A logo of a cat

Description automatically generated

**Data Driven Decision Making**

*Topic:*

***Prioritizing Patches and Updates for Cybersecurity***

*Group 4 Students*

[Gretchen N Meyers](https://psu.instructure.com/groups/989644/users/7278198)

[Vicente James Caballero Perez](https://psu.instructure.com/groups/989644/users/7175123)

[Gernie Lee Purnell](https://psu.instructure.com/groups/989644/users/7236089)

[Steven Douglas Ritchey](https://psu.instructure.com/groups/989644/users/7169551)

[Clayton Matthew Snyder](https://psu.instructure.com/groups/989644/users/7159320)

[Taylor B Turrisi](https://psu.instructure.com/groups/989644/users/6953216)

[Yang Xia](https://psu.instructure.com/groups/989644/users/7249662)

[Ping Xu](https://psu.instructure.com/groups/989644/users/7143091)

**School of Graduate Professional Studies**

MPS/MS in Data Analytics

DAAN 881 – Data Driven Decision Making

Fall, 2024

Table of Contents

[Document Control 1](#_Toc179142071)

[Work carried out by 1](#_Toc179142072)

[Revision Sheet 2](#_Toc179142073)

[Business Case 3](#_Toc179142074)

[Why This Matters for Businesses 4](#_Toc179142075)

[Cybersecurity Risk Reduction 4](#_Toc179142076)

[Compliance with Regulatory Standards 4](#_Toc179142077)

[Operational Efficiency 4](#_Toc179142078)

[Supply Chain Security 4](#_Toc179142079)

[Competitive Advantage 4](#_Toc179142080)

[Goal 5](#_Toc179142081)

[Description of Databases and Attributes 5](#_Toc179142082)

[Overview of Databases 5](#_Toc179142083)

[Dataset: PatchDB 6](#_Toc179142084)

[Source 6](#_Toc179142085)

[Description 6](#_Toc179142086)

[Citation 7](#_Toc179142087)

[Dataset: Cybersecurity in Application, Research, and Education (CARE) Lab 7](#_Toc179142088)

[Source 7](#_Toc179142089)

[Description 7](#_Toc179142090)

[Citation 7](#_Toc179142091)

[Dataset: Common Vulnerabilities and Exposures (CVE) Datafeeds 7](#_Toc179142092)

[Source 7](#_Toc179142093)

[Description 7](#_Toc179142094)

[Potential Queries 7](#_Toc179142095)

[Vulnerability Impact Prediction 8](#_Toc179142096)

[Exploitability Risk Assessment 8](#_Toc179142097)

[Trend Analysis of Vulnerabilities: 8](#_Toc179142098)

[Citation 8](#_Toc179142099)

[Titles 8](#_Toc179142100)

[Source 8](#_Toc179142101)

[Sponsors 9](#_Toc179142102)

[Number of Instances 9](#_Toc179142103)

[Number of Attributes 9](#_Toc179142104)

[Attributes 9](#_Toc179142105)

[cve 9](#_Toc179142106)

[date\_updated 9](#_Toc179142107)

[date\_reserved 9](#_Toc179142108)

[date\_published 9](#_Toc179142109)

[state 10](#_Toc179142110)

[assigner 10](#_Toc179142111)

[affected 10](#_Toc179142112)

[problem\_types 10](#_Toc179142113)

[v4\_score 10](#_Toc179142114)

[v4\_severity 10](#_Toc179142115)

[v31\_score 11](#_Toc179142116)

[v31\_severity 11](#_Toc179142117)

[v30\_score 11](#_Toc179142118)

[v30\_severity 11](#_Toc179142119)

[v20\_score 11](#_Toc179142120)

[v20\_severity 11](#_Toc179142121)

[attack\_vector 12](#_Toc179142122)

[attack\_complexity 12](#_Toc179142123)

[attack\_requirements 12](#_Toc179142124)

[privileges\_required 12](#_Toc179142125)

[user\_interaction 12](#_Toc179142126)

[exploit\_maturity 12](#_Toc179142127)

[Epss 13](#_Toc179142128)

[Percentile 13](#_Toc179142129)

[Number of Unique Values 13](#_Toc179142130)

[cve: 13](#_Toc179142131)

[date\_updated: 13](#_Toc179142132)

[date\_reserved: 13](#_Toc179142133)

[date\_published: 13](#_Toc179142134)

[state: 13](#_Toc179142135)

[assigner: 13](#_Toc179142136)

[affected: 13](#_Toc179142137)

[problem\_types: 14](#_Toc179142138)

[v4\_score: 14](#_Toc179142139)

[v4\_severity: 14](#_Toc179142140)

[v31\_score: 14](#_Toc179142141)

[v31\_severity: 14](#_Toc179142142)

[v30\_score: 14](#_Toc179142143)

[v30\_severity: 14](#_Toc179142144)

[v20\_score: 14](#_Toc179142145)

[v20\_severity: 14](#_Toc179142146)

[attack\_vector: 14](#_Toc179142147)

[attack\_complexity: 14](#_Toc179142148)

[attack\_requirements: 14](#_Toc179142149)

[privileges\_required: 14](#_Toc179142150)

[user\_interaction: 15](#_Toc179142151)

[exploit\_maturity: 15](#_Toc179142152)

[epss: 15](#_Toc179142153)

[percentile: 15](#_Toc179142154)

[Descriptive Statistics (Numeric Attributes) 15](#_Toc179142155)

[v4\_score: 15](#_Toc179142156)

[v31\_score: 15](#_Toc179142157)

[v30\_score: 15](#_Toc179142158)

[v20\_score: 16](#_Toc179142159)

[epss: 16](#_Toc179142160)

[percentile: 16](#_Toc179142161)

[v4\_severity: 17](#_Toc179142162)

[v31\_severity: 17](#_Toc179142163)

[v30\_severity: 17](#_Toc179142164)

[attack\_vector: 17](#_Toc179142165)

[attack\_complexity: 17](#_Toc179142166)

[attack\_requirements: 17](#_Toc179142167)

[privileges\_required: 18](#_Toc179142168)

[user\_interaction: 18](#_Toc179142169)

[exploit\_maturity: 18](#_Toc179142170)

[Preliminary Analysis, Visualization and Data Warehouse Suggestion 18](#_Toc179142171)

[Inconsistent Data Filling 18](#_Toc179142172)

[Possible Solution 19](#_Toc179142173)

[Outliers or Missing Scores 19](#_Toc179142174)

[Possible Solution 19](#_Toc179142175)

[Inconsistent Time Formatting 20](#_Toc179142176)

[Possible Solution 20](#_Toc179142177)

[Data Completeness Across Columns 21](#_Toc179142178)

[Possible Solution 21](#_Toc179142179)

[State/Status Redundancy 21](#_Toc179142180)

[Possible Solution 21](#_Toc179142181)

[Identify Columns that Lack Variability 21](#_Toc179142182)

[Possible Solution 22](#_Toc179142183)

[Additional Preliminary Analysis 22](#_Toc179142184)

[Check for Duplicate Rows 22](#_Toc179142185)

[Check for Unique Values in State Column 22](#_Toc179142186)

[Check for Inconsistent Categorical Values 22](#_Toc179142187)

# Document Control

## Work carried out by

*(This section is only required if the project is being completed by a group)*

|  |  |  |
| --- | --- | --- |
| **Name** | **Email Address** | **Task description** |
| Taylor Turrisi | tbt5027 | Setup Teams Collaboration Tools, Organization and Planning, Data Analysis, Deliverable 2 Submission |
| Steven Ritchey | sdr5515 | Business Case. Datasets, Deliverable 1 Submission |
| Clayton Snyder | cms8896 | Business Case. Datasets, Research, Creation and Maintenance of GitHub Repository |
| Vincente Perez | vzp5147 | Datasets, Research for Deliverable 2 Submission |
| Ping Xu | ymx5173 | Research for Deliverable 2 Submission |
| Gernie Lee | glp5261 | Data Inconsistencies and Errors Identification for Deliverable 3 Submission |
| Yang Xia | Ybx5258 | Visualization and Data Warehouse Solution for Deliverable 3 Submission |

## Revision Sheet

|  |  |  |
| --- | --- | --- |
| **Release No.** | **Date** | **Revision Description** |
| 1 | 2024/09/08 | Deliverable 1 – Topic Goal(s), Business Case, Overview of Databases |
| 2 | 2024/09/22 | Deliverable 2 – Description of Databases and Attributes |
| 3 | 2024/10/06 | Deliverable 3 – Preliminary Analysis and Suggestion |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Business Case

Operating systems and software applications are frequently updated with patches that enhance features, resolve bugs, and, most critically, address security vulnerabilities. Failing to install these updates can leave systems exposed to significant cyber risks. Unpatched vulnerabilities not only jeopardize the security of individual systems and sensitive user data but also pose a broader risk to interconnected business networks and critical data assets. Cyber attackers actively exploit these vulnerabilities to launch attacks, potentially resulting in data breaches, financial loss, and reputational damage.

However, the process of applying updates and patches introduces its own challenges. The installation process can be time-consuming, often requiring system downtime that disrupts business operations and reduces productivity. In some cases, organizations delay or avoid patching due to concerns about compatibility issues or potential disruptions to critical systems. Moreover, not all updates are created equal; some are vital for security, while others may have little impact, adding complexity to prioritization.

In addition to these operational considerations, regulatory pressure around software security is intensifying. The White House’s release of **Executive Order 14028** mandates that all software supplied to U.S. federal government systems comply with stringent secure software development standards. This includes the implementation of Software Bill of Materials (SBOMs), digital signatures, and security attestations to ensure transparency in the software supply chain and mitigate security risks.

These security requirements have extended beyond government contractors and are rapidly gaining traction across the commercial sector. SBOMs, in particular, provide a comprehensive inventory of the components within a piece of software, offering organizations increased visibility into potential vulnerabilities across their supply chain. As businesses increasingly rely on third-party software, ensuring compliance with SBOM standards is essential to maintaining trust, reducing the risk of hidden vulnerabilities, and achieving regulatory compliance.

# Why This Matters for Businesses

## Cybersecurity Risk Reduction

Proactively managing patches and updates is essential to closing known vulnerabilities and protecting against evolving cyber threats.

## Compliance with Regulatory Standards

As government requirements around SBOMs and secure software standards spread to the broader market, businesses must adopt these practices to remain competitive and avoid penalties.

## Operational Efficiency

Streamlined patch management processes reduce system downtime and minimize disruptions to productivity, helping businesses maintain operations while ensuring security.

## Supply Chain Security

By adopting SBOMs, organizations gain visibility into the components of third-party software, enabling them to address vulnerabilities at every level of the software supply chain.

## Competitive Advantage

Demonstrating compliance with secure software standards not only reduces the risk of cyberattacks but also positions businesses as trusted partners in both public and private sectors.

This business case highlights the urgency for organizations to invest in robust patch management solutions and to adopt SBOM practices, ensuring they can mitigate cybersecurity risks while adhering to evolving regulatory standards.

# Goal

This analysis seeks to develop a model that can prioritize updates and patching based on:

* current known vulnerabilities,
* current trends on which vulnerabilities have been exploited, and
* types of infrastructure of a system.

# Description of Databases and Attributes

## Overview of Databases

In the software industry, the MITRE Common Vulnerabilities and Exposures (CVE) repository serves as the primary resource for tracking and cataloging security vulnerabilities across various software systems. Since its inception in 1999, the CVE repository has grown to include nearly every known software vulnerability, offering a central, authoritative source for security teams and developers. Traditionally, MITRE distributed this data in a minimal format, often as CSV files that contained only basic metadata about each vulnerability.

In 2024, MITRE modernized this approach by transitioning to a more granular and data-rich format, distributing around 260,000 individual JSON files—one for each CVE. These JSON files contain extensive metadata for each vulnerability, including multiple metrics related to severity, complexity, exploitability, and other critical factors. This shift allows for a more comprehensive understanding of each vulnerability, providing organizations with the necessary information to make informed decisions regarding vulnerability management and patch prioritization.

To leverage this detailed dataset, we developed an algorithm capable of efficiently parsing each of the JSON files, extracting the attributes most pertinent to the task of creating a prioritization framework for vulnerability resolution and patching. The algorithm processes key metrics, such as severity scores, attack vectors, and complexity ratings, and compiles this information into a structured CSV file format, which is more suitable for analysis and integration into existing vulnerability management tools. The code for this data extraction and transformation process is openly available on GitHub at <https://github.com/ClaytonSnyder/daan881>.

In addition to leveraging the MITRE CVE program, we integrated data from the Exploit Prediction Scoring System (EPSS). EPSS is a data-driven method for estimating the likelihood that a software vulnerability will be exploited. The model generates probability scores ranging from 0 to 1 (0% to 100%) for all published CVEs, with higher scores indicating a greater chance of exploitation. EPSS is an emerging standard developed by a collaborative group of over 200 practitioners, researchers, and government personnel. The original EPSS model was released in August 2019, with the latest update being released in October 2023.

The Common Vulnerability Scoring System (CVSS) has evolved significantly from version 2 to the latest version 4, with each iteration improving the granularity and accuracy of vulnerability assessments. CVSS v2, introduced in 2007, provided a standardized method for evaluating vulnerabilities, using metrics like Access Vector and Authentication to assess severity. However, it was criticized for oversimplification and limited ability to capture real-world scenarios, particularly in addressing the confidentiality, integrity, and availability (CIA) of affected systems. In response, CVSS v3.0, released in 2015 and later refined in v3.1 in 2019, introduced more nuanced metrics such as Scope, Attack Vector, and Privileges Required, which improved its ability to assess vulnerabilities in complex environments. These changes helped offer a more detailed view of vulnerabilities, but the system still had limitations in terms of capturing the evolving nature of security risks. One clear example between v2 and v3 was around the number of severity ratings. V2 had three: low, medium and high whereas v3 has five: none, low, medium, high, and critical. Practitioners often complained about the lack of granularity and volume of “high” for which now “critical” is more nuanced and can be treated more sensitively.

With CVSS v4, currently under review as of 2024, further refinements are introduced, including new metrics like Modified Impact Metrics and a focus on the safety implications of vulnerabilities, particularly for physical systems and critical infrastructure. This version also emphasizes exploitability in dynamic environments, such as IoT and cloud systems, and provides greater adaptability and precision in scoring. Additionally, CVSS v4 enhances contextual awareness to reflect the real-world impact of vulnerabilities more effectively, making it better suited for today’s fast-changing threat landscape. As an example, the CVSS score was extended from only one score per vulnerability to multiple in v4. Another being the ability to offer more customization of metrics to extend the context.  Overall, each version of CVSS has aimed to improve transparency and relevance, with CVSS v4 offering the most detailed and flexible approach to date.

By concatenating the EPSS scores with data from the MITRE CVE Program, the resulting database offers a robust, multi-dimensional view of vulnerability severity. This enhanced database provides us with the ability to use the data driven decision making to prioritize vulnerabilities not only based on their inherent severity but also on the real-world risk of exploitation, thereby supporting a more proactive approach to vulnerability management.

## Dataset: PatchDB

### Source

[PatchDB (sunlab-gmu.github.io)](https://sunlab-gmu.github.io/PatchDB/)

### Description

Developed through the Center for Secure Information Systems, George Mason University, PatchDB is a large-scale security patch dataset that contains around 12K security patches and 24K non-security patches from the real world.

### Citation

*Xinda Wang, Shu Wang, Pengbin Feng, Kun Sun and Sushil Jajodia, "PatchDB: A Large-Scale Security Patch Dataset," 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2021), 2021, pp. 149-160, doi: 10.1109/DSN48987.2021.00030.*

## Dataset: Cybersecurity in Application, Research, and Education (CARE) Lab

### Source

Temple University, Cybersecurity in Application, Research & Education <https://sites.temple.edu/care/cira/>

### Description

A dataset of Critical Infrastructures Ransomware Attacks (CIRAs). These are based on publicly disclosed incidents in the media or security reports. This dataset **(version 12.10)** now has **1,898** records assembled from publicly disclosed incidents between **November 2013** and **August 31, 2024,** and has been mapped to the [**MITRE ATT&CK Framework**](https://attack.mitre.org/).

### Citation

*Rege, A. (2024). “Critical Infrastructure Ransomware Attacks (CIRA) Dataset”. Version 12.9. Temple University. Online at* [*https://sites.temple.edu/care/cira/*](https://sites.temple.edu/care/cira/)*. ORCID: 0000-0002-6396-1066.*

## Dataset: Common Vulnerabilities and Exposures (CVE) Datafeeds

### Source

The MITRE CVE dataset available at the [CVE Downloads page](https://cve.mitre.org/data/downloads/index.html).

### Description

This dataset includes detailed records of publicly disclosed cybersecurity vulnerabilities. It contains attributes such as CVE IDs, descriptions, affected products, and published dates. This data will be used to develop and train machine learning models for impact and exploitability predictions.

### Potential Queries

### Vulnerability Impact Prediction

#### Query

How can we predict the potential impact of newly discovered vulnerabilities (e.g., critical, high, medium, low) based on historical data and CVE characteristics?

#### Purpose

To prioritize vulnerabilities for patching and mitigation efforts based on their predicted impact, helping organizations allocate resources more effectively.

### Exploitability Risk Assessment

#### Query

What factors contribute to the likelihood that a vulnerability will be exploited in the wild, and how can we predict this likelihood using historical CVE data?

#### Purpose

To identify vulnerabilities with high exploitability risk and provide actionable insights for preemptive measures and enhanced security posture.

### Trend Analysis of Vulnerabilities:

#### Query

What are the trends in the frequency and types of vulnerabilities reported over time, and how can these trends inform future vulnerability management strategies?

#### Purpose

To understand patterns and trends in vulnerability disclosures, which can guide strategic planning and risk management in cybersecurity.

### Citation

"Common Vulnerabilities and Exposures (CVE) Data Feeds." MITRE, MITRE Corporation, [https://cve.mitre.org/data/downloads/index.html. Accessed 8 Sept. 2024](https://cve.mitre.org/data/downloads/index.html.%20Accessed%208%20Sept.%202024).

## Titles

1. MITRE Common Vulnerabilities and Exposures (CVE) Program
2. Exploit Prediction Scoring System (EPSS)

## Source

1. The MITRE Corporation
2. Forum of Incident Response and Security Teams (FIRST)

## Sponsors

1. U.S. Department of Homeland Security (DHS), Cybersecurity and Infrastructure Security Agency (CISA)
2. AlienVault, Cisco, Cyentia Institue, Efflux, F5, Fortinet, Greynoise Intelligence, Shadow Server

## Number of Instances

263,421

## Number of Attributes

24

## Attributes

### cve

Description: The unique CVE identifier for a vulnerability.   
Example: CVE-2023-12345   
Measurement Scale: Nominal

Missing Values: 0

### date\_updated

Description: The timestamp when the CVE entry was last updated in the CVE list.   
Example: 2024-08-01T16:03:04.917Z   
Measurement Scale: Interval

Missing Values: 0

### date\_reserved

Description: The date when the CVE ID was reserved but not yet published.   
Example: 2023-07-12T00:00:00Z   
Measurement Scale: Interval

Missing Values: 0

### date\_published

Description: The date when the CVE entry was published in the CVE list.   
Example: 2023-08-14T10:00:00Z   
Measurement Scale: Interval

Missing Values: 0

### state

Description: The status of the CVE entry, e.g., "PUBLISHED" when public or "RESERVED" when reserved for future publication.   
Example: “PUBLISHED”   
Measurement Scale: Nominal

Missing Values: 0

### assigner

Description: The organization that assigned the CVE.   
Example: “MITRE”   
Measurement Scale: Nominal

Missing Values: 0

### affected

Description: A space-delimited string that lists affected vendor and product pairs.   
Example: vendor::product such as apache::http\_server   
Measurement Scale: Nominal

Missing Values: 0

### problem\_types

Description: Lists the type of problem related to the vulnerability, such as "buffer overflow" or "SQL injection."   
Example: "buffer overflow"   
Measurement Scale: Nominal

Missing Values: 48

### v4\_score

Description: The CVSS version 4.0 base score representing the severity of the vulnerability, on a scale from 0 to 10.   
Example: 7.8   
Measurement Scale: Ratio

Missing Values: 261,641

### v4\_severity

Description: A qualitative representation of the severity based on the v4 base score.   
Example: "Low", "Medium", "High", or "Critical"   
Measurement Scale: Ordinal

Missing Values: 261,641

### v31\_score

Description: The CVSS version 3.1 base score, also representing vulnerability severity, on a scale from 0 to 10.   
Example: 5.3   
Measurement Scale: Ratio

Missing Values: 207,947

### v31\_severity

Description: A qualitative severity rating based on the CVSS v3.1 score.   
Example: “Medium”   
Measurement Scale: Ordinal

Missing Values: 207,947

### v30\_score

Description: The CVSS version 3.0 base score, another representation of vulnerability severity on a scale from 0 to 10.   
Example: 9.1   
Measurement Scale: Ratio

Missing Values: 246,557

### v30\_severity

Description: A severity rating based on the CVSS v3.0 score.   
Example: “Critical”   
Measurement Scale: Ordinal

Missing Values: 246,557

### v20\_score

Description: The CVSS version 2.0 base score, using the older CVSS version. Also on a scale from 0 to 10.   
Example: 4.7   
Measurement Scale: Ratio

Missing Values: 259,287

### v20\_severity

Description: Severity ratings based on CVSS v2.0 scores.   
Example: “Moderate”   
Measurement Scale: Ordinal

Missing Values: 207,947

### attack\_vector

Description: Describes how an attack is delivered   
Example: "Network", "Physical"   
Measurement Scale: Nominal

Missing Values: 0

### attack\_complexity

Description: Describes how difficult it is to execute an attack   
Example: "Low" or "High"   
Measurement Scale: Ordinal

Missing Values: 0

### attack\_requirements

Description: Lists the specific requirements for an attack to be successful, often derived from the attack complexity or privileges.   
Example: “NONE”, “PRESENT”   
Measurement Scale: Nominal

Missing Values: 0

### privileges\_required

Description: Describes the level of privileges needed for the attacker to execute an exploit.   
Example: "HIGH", "LOW", "NONE"   
Measurement Scale: Ordinal

### user\_interaction

Description: Indicates whether or not user interaction is required for an exploit to succeed.   
Example: “ACTIVE”, “REQUIRED, “NONE”   
Measurement Scale: Nominal

Missing Values: 0

### exploit\_maturity

Description: Describes how mature an exploit is   
Example: "HIGH", "PROOF\_OF\_CONCEPT", "FUNCTIONAL", "NOT\_DEFINED", “UNPROVEN”   
Measurement Scale: Ordinal

Missing Values: 0

### Epss

Description: The Exploit Prediction Scoring System (EPSS) score, representing the likelihood of a vulnerability being exploited in the wild. Values from 0-1.   
Example: 0.7234   
Measurement Scale: Ratio

Missing Values: 3,860

### Percentile

Description: The percentile ranking of the CVE’s EPSS score, e.g. proportion of all scored vulnerabilities with same or lower EPSS score.   
Example: 85.73   
Measurement Scale: Ratio

Missing Values: 3,860

## Number of Unique Values

### cve:

263,421

### date\_updated:

225,394

### date\_reserved:

33,720

### date\_published:

127.820

### state:

3

### assigner:

355

### affected:

30,565

### problem\_types:

14,196

### v4\_score:

55

### v4\_severity:

5

### v31\_score:

86

### v31\_severity:

6

### v30\_score:

83

### v30\_severity:

6

### v20\_score:

86

### v20\_severity:

1

### attack\_vector:

7

### attack\_complexity:

5

### attack\_requirements:

3

### privileges\_required:

4

### user\_interaction:

5

### exploit\_maturity:

6

### epss:

16,852

### percentile:

47,435

## Descriptive Statistics (Numeric Attributes)

### v4\_score:

1. Minimum:1.00
2. 1st Quartile: 5.30
3. Median: 6.00
4. Mean: 6.51
5. 3rd Quartile: 7.50
6. Maximum: 10.00

### v31\_score:

1. Minimum: 0.00
2. 1st Quartile: 5.40
3. Median: 6.50
4. Mean: 6.61
5. 3rd Quartile: 7.80
6. Maximum: 10.00

### v30\_score:

1. Minimum: 0.00
2. 1st Quartile: 5.30
3. Median: 6.40
4. Mean: 6.47
5. 3rd Quartile: 7.80
6. Maximum: 10.00

### v20\_score:

1. Minimum: 0.80
2. 1st Quartile: 4.00
3. Median: 5.80
4. Mean: 5.64
5. 3rd Quartile: 6.50
6. Maximum: 10.00

### epss:

1. Minimum: 0.000
2. 1st Quartile: 0.001
3. Median: 0.001
4. Mean: 0.031
5. 3rd Quartile: 0.004
6. Maximum: 0.976

### percentile:

1. Minimum: 0.004
2. 1st Quartile: 0.256
3. Median: 0.500
4. Mean: 0.503
5. 3rd Quartile: 0.750
6. Maximum: 1.000

**Class Distribution (Categorical Attributes)**

### v4\_severity:

1. Low: 34 (0.01%)
2. Medium: 1,161 (0.44%)
3. High: 452 (0.17%)
4. Critical:133(0.44%)

### v31\_severity:

1. Low: 3,549 (1.35%)
2. Medium: 27,055 (10.27%)
3. High: 19,993 (7.59%)
4. Critical: 4,852 (1.84%)

### v30\_severity:

1. Low: 1,896 (0.72%)
2. Medium: 7,786 (2.96%)
3. High: 5,994 (2.28%)
4. Critical: 1,187 (0.45%)

### attack\_vector:

1. Adjacent: 77 (0.03%)
2. Adjacent Network: 2,792 (1.06%)
3. Local: 12,168 (4.62%)
4. Network: 39,098 (14.84%)
5. Physical: 641 (0.24%)

### attack\_complexity:

1. Low: 46,099 (17.50%)
2. Medium: 1 (0.00%)
3. High: 8,677 (3.29%)

### attack\_requirements:

1. None: 474 (0.18%)
2. Present: 156 (0.06%)

### privileges\_required:

1. None: 28,820 (10.94%)
2. Low: 17,574 (6.67%)
3. High: 8,382 (3.18%)

### user\_interaction:

1. None: 35,591 (13.51%)
2. Passive: 72 (0.03%)
3. Active: 104 (0.04%)
4. Required: 19,009 (7.22%)

### exploit\_maturity:

1. Not Defined: 926 (0.35%)
2. Unproven: 1,889 (0.72%)
3. Proof of Concept: 392 (0.15%)
4. Functional: 85 (0.03%)
5. High: 452 (0.17%)

# [Preliminary Analysis, Visualization and Data Warehouse Suggestion](#_Preliminary_Analysis,_Visualization)

## Inconsistent Data Filling

Many columns, such as "affected", "problem\_type", and "cvss\_v4\_score", contain "n/a" values. A high number of missing values could hinder analysis, and the reasons for these gaps should be addressed. Missing data might need to be imputed or handled appropriately depending on the context.

There are several different types of columns where the n/a should not be considered for removal.  For example, “affected” is a combined list of known software that are compromised by the security vulnerability. This is a plain text field that should probably just be removed since it will probably just muddy our analysis if we keep it around.

### Possible Solution

Since the number of entries are large, it is possible to delete entries containing “n/a” values. However, deletion could cause a huge volume loss. Therefore, one can apply imputation methods such as filling the most frequent category (mode) or use mean or median. Nevertheless, such method does not apply to columns like “affected”. Therefore, one could make further investigation of missingness to find whether inconsistent collection or system update causes the inconsistencies to occur. We could merge the “severities” for each of these columns. Additionally, we should remove any rows that do not have at least one of these scores. For example, column entitled “problem\_type” is in a similar boat and will also probably be a column we should drop from the dataset. Columns “cvss\_v4\_score”, “cvss\_v3\_1\_score”,”cvss\_v3\_0\_score”, “cvss\_v2\_0\_score” could be merged into a single cvss column. This will reduce the number of empty values we have for each of them since they are iterations on the same score system.

## Outliers or Missing Scores

Columns such as "cvss\_v3\_1\_score" and "cvss\_v3\_1\_sev" have values filled inconsistently. Some records have severity scores like "HIGH," "CRITICAL," “Low,” “Medium,” or “None” Low while others are left blank. This inconsistency may introduce bias or affect the conclusions drawn from the data, especially if severity or scores are key variables in our analysis.

### Possible Solution

A blue box plot with black text

Description automatically generatedA blue box plot with black text

Description automatically generated

For outliers, one can use visualization to gain some insights. The left boxplot shows that around 50% percent of the ‘cvss\_v3\_1\_score’ lies within 4.7 and 7.9. Scores that are bigger than 10 or smaller than 2 should be considered as outliers. The boxplot for ‘cvss\_v4\_score’ shows a similar trend. Although those plots do not show accurate numbers for the distribution of data, if one use imputation methods to fill out missing values, the plots will have similar shapes.

A graph of a distribution of cvss

Description automatically generatedA graph of a distribution of cvss

Description automatically generated

For missing scores, the distributions of severity scores provide one with information about the trend for current non-missing values. For example, for certain undetermined missing severities in ‘cvss\_v4\_severity’, one can apply mode substitution and use ‘MEDIUM’ for the missing values. However, mode substitution may not be appropriate for ‘cvss\_v3\_1\_severity’ since both ‘HIGH’ and ‘MEDIUM’ are both the likely cases. Additionally, one could fill in missing scores based on patterns in related data, like using median severity scores for vulnerabilities with similar characteristics.

Inconsistent Time Formatting

The "date\_reserved," "date\_published," and "date\_updated" columns show some inconsistencies in time formatting. All dates appear in a uniform format (ISO 8601), but the presence of duplicate date values in some records suggests that data is updated over time. This could indicate potential data quality or integrity issues or we don’t have a live source. Additionally, for easy searching time and data could be separated.

Possible Solution

Apply standardization, and ensure all dates follow ISO 8601 format. For the duplicate values issue, investigate whether it’s a data entry error or indicative of versioning. Additionally, we could drop “deate\_reserved” and “date\_published” since “date\_updated” will always be the latest date.

Additionally, we have an abundance of data which may make analysis difficult. Since our data dates to 1999, we can sort the data by either published date or updated date (up for discussion) and remove all data beyond a threshold date in the past. This is an acceptable approach since we should assume that any system that is reasonably modern (and thus looking for valid patching strategies) would have patched “old” vulnerabilities a long time ago.

## Data Completeness Across Columns

Fields like "attack\_vector", "attack\_complexity", "privileges\_required", and "user\_interaction" are populated for some records, but many others remain empty. This lack of completeness can hinder the analysis of vulnerability patterns or comparisons between vulnerabilities.

### Possible Solution

Depending on the importance of the columns like "attack\_vector", "attack\_complexity", "privileges\_required", and "user\_interaction", it is better to keep them in the dataset now. Therefore, one could add binary columns that indicate whether a field was missing, allowing for easy tracking and exclusion in the further analysis.

## State/Status Redundancy

The "state" column shows "PUBLISHED" or “REJECTED.” The redundancy may not add much value to the dataset unless there are more varied statuses to track.

### Possible Solution

A graph of a distribution of state values

Description automatically generated

A bar plot is created to compare the counts between “PUBLISHED” and “REJECTED”. The bar plot of the ‘state’ column indicates that most of the values are ‘PUBLISHED’ while a minority of them are ‘REJECTED’. The data description for this column is: The status of the CVE entry, e.g., "PUBLISHED" when public or "REJECTED" when rejected for future publication.

## Identify Columns that Lack Variability

The "assigner," “affected,” and “problem type” column may not add value to the dataset unless variability is expected or it's essential for filtering based on assignments.

### Possible Solution

Evaluate the importance of columns. If it is not crucial, one could drop it to reduce the complexity of the model (since it will not contribute much to the model).

## Additional Preliminary Analysis

No duplicate rows exist. The state column only has 2 unique values. The severity only uses text not numbers in all cases. See the following link for preliminary analysis.

<https://github.com/ClaytonSnyder/daan881/blob/main/Deliverable3/problem_identification.ipynb>

**Deliverable 4 Assignment**

This week you will focus on cleaning your in-house dataset/database datamart/data warehouse. Based on the issues identified in your dataset/database/data mart/data warehouse and the proposal for resolving the issues (reported in Deliverable 3) you must to take appropriate steps to clean your data. You can use any software you like to clean your data. Some of the popular software used for data cleaning are:

* WEKA
* Excel
* R
* KNIME
* Python

Please visit [KDnuggets](https://www.kdnuggets.com/2011/04/free-tools-data-visualization-analysis.html) for a list of free tools available for data cleaning, visualization and analysis.

You will update your template document with the following:

* Describe the entire data cleaning process and the outcome for each of the data cleaning steps.
* Add appropriate tables and figures if needed.
* Feel free to add an appendix section if needed.

Submit your completed **Deliverable 4** to the Canvas assignment.

**Deliverable 5 Assignment**

This week the focus will be on variable selection, transformation, data reduction, etc. After data cleansing in the previous week, your deliverables for this week are noted below:

Update your template document with the following:

* Describe all the steps performed for variable selection, transformation, data reduction etc.
* Discuss your plans for data modelling *i.e.* what modelling steps will be performed in future week for addressing your research/business queries [**Example**: Perform a Multiple Linear Regression Model to determine the characteristic of the continuous response variable]

Submit your completed **Deliverable 5** to the Canvas assignment.

**Deliverable 6 Assignment**

This week the focus will be on data modeling. You will update the template document with the following:

* Describe all the steps performed for data modelling.
* Discuss the performance measure(s) used to determine the goodness of fit for the proposed model.
* Do you suspect your initial effort (model) suffers overfitting? If so, discuss what steps you took to overcome overfitting.
* Discuss your findings or inferences for each of the research/ business queries you identified in Deliverable 1.

Submit your completed **Deliverable 6** to the Canvas assignment.

**Deliverable 7 Assignment**

**Purpose:**

To provide a demonstration of your project in this course. In addition to submitting the final document, you will prepare a video presentation to share with the rest of the class.

**Tasks:**

**Part 1**

1. Before making the final submission, please ensure that you have implemented all instructor feedback from the previous Deliverables (1-6).
2. Please ensure that the title page, document control section, table of contents and the reference section is updated. Any appendix section should also be updated.
3. Submit the final document to the Deliverable 7 Canvas assignment in Lesson 14.
4. Submit your dataset(s), code/script, and PowerPoint to the Box account provided by your instructor.

**Part 2**

1. You will prepare a video presentation to demonstrate the designed data-driven Analytics System to the rest of the class.
2. The presentation should include PowerPoint slides and video demonstration.
3. Submit the link for your video presentation to the Project Presentation Discussion in Lesson 14.

Make all submissions by the due date noted in the Course Schedule.

More instructions (if needed) will be provided by the instructor via e-mail or Canvas Announcements.