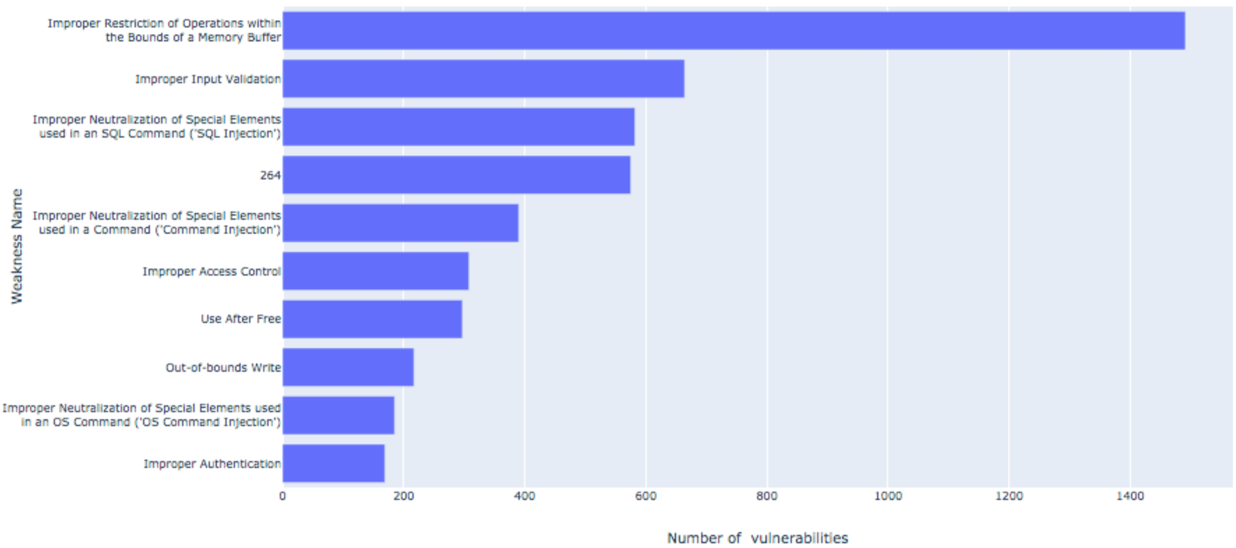
The corresponding chart with the vendors having the highest variety of vulnerabilities is shown in Figure 26. Debian, Redhat, Canonical and Cisco have over 55 distinct vulnerabilities.

**Figure 26:** Top 10 vendors with unique CWEs




Injections and improper input validation dominate the top 10 CWEs with a high CVSS score (Figure 27). It is reminded that ID 264 refers to a category rather than a specific weakness, namely “Permissions, Privileges and Access Controls”.

**Figure 27:** Top 10 CWEs with high CVSS score




## 3.7 SOFTWARE CATEGORIES

The following results refer to the 12.26% of the dataset, to vulnerabilities accompanied with a software category label, according to Vuldb. Although this subset failed the comparison test (that is, the distribution of vulnerabilities with description is different from the distribution of those lacking a description), it can be seen from Figure 28 that the two distributions are macroscopically similar and as such the subsequent analysis can be generalised for the whole vulnerabilities dataset.

**Figure 28:** Comparison of vulnerabilities with and without software category description

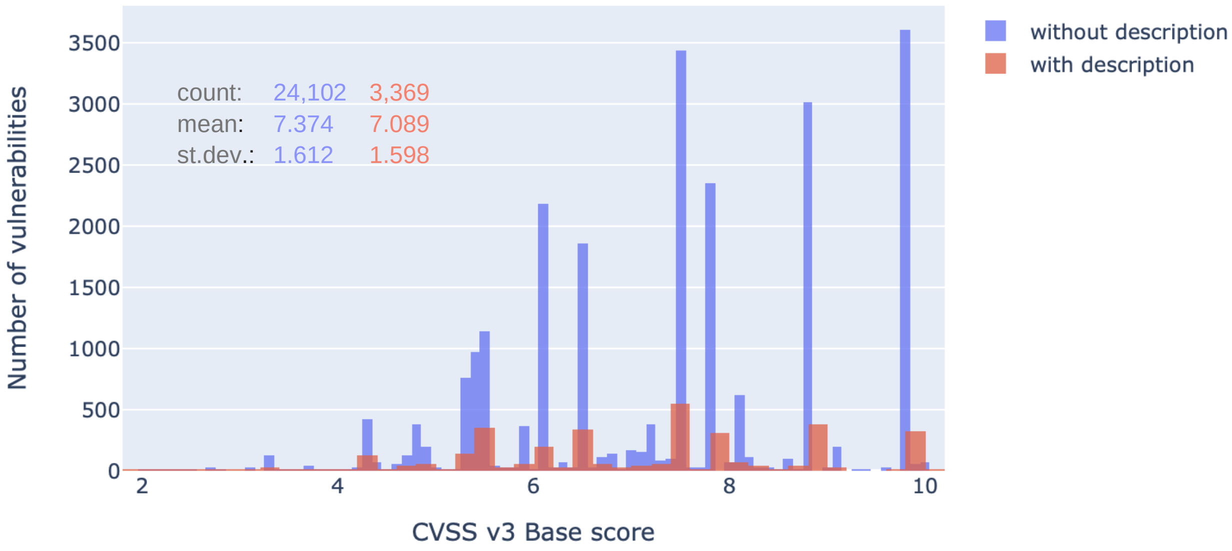
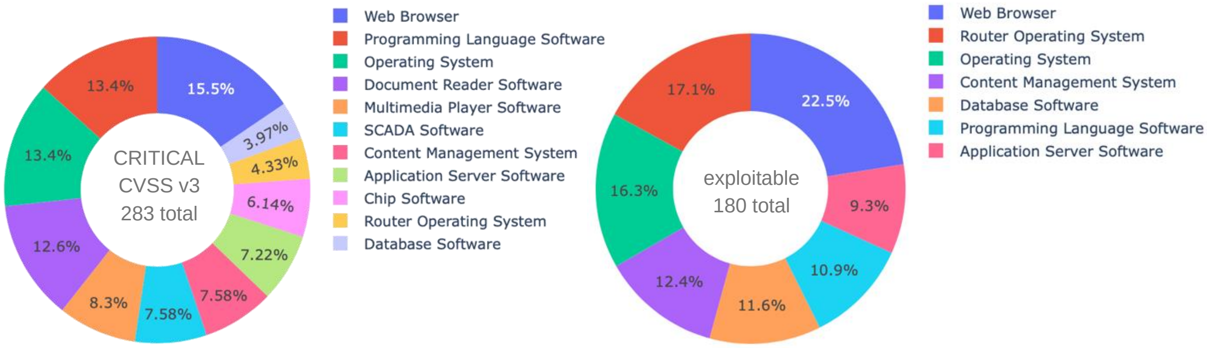



Figure 29 summarises the share of the categories that have more than 10 critical vulnerabilities and those that have at least one exploit (of any severity level). In both cases the Web browser category is on the top; it should be noted though that there are no exploitable vulnerabilities of critical severity (CVSS v3 score greater or equal to 9). The total number of software categories is 54.

**Figure 29:** Distribution of application types for more than 10 vulnerabilities and at least one exploit (of any severity)




**21** software categories having more than **50** vulnerabilities are ranked over their mean CVSS v3 base score in Figure 36. The ranking was in accordance to ANOVA test, showing the statistically significant differences between the means. Although web browsers have the highest number of vulnerabilities as shown in the previous figure, log management, multimedia player and SCADA software have a higher average severity (base) score.

**Figure 30:** Top categories with high CVSS v3 base score (having more than 50 vulnerabilities).

| Software category (mean CVSS v3 base score) | |
|---|---|
| **1.** Log Management Software (8.78) | = Router Operating System (7.08) |
| = Multimedia Player Software (8.34) | **3**. Operating System (6.91) |
| = SCADA Software (8.30) | = Firewall Software (6.88) |
| **2**. Chip Software (7.74) | = Content Management System (6.88) |
| = Document Reader Software (7.73) | = Application Server Software (6.87) |
| = Office Suite Software (7.44) | = Hosting Control Software (6.61) |
| = Web Browser (7.39) | = Programming Tool Software (6.51) |
| = Image Processing Software (7.25) | = Database Software (6.47) |
| = Packet Analyzer Software (7.21) | **4**. Groupware Software (6.26) |
| = Programming Language Software (7.21) | = Enterprise Resource Planning Software (6.12) |
| = Virtualization Software (7.16) | |

### 3.7.1 Software categories and their weaknesses

Figure 31 tessellates the most popular software categories and weaknesses pairs. The highest number of occurrences of weakness is **Improper Restriction of Operations within the Bounds of a Memory Buffer** which is found on **165** vulnerabilities affecting **Web Browsers.**

**Figure 31:** Top software categories – weaknesses pairs.

| | Improper Access Control | Improper Input Validation | 264 (Permissions, Privileges, Access Control) | Cross-site Scripting | Information Exposure | Use After Free | Out-of-bounds Read | Improper Restriction of Operations within the Bounds of a Memory Buffer |
|---|---|---|---|---|---|---|---|---|
| Virtualization Software | X | X | X | X | | X | X | X |
| Programming Language Software | X | X | X | X | X | X | X | X |
| Enterprise Resource Planning Software | X | | | X | X | | X | X |
| Document Reader Software | | | | | X | X | X | X |
| Operating System | X | X | X | | X | X | X | X |
| Web Browser | X | X | X | X | X | X | X | X |
| Database Software | X | X | X | | X | | X | X |

**CWE 119: Improper Restriction of Operations within the Bounds of a Memory Buffer, is a common weakness in Web Browser software**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Firewall Software | X | X | X | X | X | | | X |
| Router Operating System | X | X | X | X | X | | | X |
| Application Server Software | X | X | X | X | X | | | |
| Content Management System | X | X | X | X | X | | | |
| Programming Tool Software | X | X | | | X | X | X | X |
| Supply Chain Management Software | X | | | | X | | | |

### 3.7.2 Software categories with the highest number of vulnerabilities

Figure 32 shows the top 10 software categories with the most vulnerabilities. Out of these, the top 3 are further examined to establish if there are statistical differences in their means. Following the Mann-Whitney test, Web Browsers (mean base score: 7.39) have a significantly different (higher) CVSS v3 base score than the two runner ups, Operating Systems (mean score: 6.9) and Content Management Systems (mean score: 6.87).

**Figure 32:** Top 10 categories (highest number of vulnerabilities)

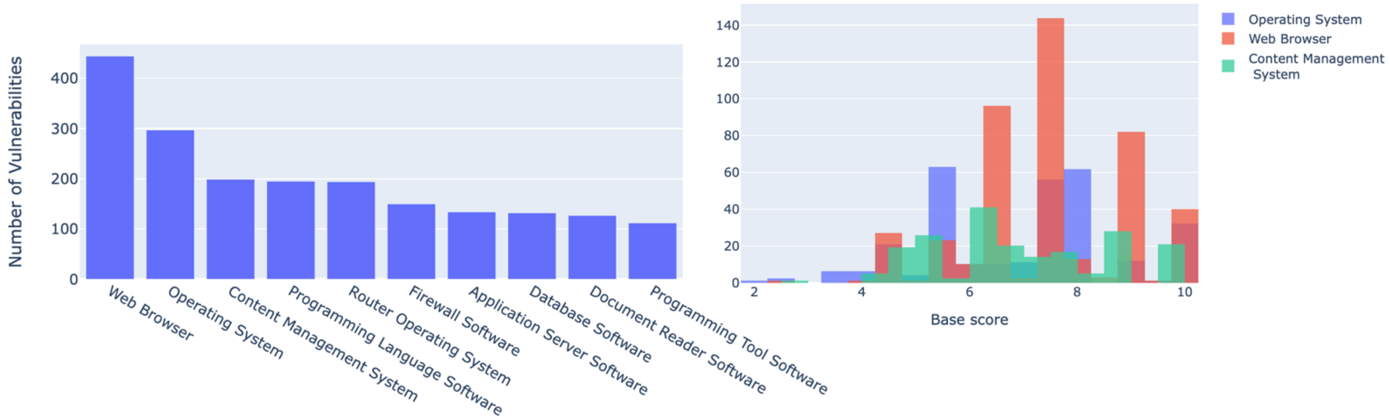



Figure 33 shows the top 20 software categories having most vulnerabilities, against the ATT&CK tactics. Note that as many points had a low number of observations, the maximum values are displayed.

**Figure 33:** Top 20 software categories against ATT&CK tactics (max base score values)

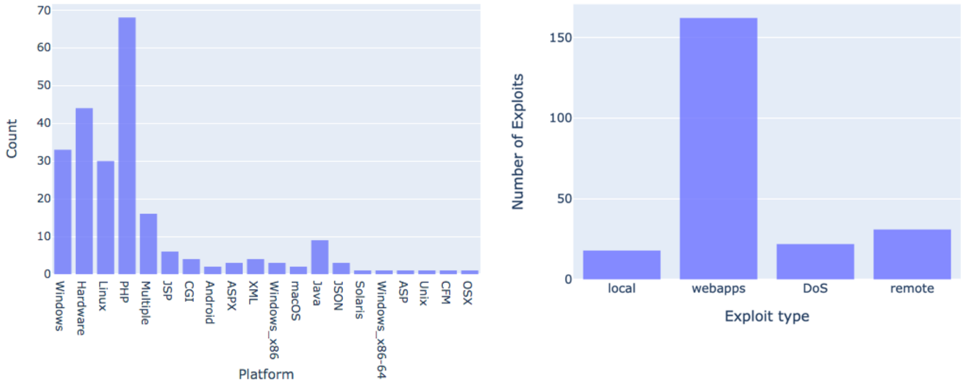
| | Collection | Command and Control | Credential Access | Defense Evasion | Discovery | Execution | Lateral Movement | Persistence | Privilege Escalation |
|---|---|---|---|---|---|---|---|---|---|
| Web Browser | 4.3 | 4.3 | 6.5 | 10.0 | 7.5 | 8.8 | 8.8 | 8.8 | 8.8 |
| Virtualization Software | 9.9 | 9.9 | 9.8 | 9.9 | 0.0 | 8.8 | 8.8 | 9.9 | 9.9 |
| SCADA Software | 9.8 | 9.8 | 0.0 | 9.8 | 0.0 | 7.8 | 7.8 | 9.8 | 9.8 |
| Router Operating System | 10.0 | 10.0 | 7.8 | 10.0 | 7.8 | 8.1 | 8.1 | 10.0 | 10.0 |
| Programming Tool Software | 0.0 | 0.0 | 0.0 | 8.2 | 7.2 | 8.2 | 8.2 | 8.2 | 8.2 |
| Programming Language Software | 9.8 | 9.8 | 0.0 | 9.8 | 7.5 | 9.8 | 9.8 | 9.8 | 9.8 |
| Packet Analyzer Software | 0.0 | 0.0 | 0.0 | 7.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Operating System | 9.1 | 9.1 | 0.0 | 10.0 | 7.5 | 9.8 | 9.8 | 9.8 | 9.8 |
| Office Suite Software | 0.0 | 0.0 | 0.0 | 9.1 | 6.5 | 6.5 | 6.5 | 6.5 | 6.5 |
| Log Management Software | 9.8 | 9.8 | 0.0 | 9.8 | 5.3 | 0.0 | 0.0 | 9.8 | 9.8 |
| Image Processing Software | 0.0 | 0.0 | 0.0 | 7.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Hosting Control Software | 0.0 | 0.0 | 0.0 | 8.8 | 8.8 | 8.8 | 8.8 | 8.8 | 8.8 |
| Groupware Software | 0.0 | 0.0 | 0.0 | 8.8 | 6.5 | 7.8 | 7.8 | 7.8 | 7.8 |
| Firewall Software | 0.0 | 0.0 | 8.8 | 9.8 | 8.8 | 8.8 | 8.8 | 8.8 | 8.8 |
| Enterprise Resource Planning Software | 0.0 | 0.0 | 0.0 | 8.8 | 5.9 | 8.8 | 8.8 | 8.8 | 8.8 |
| Document Reader Software | 0.0 | 0.0 | 0.0 | 5.3 | 5.3 | 0.0 | 0.0 | 0.0 | 0.0 |
| Database Software | 0.0 | 0.0 | 0.0 | 9.9 | 6.5 | 9.9 | 9.9 | 9.9 | 9.9 |
| Content Management System | 0.0 | 0.0 | 0.0 | 9.8 | 8.1 | 9.8 | 9.8 | 9.8 | 9.8 |
| Chip Software | 7.8 | 7.8 | 7.8 | 9.8 | 9.8 | 7.8 | 7.8 | 7.8 | 7.8 |
| Application Server Software | 6.3 | 6.3 | 8.8 | 9.8 | 8.8 | 9.8 | 9.8 | 9.8 | 9.8 |

## 3.8 WEB

### 3.8.1 All products

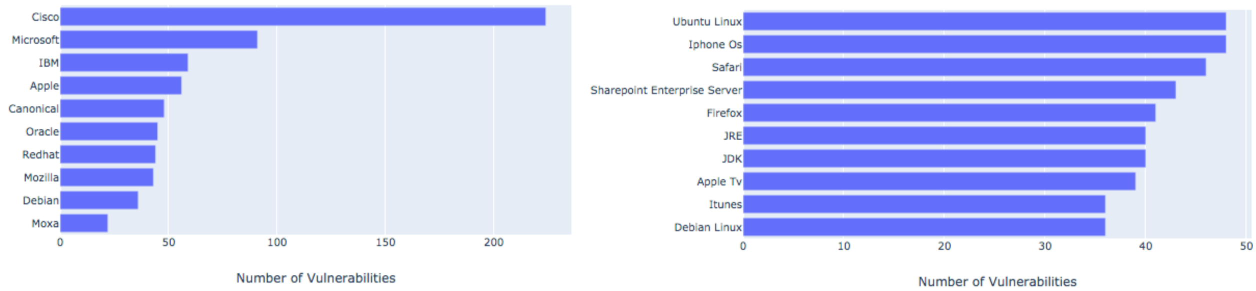
The following results refer to all products that relate somehow to web applications or services, relating to the 7.6% of the data. The selection process involved the inclusion of those vulnerabilities containing the keyword “web” in their description. The corresponding exploits are shown in Figure 34. Webapps type of exploit and PHP are the most popular focus of exploits.

**Figure 34:** Target platforms and exploit types




With regards to vulnerabilities on web products Cisco displays by far the most vulnerabilities (over 250), whereas the follow up vendor, Microsoft, has less than 100. The variation of vulnerabilities within the top 10 most vulnerable web products is limited, approximately between 37 to 57 (Figure 35).

**Figure 35:** Vendors and products with the most vulnerable web products




### 3.8.2 Windows applications

The CVSS scores and differences between CVSS v2 and v3 in Windows applications follows the overall distributions as presented earlier, with v3 showing a higher mean base score. Unsurprisingly, Microsoft as a vendor and its products dominate the top 10 vulnerabilities list (Figure 36).

**Figure 36:** Vendors and products with the most vulnerable Windows applications

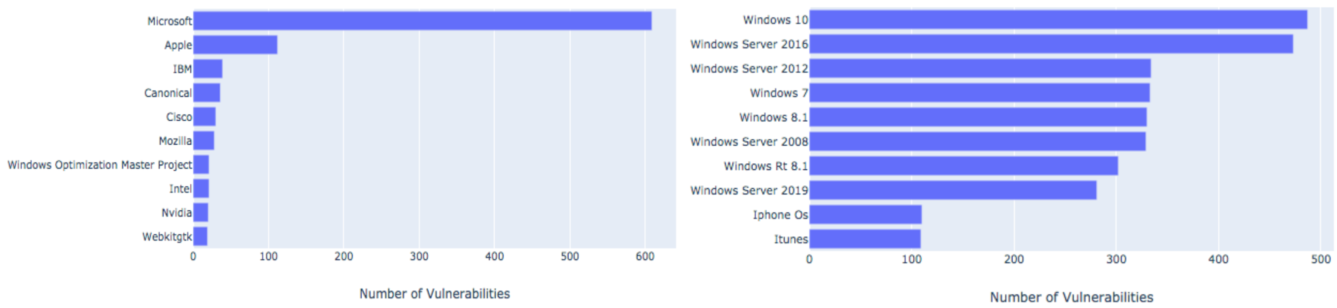
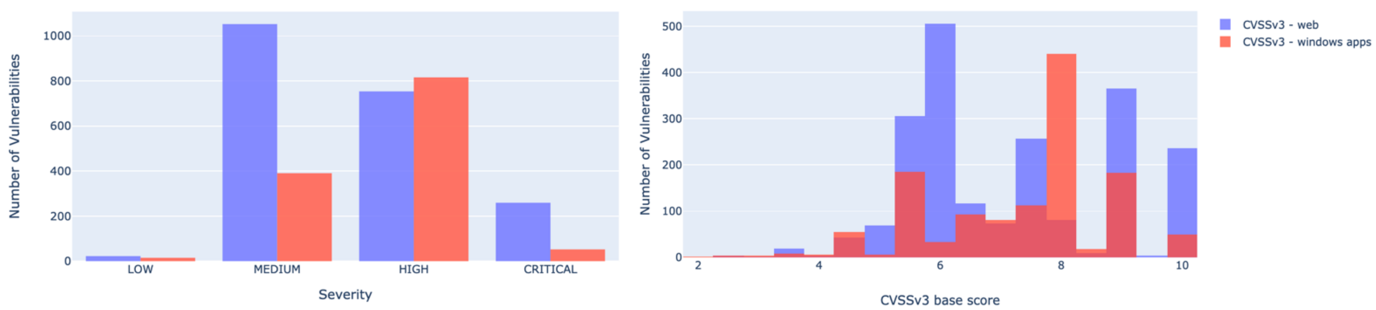



Figure 37 shows the severity and base scores of web and windows applications. As the CVSS base score distributions are clearly distant from a normal distribution, the Mann Whitney non parametric test was performed in order to check if these two sets differ. Indeed, the probability of the test was virtually equal to zero. Hence, we accept the hypothesis of the two samples having different means and in this case the Windows applications have a higher mean CVSS base score than web applications (7.25 as opposed to 7.1).

**Figure 37:** Severity ratings for web and windows applications




## 3.9 OPEN SOURCE SOFTWARE

The subset containing the vulnerabilities of the open source was extracted from the main dataset manually, after inspecting the rows and isolating the most popular open source projects. This led to a dataset of 3,221 vulnerabilities which accounts for the 11% of the whole dataset. The CVSS score distribution and differences between version 2 and version 3 of the sample followed the complete dataset (Figure 38). Once more, the discrepancy between the two versions is considerable.

**Figure 38:** CVSS v2 vs. v3 in open source software

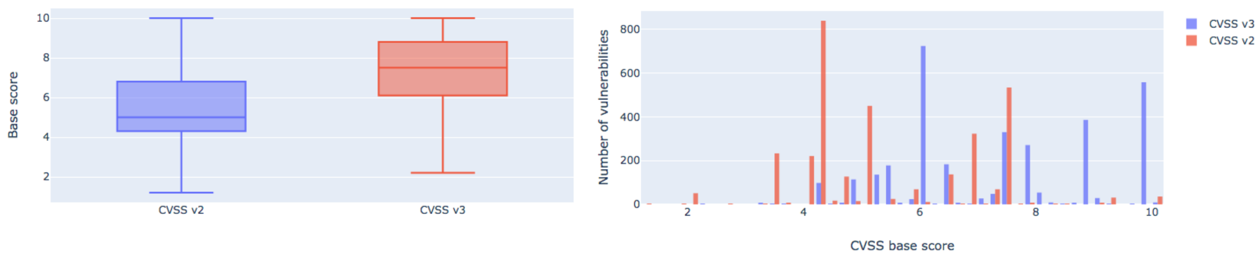
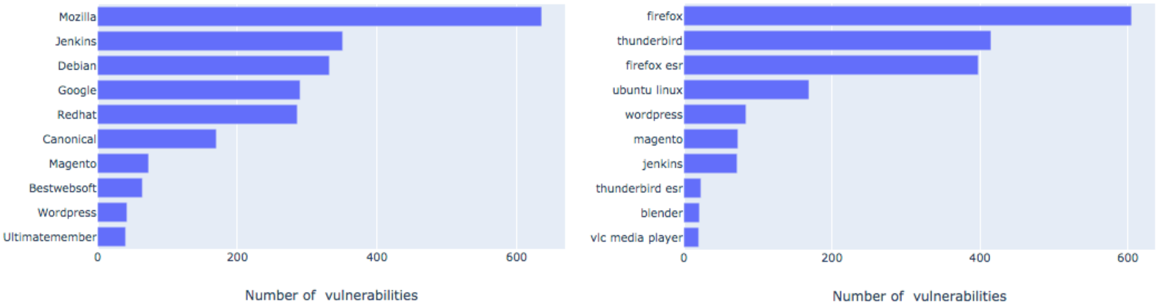


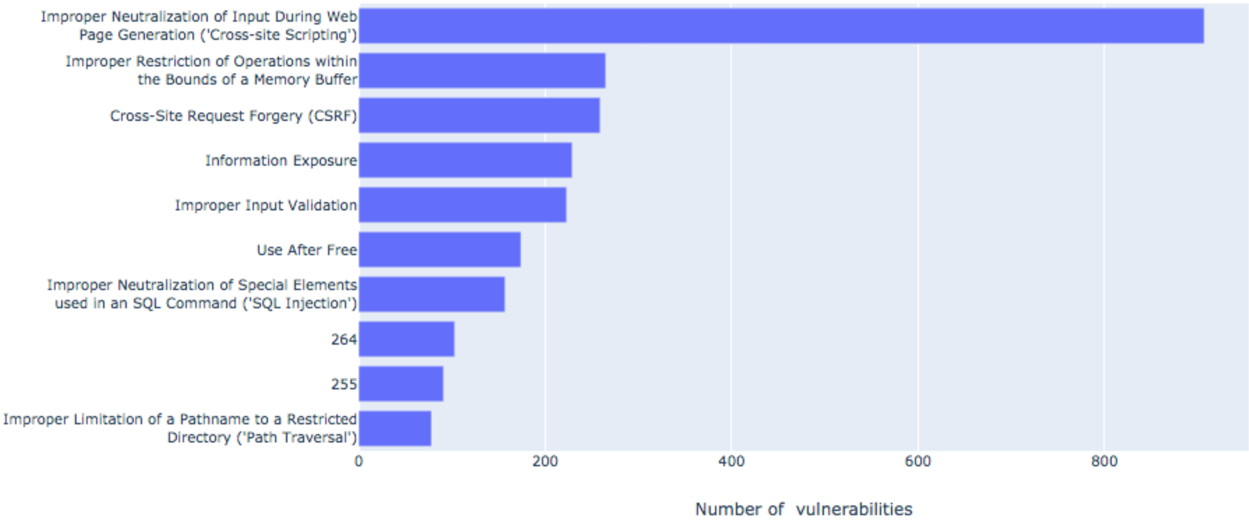

Figure 39 shows the vendors and the projects of the open source software where Mozilla and Firefox are in the lead.

**Figure 39:** Distribution of vulnerabilities on vendors and projects of Open Source software




In Figure 40 the top weaknesses in open source software are presented. CWE 255 refers to a weakness category, **Credentials Management**.

**Figure 40:** CWEs in open source software

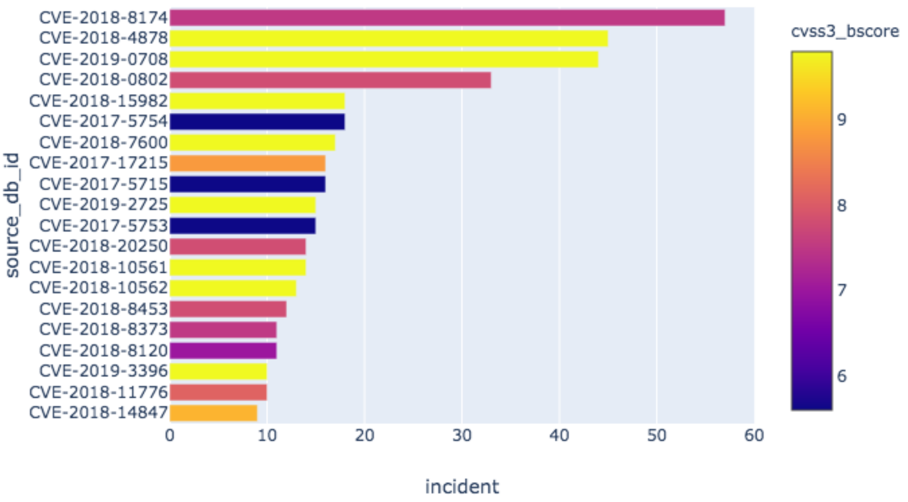



## 3.10 POPULARITY

Popularity refers to the number of times a particular vulnerability is referenced by articles, posts, whitepapers, etc. In the following charts the data presented refer to the number of appearances (or "incidents") as these have been summarised and collected by ThreatConnect.

Figure 41 summarises the 20 most popular vulnerabilities. The colour code maps to the CVSS v3 score. From a visual inspection there is no correlation between the popularity of a vulnerability and the CVSS score (to be precise the correlation is very low, 0.084). Nevertheless, the most "popular" CVEs with the highest score (CVE-2018-4878 and CVE-2019-0708) refer to Redhat, Microsoft, Adobe and more particularly their Operating Systems software and flash player, in the case of Adobe. CVE-2018-8174 which is the vulnerability with the highest number of appearances refers to Microsoft's Windows Server products.

**Figure 41:** "Celebrity" vulnerabilities

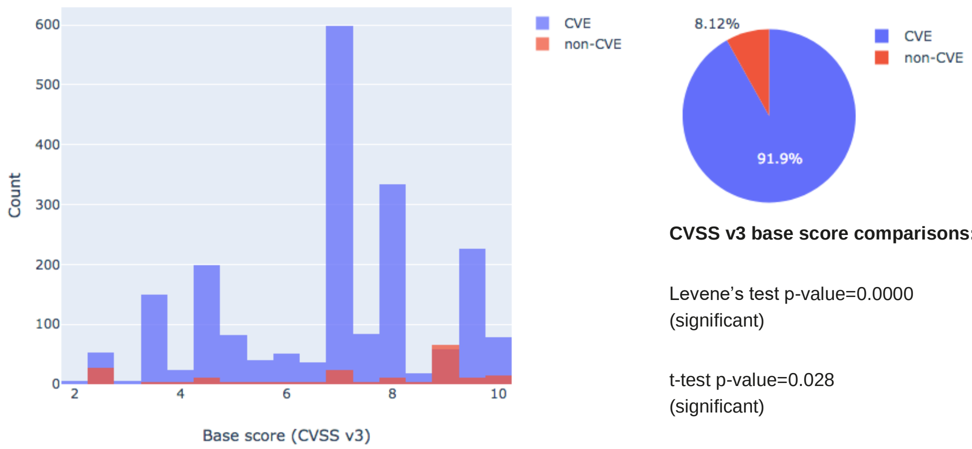



## 3.11 NON-CVE ANALYSIS

A considerable amount of activity surrounds vulnerabilities that do not enter the CVE ecosystem or if they do so, it happens at a very late stage. The main dataset for example, contained vulnerabilities that received a CVE ID even 5 years after an exploit was published. In this section the Zero Day Initiative (ZDI) data were analysed. This dataset was selected as it contained structured information and an adequate number of non-CVE vulnerabilities that were scored based on the CVSS v3 system. It should also be highlighted that the data that considered to be within the scope/time range of this study are those that ZDI used an ID of the form ZDI-18-xxx. From the findings shown in Figure 42, the non-CVE vulnerabilities have a statistically significant different (higher) CVSS v3 score mean from those with CVE.

In terms of percentage between the CVE and non-CVE vulnerabilities, for the ZDI data the non-CVE vulnerabilities account for slightly over 8%, whereas VulDB reports this figure to be up to approximately 30%. This is because ZDI is primarily a subset of VulDB who gathers vulnerability information from a wider number of sources.

**Figure 42:** CVE vs non-CVE in ZDI data




**CVSS v3 base score comparisons:**

Levene's test p-value=0.0000
(significant)

t-test p-value=0.028
(significant)

## 3.12 EXPLOITS, TACTICS AND PATTERNS

A threat actor can only succeed in an attack if they manage to develop and deploy an exploit against an existing vulnerability. The existence of exploits and their characteristics (such as complexity, privileges required, and so forth) can significantly affect the level of risk. In this section the exploitation aspects of the vulnerability ecosystem are explored. According to the dataset, the lower bound of the percentage of vulnerabilities being exploited in the wild is **8.65%.**

Out of the recorded 2,371 exploitable vulnerabilities, **492** have a CRITICAL (version 3) severity score. The top 10 are presented in Figure 43. Out of these 492 vulnerabilities, **4** had **10** or more published exploits (CVE-2015-2003, CVE-2014-2048, CVE-2015-2000, CVE-2015-2001), all having CVSS score 9.8.

There is a **0.46** (positive) correlation between the number of exploits and the popularity (number of incidents) of a vulnerability. CVE-2018-4878 and CVE-2019-0708, both having a v3 score equal to 9.8, have a staggering number of 45 and 44 incidents published by ThreatConnect.

**LOWER BOUND OF NUMBER OF EXPLOITS**

At least 8.65% of the vulnerabilities can be practically exploited.

This accounts to 2,377 exploitable vulnerabilities for 2018 (and half of 2019)

**Figure 43:** Top 10 most critical exploitable vulnerabilities

| CVE-ID | CNA | CVSS v3 base score | Platform | Vendor | exploits |
|---|---|---|---|---|---|
| **CVE-2018-10718** | MITRE Corporation | 10 | Windows | - | 2 |
| **CVE-2017-12542** | Hewlett Packard Enterprise (HPE) | 10 | Multiple | HP | 1 |
| **CVE-2019-11510** | MITRE Corporation | 10 | Multiple | Pulsesecure | 1 |
| **CVE-2018-0101** | Cisco Systems, Inc. | 10 | Hardware | Cisco | 2 |
| **CVE-2019-0007** | Juniper Networks, Inc. | 10 | Windows | - | 2 |
| **CVE-2018-3110** | Oracle | 9.9 | Linux | Oracle | 2 |
| **CVE-2018-3856** | Talos | 9.9 | Linux | Samsung | 1 |
| **CVE-2018-1712** | IBM Corporation | 9.9 | Linux | IBM | 2 |
| **CVE-2018-3904** | Talos | 9.9 | Linux | Samsung | 1 |
| **CVE-2017-16339** | Talos | 9.9 | Linux | Insteon | 1 |

In total there are 11 platforms that have more than 20 exploitable vulnerabilities (Figure 44). PHP has the highest number of vulnerabilities, followed by Linux and Windows.

**Figure 44:** Platforms with more than 20 exploitable vulnerabilities

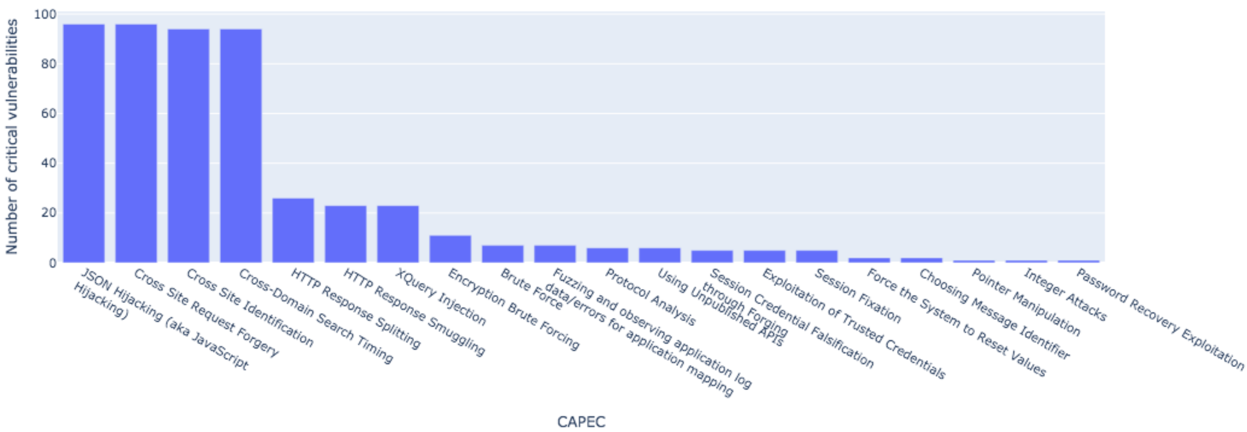
| Platform | Number of vulnerabilities | CVSS v3 base score | CVSS v2 base score |
|---|---|---|---|
| **PHP** | 769 | 7.548114 | 5.897529 |
| **Linux** | 411 | 7.310706 | 5.720925 |
| **Windows** | 393 | 7.417048 | 5.947583 |
| **Hardware** | 240 | 7.634583 | 5.964167 |

| | | | |
|---|---|---|---|
| **Multiple** | 221 | 7.49819 | 5.819457 |
| **Java** | 61 | 7.206557 | 5.777049 |
| **JSP** | 48 | 6.9625 | 5.760417 |
| **XML** | 28 | 6.889286 | 4.853571 |
| **ASPX** | 27 | 7.655556 | 5.67037 |
| **JSON** | 22 | 7.259091 | 5.431818 |
| **macOS** | 21 | 7.095238 | 5.990476 |

### 3.12.1 CAPECs

Figure 45 summarises the most frequent CAPECs for critical vulnerabilities (score>9) with an available exploit. The 4 most frequent CAPECs exploiting over 90 vulnerabilities refer to web and client-side exploits (note that Cross Site Identification is essentially a form of Cross Site Request Forgery that does not require user actions).

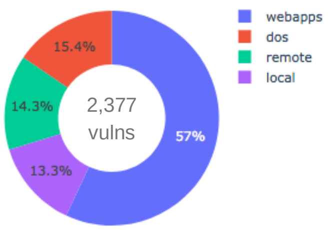
**Figure 45:** Top 20 most frequent CAPECs for exploitable critical vulnerabilities




**Remote, web based attacks targeting web clients are the most common attack pattern**

An analysis of variance carried out across all 2,377 exploitable vulnerabilities, indicated that the CVSS v3 score for remote attacks is the highest, followed by local and webapps (which have statistically the same score), followed by DoS (Figure 46). It is noteworthy that similar results were obtained with CVSS v2 base score, but with significantly lower means (ranging from 5.44 to 6.5).

**Figure 46:** ANOVA results for exploit type CVSS v3 base score differences




```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1  group2 meandiff p-adj   lower   upper  reject
-----------------------------------------------------
   dos   local   0.2543 0.1955 -0.0757  0.5842  False
   dos  remote    0.762  0.001  0.4383  1.0856   True
   dos webapps   0.3995  0.001  0.1465  0.6525   True
 local  remote   0.5077  0.001  0.1717  0.8437   True
 local webapps   0.1452 0.5051 -0.1234  0.4139  False
remote webapps  -0.3625  0.002 -0.6233 -0.1017   True
-----------------------------------------------------
```

| exp_type | N | Mean |
|---|---|---|
| dos | 366 | 7.074317 |
| local | 315 | 7.328571 |
| remote | 339 | 7.836283 |
| webapps | 1351 | 7.473797 |

### 3.12.2 ATT&CK techniques

There were 132 identified exploitable critical CVEs with ATT&CK technique identifiers. Figure 47 shows the distribution of techniques against the matching vulnerabilities. **26** vulnerabilities were shown to be open to as many as **19** ATT&CK techniques. Out of these, **HISTCONTRO**L was the most popular technique, affecting **49** critical vulnerabilities, followed by **Obfuscated Files or Information** which is applicable to **35** vulnerabilities. The arrangement of critical vulnerabilities on the ATT&CK framework is shown in Figure 48. **Persistence and Discovery** are the most popular tactics, followed by **Defense Evasion** and **Privilege Escalation**.

**Figure 47:** Number of techniques affecting critical exploitable vulnerabilities

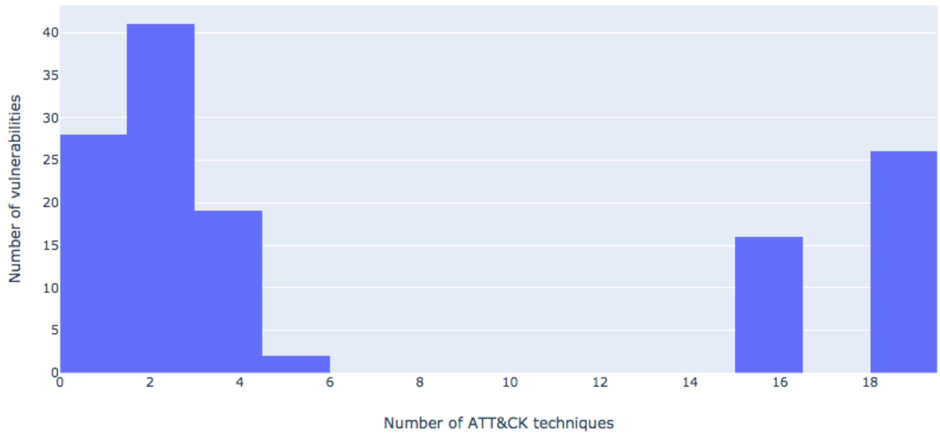



**Figure 48:** Critical exploitable vulnerabilities positioned in the ATT&CK framework – The number in parentheses represents the number of vulnerabilities affected by the technique.

| Execution | Persistence | Defense Evasion | Privilege Escalation | Credential Access | Discovery | Lateral Movement | Collection | Command & Control |
|---|---|---|---|---|---|---|---|---|
| T1152. Launchctl (23) | T1062. Hypervisor (23) | T1148. HISTCONTROL (49) | T1160. Launch Daemon (23) | T1081. Credentials in Files (8) | T1124. System Time Discovery (13) | T1037. Logon Scripts (23) | T1185. Man in the Browser (14) | T1090. Connection Proxy (14) |
| | T1031. Modify Existing Service (23) | T1027. Obfuscated Files or Information (35) | T1050. New Service (23) | T1214. Credentials in Registry (1) | T1007. System Service Discovery (13) | T1051. Shared Webroot (23) | | |
| | T1162. Login Item (23) | T1152. Launchctl (23) | T1058. Service Registry Permissions Weakness (23) | | T1087. Account Discovery (13) | T1080. Taint Shared Content (23) | | |
| | T11.52 Launchctl (23) | T1130. Install Root Certificate (23) | T1015 Accessibility Features (23) | | T1033. System Owner/User Discovery (13) | | | |
| | T1160. Launch Daemon (23) | T1014. Rootkit (23) | T1134. Access Token Manipulation (14) | | T1049. System Network Connections Discovery (13) | | | |
| | T1050. New Service (23) | T1089. Disabling Security Tools (23) | T1100. Web Shell (14) | | T1016. System Network Configuration Discovery (13) | | | |

| Execution | Persistence | Defense Evasion | Privilege Escalation | Credential Access | Discovery | Lateral Movement | Collection | Command & Control |
|---|---|---|---|---|---|---|---|---|
| | T1159. Launch Agent (23) | T1090. Connection Proxy (14) | T1044. File System Permissions Weakness (12) | | T1046. Network Service Scanning (13) | | | |
| | T1215. Kernel Modules and Extensions (23) | T1134. Access Token Manipulation (14) | | | T1018. Remote System Discovery (13) | | | |
| | T1156. .bash_profile and .bashrc (23) | T1126. Network Share Connection Removal (13) | | | T1424. Process Discovery (13) | | | |
| | T1037. Logon Scripts (23) | | | | T1069. Permission Groups Discovery (13) | | | |
| | T1058. Service Registry Permissions Weakness (23) | | | | T1120. Peripheral Device Discovery (13) | | | |
| | T1042. Change Default File Association (23) | | | | T1135. Network Share Discovery (13) | | | |
| | T1067. Bootkit (23) | | | | T1082. System Information Discovery (13) | | | |
| | T1015 Accessibility Features (23) | | | | T1083. File and Directory Discovery (1) | | | |
| | T1100. Web Shell (14) | | | | T1012. Query Registry (1) | | | |
| | T1044. File System Permissions Weakness (12) | | | | | | | |

## 3.13 ECONOMIC ASPECTS

Economic analysis of vulnerabilities comes with a different set of challenges. The technical description of a vulnerability is expected to be fairly objective; CVE-2018-1163 for example refers to Quest's Netvault backup product and is a critical vulnerability as it fully affects all three impact categories (confidentiality, integrity and availability), and since this can be performed with a low complexity, and no privileges required it yields a v2 base score of 10 and a v3 base score of 9.8.

However, when studying the economic aspects of vulnerabilities, there is higher inherent subjectivity across two directions. First, there is the more esoteric direction of the impact of a particular vulnerability to an organisation. This impact feeds into the risk assessment process. The CVSS scoring system caters for refinements of the base score by introducing temporal and environmental metric groups. The latter in particular is an attempt to take into consideration the particular user's environment in order to further contextualise the vulnerability. Naturally, this metric is rarely published in a publicly available vulnerability database. As such, this aspect of economic analysis was excluded from this study.

Second, the economic aspects of vulnerabilities can be indirectly assessed through the prices of the associated exploits. A number of sources and initiatives were analysed and it is conjectured that a considerable amount of unsubstantiated evidence exists in the wild – or to be more precise, it was not always possible to assess the validity and correctness of the claims relating to the publicly available economic data. The analysis that follows considers the price estimates from Vuldb who use a proprietary algorithm as well as Zerodium's bug bounty programme. It can be evident from the findings below that there can be great ranges and discrepancies in the price of an exploit, which could be compared to the complex dynamics of a market led by opportunity costs.

### 3.13.1 Bug bounty

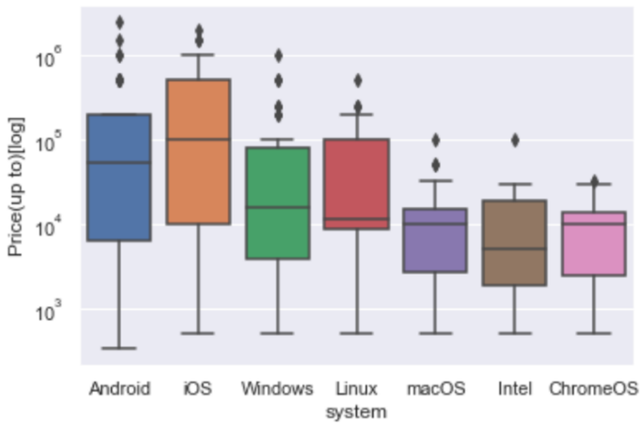
Figure 49 presents the maximum awards for exploits for the major systems as advertised by Zerodium. The mobile operating systems are in the lead with Android having the higher maximum (outlier) value set to 2.5 million dollars. Although Google has paid since 2010 over $15m in total to researchers to date[20], the highest amount recorded in their bug bounty programme is for bugs discovered for the Titan M chip used in Pixel smartphones, which may reach a maximum of $1.5m[21].

In addition to the differentiation of prices based on the different systems, the bug bounty scheme notes high payouts for exploits delivering Local Privilege Escalation (LPE) and if done through Remote Code Execution. The highest payout which is for Android systems refers to a zero-click exploit, that is a 0day exploit that takes over an Android phone with no interaction from a user.

**HIGHEST BUG BOUNTY PAYOUTS**

Zero-click, zero-day exploits for Android OS are the highest paid exploits according to Zerodium's bug bounty programme

**Figure 49:** Zerodium's bug bounty upper limit prices for the major systems (in $)




| System | Observations | Mean | Max |
|---|---|---|---|
| **Android** | 50 | 267K | 2.5M |
| **iOS** | 49 | 226K | 2M |
| **Windows** | 42 | 93K | 1M |
| **Linux** | 48 | 68K | 0.5M |
| **Intel** | 12 | 17K | 100K |
| **macOS** | 26 | 15K | 100K |
| **ChromeOS** | 25 | 9K | 32.7K |

[20] https://www.techradar.com/uk/news/google-ups-play-store-bug-bounties
[21] https://www.techradar.com/uk/news/google-offers-million-dollar-bug-bounty-reward

### 3.13.2 Exploit price estimation

The analysis was performed on the publicly available subset from VulnDB. Following the ANOVA results on the CVSS base score over the four price categories (0-day low, 0-day upper, today low, today upper) it seemed that, as expected, the price is dependent on the score, with the 0-day values showing more significant correlations. The price is distinctively highest in particular for vulnerabilities of critical severity (Figure 50).

**Figure 50:** VulDB's exploit price estimates

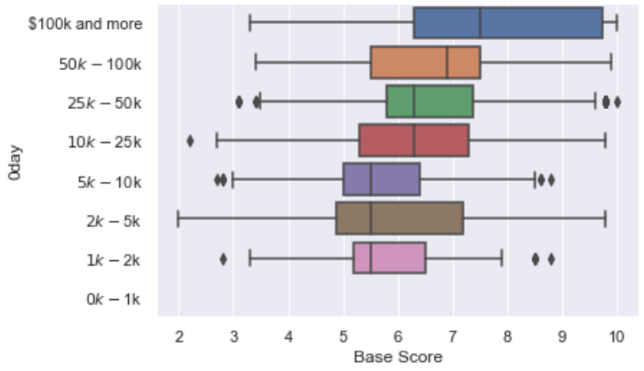



Figure 51 shows the exploit prices over the base score and in relation to the existence of a remedy (official fix, workaround, not defined). Although there are no noteworthy differences and patterns internally in the graphs, "Today's price" values seem to be close to a "right shift" in the price band from the 0-day price values.

**Figure 51:** Exploit prices based on existence of remedies (fixes) and state (0-day vs. current/today's price)

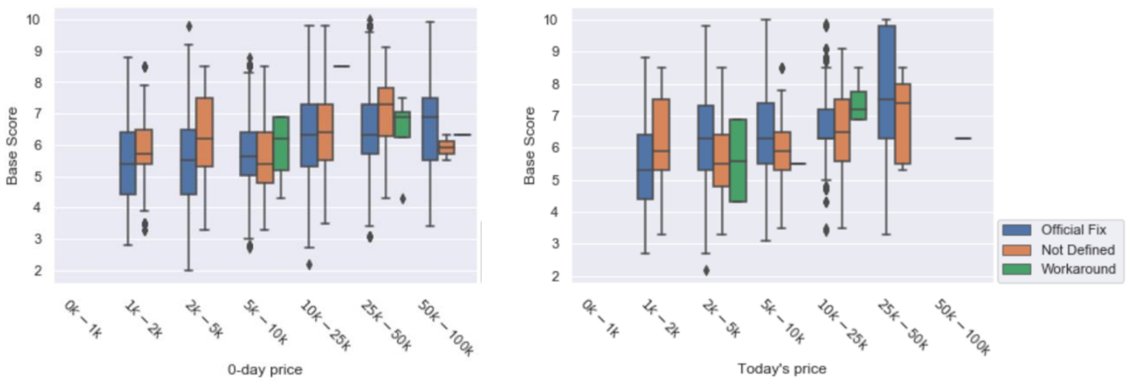
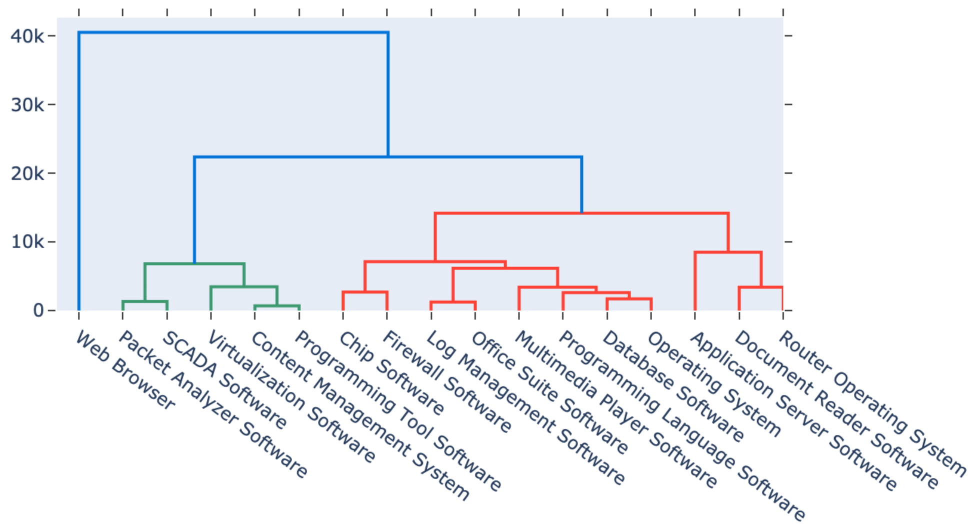



Figure 52 shows the dendrogram following hierarchical clustering on the prices; the upper price data for 0-day was excluded as it included an open upper bound value and would therefore yield less realistic results. Moreover, only the software categories with more than 40 observations were included. According to the clustering output, Web Browsers are clearly distant from all other categories. Setting a threshold at $20k, there are three clusters in total, as seen by the different colouring.

**Figure 52:** Hierarchical clustering of software categories on Today's prices and 0-day low prices (for categories with >40 recorded vulnerabilities)




## 3.14 CONCLUDING REMARKS

The analyses presented in the previous sections are an indication of the potential to generate intelligence, make informed decisions, and perform risk assessment exercises on software vulnerabilities. The systematic efforts of the computer security community to create a taxonomy, and develop databases populated by structured vulnerability descriptors, paves the way to a deeper exploration of the vulnerability ecosystem.

The results and findings show potentially revealing relationships and crucial information on how vulnerabilities are spawned from weaknesses. These fuel the development of exploits and the formation of attack patterns, techniques, and tactics. At the same time, when assessing the quality and accuracy of the collected data, we caution the reader to be aware of two main caveats. Firstly, the frameworks and metrics developed to express and assess the vulnerabilities are in a journey of their own: they go through continuous transformation, revision, and development. This suggests that future, more improved and rigorously validated frameworks may not be "compatible" or in agreement with the current/preceding versions, affecting the interpretation of the outcomes. Moreover, the taxonomies, frameworks, and metrics have grown to become exceedingly elaborate and detailed; whereby the approximate sample size of 27k observed vulnerabilities over the study period did not cover the whole range of categories. Although this is not necessarily a drawback for instruments such as the ATT&CK framework, it may be an issue for the CVSS metric groups, as there are substantial differences between the two prevailing quantitative descriptor versions (CVSS version 2 and version 3.x). For instance, the two CVSS versions have different definitions for the exploitability metric with significantly different ranges (where the version 2 range is 0-10 and the version 3 range is 0-3.9). Therefore, it is critical to be clear on the version of the CVSS metrics employed when conducting any analyses, visualisations, or reports. In addition, a possible migration from one version to the other must be carefully planned and executed.

Secondly, the datasets found in both "authoritative" sources and in the wild show significant discrepancies which in some cases can cause a substantial (mis)classification of a vulnerability in terms of its severity and impact. Moreover, due to the nature of this problem domain, the vulnerability datasets are expected to be incomplete, however, we do not consider this to be a critical issue in contrast to the discrepancies caveat. We make this assumption based on the observation that fewer samples and data points mostly follow the statistical behaviour of the greater population, with the exception of non-CVE vulnerabilities that were found to differ from those registered under the CVE scheme.

When considering CVSS scores in particular, there are differences both in the definitions between the two scoring versions as mentioned above, as well as in the actual values themselves. The latter discrepancies may affect the reliability and trustworthiness of these quantitative schemes. For instance, although the overall impact metric formula is different between versions 2 and 3, with the latter version having the impact dependent on the *Scope* variable, the main impact factors used in the calculation (Confidentiality, Integrity, Availability) are not consistent within the same database, yielding alarmingly low correlations. As a result, when using vulnerability data, decision makers and risk assessors should either consider the worst-case scenario by accepting the higher impact value or, if resources permit, they should independently assess the impact of the underlying vulnerability. In any case, the risk assessment process should integrate outlier and discrepancy detection layers and functions into the underlying risk assessment toolbox.

This publicly available report is accompanied by the vulnerability dataset and source code. These have been made available as a collection of Jupyter notebooks written in Python, not only to promote transparency by empowering the independent validation of the findings contained in this report, but also to enable the cyber security community to conduct further investigations and analyses. An exhaustive exploration of the vulnerability ecosystem was prohibited by the richness of the dataset and the significant efforts in the recent literature to capture the various aspects in a streamlined and standardised form. Although the Jupyter notebooks contain a more detailed and diverse set of findings, the following ones were considered most significant and presented in the present report:

- There are significant differences between the two vulnerability measurement systems (CVSS v2 and CVSS v3), possibly attributed to the different wording of the categorical variables, fuelling subjective bias. In either case, the correlations of the three impact measures (Confidentiality, Integrity and Availability) were surprisingly low, with Integrity and Availability being less than 0.4.
- There are inconsistencies and discrepancies between the different sources. Although there is an authoritative database capturing vulnerability details, this does not imply that the information in that database is accurate.
- The developed taxonomies and standards used to describe the vulnerabilities are indeed rich and detailed, but only a subset of the categories were present in the 2018-2019 vulnerabilities dataset.
- There are statistically significant differences between the severity level of CVE (officially recorded) and non-CVE vulnerabilities (i.e. those that were not listed or included in the CVE databases), with the latter showing a higher score.
- The exploit publication date of CRITICAL vulnerabilities is attracted near the vulnerability publication date, with most exploits being published shortly before or after the vulnerability publication date.
- At least 8.65% of the vulnerabilities are exploitable. This number is expected to be higher due to zero-day exploits and the incompleteness of the datasets.
- Defence Evasion, Persistence, and Discovery are the preferred tactics for the exploits.
- Most exploits target web and client-side related vulnerabilities.
- The top 10 weaknesses account for almost two thirds (64%) of the vulnerabilities.

# 4. REFERENCES

Arbaugh, W., Fithen, W., McHugh, J.  Windows of Vulnerability: A Case Study Analysis. IEEE Computer, Vol 3, No. 12, December 2000.

ENISA, Good Practice Guide for Vulnerability Disclosure: From Challenges to recommendations, 2015. Available from: https://www.enisa.europa.eu/publications/vulnerability-disclosure

ENISA, Economics of Vulnerability Disclosure, 2018. Available from: https://www.enisa.europa.eu/publications/economics-of-vulnerability-disclosure

ENISA, Threat Landscape Report, 2019. Available from: https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2018

Hutchins, E., Cloppert, M. and Amin,R. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Bethesda, MD: Lockheed Martin Corporation, 2010. Available from: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf

Kritikos, K., Magoutis, K., Papoutsakis, M., Ioannidis, S., A survey on vulnerability assessment tools and databases for cloud-based web application. Array, Vol 3, No. 4, 2019., pp. 1-21.

# A ANNEX: LIST OF VULNERABILITY DATABASES

## A.1 FREE ACCESS DATABASES

https://exchange.xforce.ibmcloud.com/
https://www.securityfocus.com/vulnerabilities
https://nvd.nist.gov/
https://www.cvedetails.com/
https://vuldb.com/
https://www.exploit-db.com/
https://www.rapid7.com/db/
https://snyk.io/features/vulnerability-database/
https://www.kb.cert.org/vuls/
https://www.first.org/global/sigs/vrdx/vdb-catalog
https://help.veracode.com/reader/hHHR3gv0wYc2WbCclECf_A/lQYKhC8AvpIbz5_ULOCYMw
https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/
https://www.cerias.purdue.edu/site/about/history/coast/projects/vdb.php
https://wpvulndb.com
https://packetstormsecurity.com/
http://cve.mitre.org/
https://0day.today/
https://www.misp-project.org/features.html
https://cert.europa.eu/cert/newsletter/en/latest_SecurityBulletins_.html
http://www.cnnvd.org.cn/
https://www.us-cert.gov/ics/advisories
https://jvn.jp/en/
https://www.kyberturvallisuuskeskus.fi/en/homepage
https://securiteam.com/
https://securitytracker.com/
https://www.zerodayinitiative.com/advisories/published/
https://www.vulnspy.com/
https://github.com/AUEB-BALab/VulinOSS
https://oval.cisecurity.org/
https://seclists.org/fulldisclosure/
https://www.seebug.org/
https://cxsecurity.com/
https://en.0day.today/
https://developer.shodan.io/api/exploits/rest
https://www.talosintelligence.com/
https://www.us-cert.gov/ncas/bulletins
https://github.com/0x4D31/awesome-threat-detection
https://www.zerodayinitiative.com/advisories/published/
https://www.zerodayinitiative.com/advisories/upcoming/

## A.2 COMMERCIAL DATABASES

https://www.symantec.com/services/cyber-security-services/deepsight-intelligence
https://vulndb.cyberriskanalytics.com/
https://www.flexera.com/products/operations/software-vulnerability-management.html

https://www.accenture.com/us-en/blogs/blogs-vulnerability-intelligence
https://www.auscert.org.au/services/security-bulletins/
https://www.synopsys.com/software-integrity/security-testing/software-composition-analysis/technology/vulnerability-reporting.html
https://www.cisco.com/c/en/us/td/docs/security/firepower/Application_Detectors/library-vdb/fp-app-detectors-library.html
https://www.manageengine.com/vulnerability-management/help/vulnerability-database-settings.html

# B ANNEX: JUPYTER ENVIRONMENT INFORMATION

The Jupyter notebooks used in this report can be found at: https://github.com/enisaeu/vuln-report

## B.1 README.MD

### B.1.1 Getting started

**1. Install Anaconda**

Visit the Anaconda website and download the Anaconda installer for your OS (Python 3.7 version).

**2. Create Environment**

Clone this repository using terminal.

```
git clone https://github.com/enisaeu/vuln-report.git
```

Run the following from within the root of the repository

```
conda env create --file environment.yml
```

**3. Install required Jupyter Lab extensions**

In terminal, make sure you have the enisa environment activated.

```
conda activate enisa
```

Once activated, install the plotly Jupyter Lab extension.

```
jupyter labextension install @jupyterlab/plotly-extension
```

**4. Launch Jupyter Lab**

Run the following from within the root of the repository.

```
jupyter lab
```

If it asks you to build/rebuild make sure to accept all the prompts. You can keep an eye on the terminal window that launched Jupyter Lab to see when it's ready. You should refresh your browser once it's done.

**5. Check out the example notebook**

All done, check out the example notebook located at notebooks/example.ipynb.

## B.2 LIST OF JYPTER NOTEBOOKS

| Jupyter Notebook Name | Content |
|---|---|
| **0_table_of_contents** | Table of contents. |
| **1_average_CVSS_scores** | Comparison of different scores between vulnerabilities. |
| **2_high_CVSS_distribution** | Top products/vendors with the most vulnerabilities, correlation between vendors, CVSS scores and tactics. |
| **3_CWE_with_high_CVSS** | Top weaknesses, Top products/vendors with most (unique) weaknesses, Average CVSS scores for weaknesses, CWEs with high CVSS scores. |
| **4_top_10_weaknesses_per_product** | Weaknesses regarding web browsers and operating systems. |
| **5_top_vulnerabilities_per_sector** | Vulnerabilities in different sectors, top CVEs in them. |
| **6_vulnerabilities_in_web_vs_native** | CVSS scores, severity ratings, exploits, vendors In web browsers and windows applications. |
| **7_vulnerabilities_in_open_source** | CVSS scores, severity ratings, exploits, vendors, weaknesses in open source projects. |
| **8_events_attributed_to_vulnerabilities** | Incidents and vulnerabilities. |
| **9_high_score_and_known_exploits** | Exploitation based CVSS scores. |
| **10_vulnerability_lifecycle** | Exploits before/after published date, end of support. |
| **11_ATT&CK_capec** | ATT&CK Framework and CAPEC patterns. |
| **12_application_types** | Top software categories, cvss scores, ATT&CK and tactics. |
| **13_prices** | Analysis of the prices of Vulndb and Zerodium exploit price data. |

TP-03-20-014-EN-N

## ABOUT ENISA

The mission of the European Union Agency for Cybersecurity (ENISA) is to achieve a high common level of cybersecurity across the Union, by actively supporting Member States, Union institutions, bodies, offices and agencies in improving cybersecurity. We contribute to policy development and implementation, support capacity building and preparedness, facilitate operational cooperation at Union level, enhance the trustworthiness of ICT products, services and processes by rolling out cybersecurity certification schemes, enable knowledge sharing, research, innovation and awareness building, whilst developing cross-border communities. Our goal is to strengthen trust in the connected economy, boost resilience of the Union's infrastructure and services and keep our society cyber secure. More information about ENISA and its work can be found at www.enisa.europa.eu.

**ENISA**
European Union Agency for Cybersecurity

**Athens Office**
1 Vasilissis Sofias Str
151 24 Marousi, Attiki, Greece

**Heraklion office**
95 Nikolaou Plastira
700 13 Vassilika Vouton, Heraklion, Greece

enisa.europa.eu

ISBN: 978-92-9204-325-4
DOI: 10.2824/139301