Validating Vulnerability Analysis

with Statistical Analysis of Metadata

Alexander Bushkin | Keith Grant | Chess Hazlett | Marian Rehak
Red Hat Product Security

Red Hat

Overview and Motivation

Dramatically increased number of vulnerabilities reported lately. Same number of flaws means more flaws per analyst.

Vulnerabilities are each unique, but standards like CVSS & CWE reflect commonalities.

We have a lot of data. Can we use it to reduce time per vulnerability? To validate analysis as it happens?

Vulnerability Metadata and Its Uses

A brief overview of the data under consideration, what it represents, and how it's related.

Industry Standards for Vulnerability Metadata

- **CVSS -** CVSS is already a complex vector of values; the summarizing CVSS score is a known calculation.
- **CWE -** There are some inferred relationships between CWE and CVSS, but nothing defined.
- **Component -** Not a standard form of metadata (though CPE could function this way), different components are affected by different vulnerabilities.
- **Impact -** The variable under consideration since this is what determines our level of effort, so we need accuracy

Clustering Prerequisites

Requirements for Good Cluster Analysis

- Feature selection, mutual information, orthogonality
- Relative cluster sizes, cluster counts, dataset dimensionality
- Our choices and why we made them (for visualization and analysis)

CVSS

Easy but not simple to calculate

CVSS v3.1 Equations
The CVSS v3.1 equations are defined below.
Base
The Base Score is a function of the Impact and Exploitability sub score equations. Where the Base score is defined as,

If (Impact sub score <= 0) *0 else,*
Scope Unchanged[4] $Roundup(Minimum[(Impact + Exploitability), 10])$
Scope Changed $Roundup(Minimum[1.08 \times (Impact + Exploitability), 10])$

and the Impact sub score (ISC) is defined as,

Scope Unchanged $6.42 \times ISC_{Base}$
Scope Changed $7.52 \times [ISC_{Base} - 0.029] - 3.25 \times [ISC_{Base} - 0.02]^{15}$

Where,

$ISC_{Base} = 1 - [(1 - Impact_{Conf}) \times (1 - Impact_{Integ}) \times (1 - Impact_{Avail})]$

And the Exploitability sub score is,

$8.22 \times AttackVector \times AttackComplexity \times PrivilegeRequired \times UserInteraction$

Temporal
The Temporal score is defined as,

$Roundup(BaseScore \times ExploitCodeMaturity \times RemediationLevel \times ReportConfidence)$

Environmental
The environmental score is defined as,

If (Modified Impact Sub score <= 0) 0 else,

If Modified Scope is Unchanged Round up(Round up (Minimum [(M.Impact + M.Exploitability) ,10]) × Exploit Code Maturity × Remediation Level × Report Confidence)

If Modified Scope is Changed Round up(Round up (Minimum [1.08 × (M.Impact + M.Exploitability) ,10]) × Exploit Code Maturity × Remediation Level × Report Confidence)

And the modified Impact sub score is defined as,

If Modified Scope is Unchanged $6.42 \times [ISC_{Modified}]$

If Modified Scope is Changed $7.52 \times [ISC_{Modified} - 0.029] - 3.25 \times [ISC_{Modified} \times 0.9731 - 0.02]^{13}$

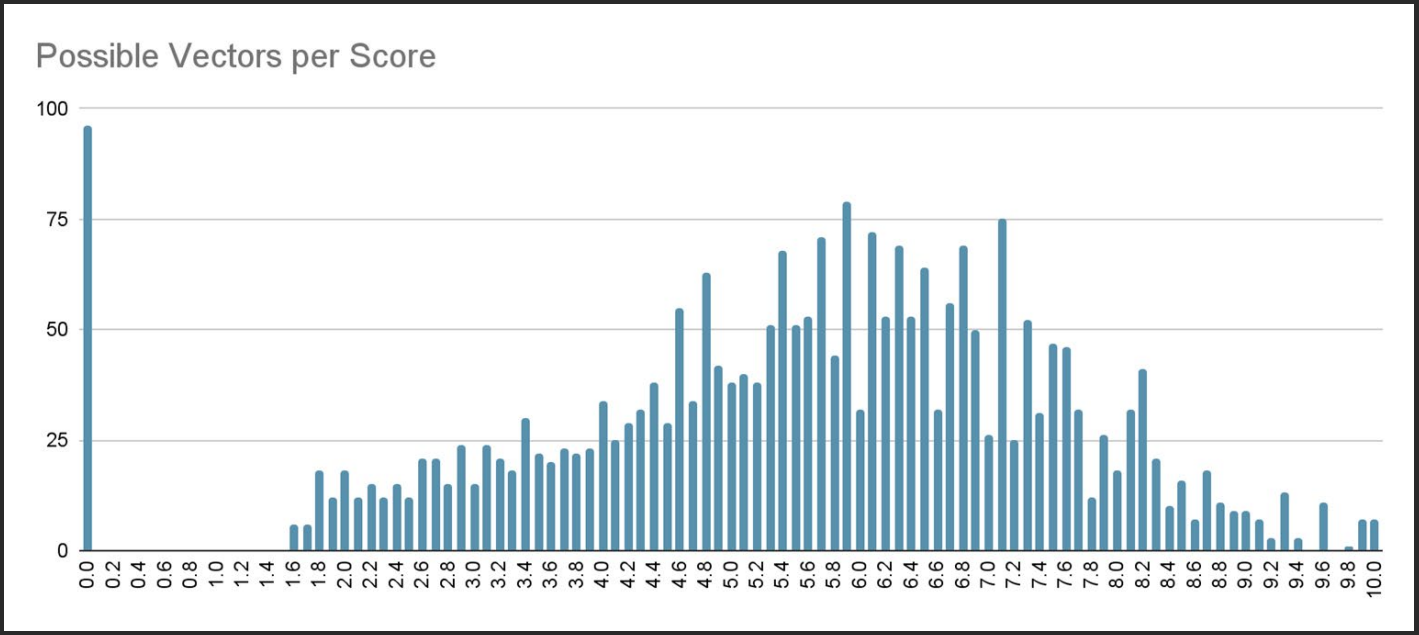
Where,
$ISC_{Modified} = Minimum [[1 - (1 - M.IConf \times CR) \times (1 - M.IInteg \times IR) \times (1 - M.IAvail \times AR)], 0.915]$

The Modified Exploitability sub score is,

$8.22 \times M.AttackVector \times M.AttackComplexity \times M.PrivilegeRequired \times M.UserInteraction$

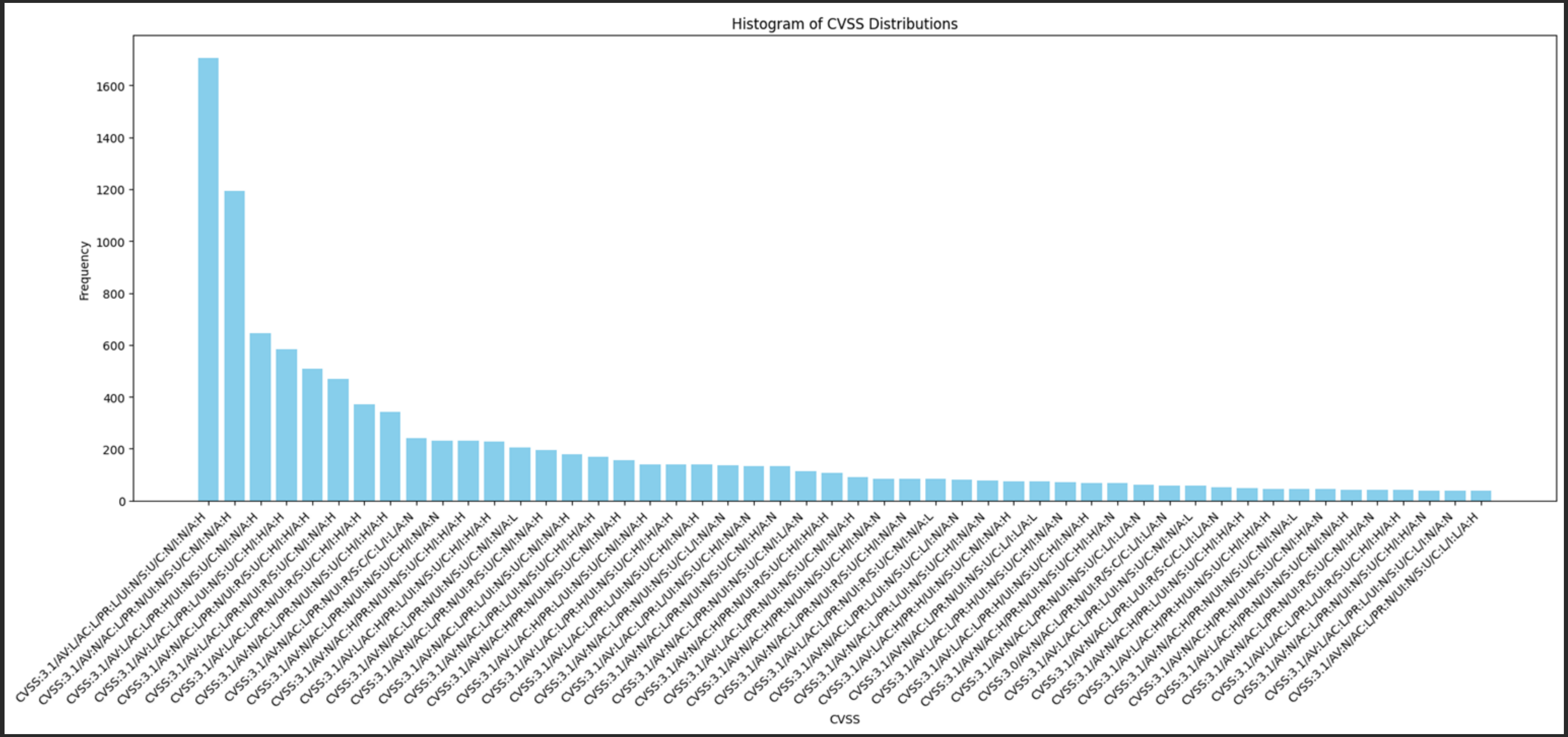
CVSS

Some Brief Trivia



- **2592** unique CVSS vectors
  - We have **836** unique vectors
- **101** "possible" scores
  - **17** are unreachable
  - 101 - 17 = **84** actual scores
  - We have **80** unique scores
- How many are represented in our vulnerabilities, and how are they distributed?

CVSS Vector Distributions (Ours)

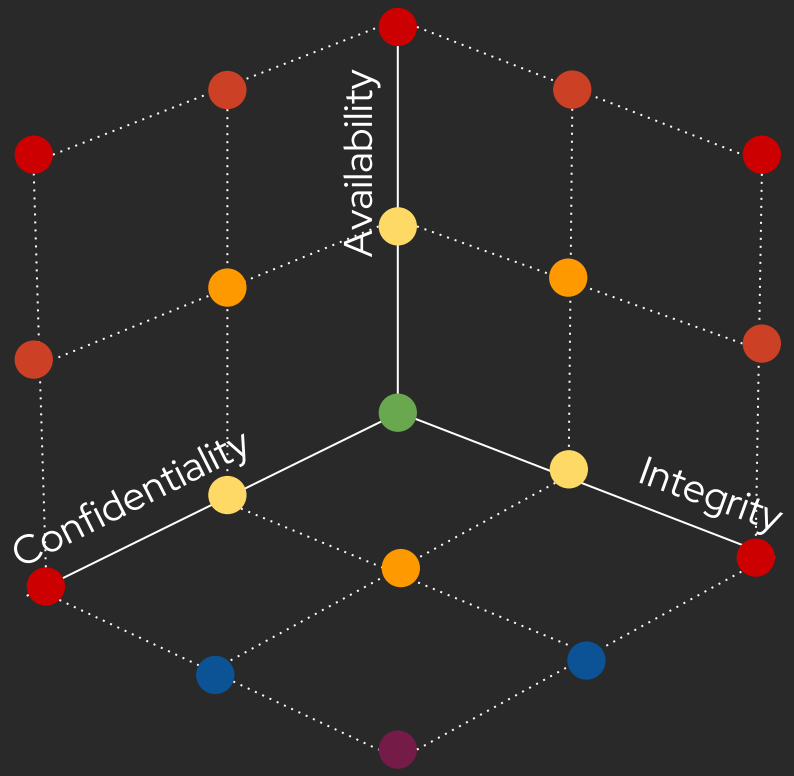


CVSS and Impact

	encoded_impact
AV	0.349903
AC	-0.002062
PR	0.284493
UI	-0.020142
S	0.098400
C	0.398274
I	0.444726
A	0.110656

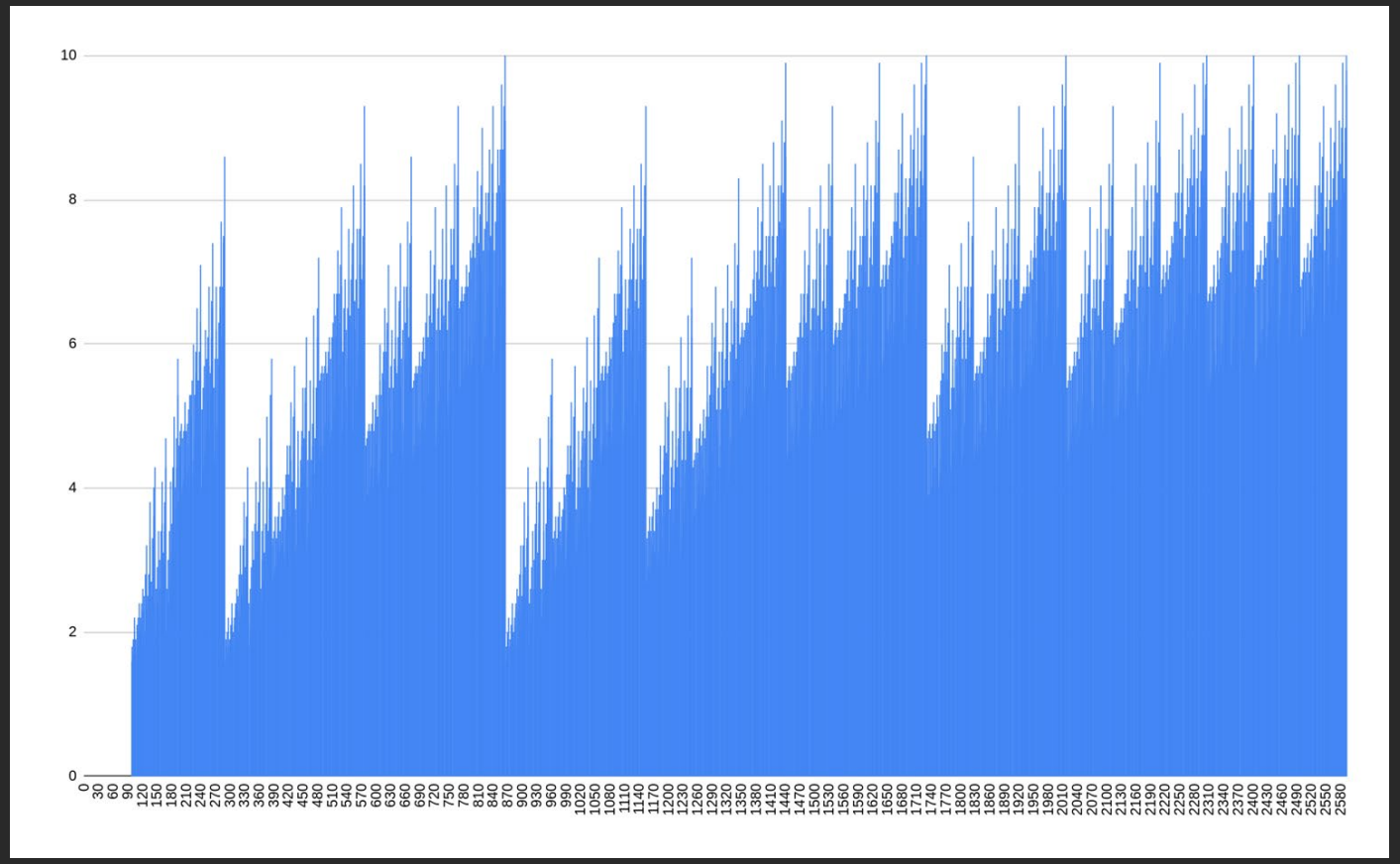
- Unsurprisingly, the most influential CVSS elements on Impact are Integrity & Confidentiality
- Attack Vector is a little surprising, but the difference in impact between Local and Network is intuitive
- Availability is a little surprising. Is this noise in our data?

CVSS - C/I/A Cube!



- **C/I/A** aren't perfectly orthogonal, but they're close enough
- Using C/I/A as basis vectors, you can calculate the "distance" between two vulnerabilities
- 3 x 3 x 3 =**27 unique** C/I/A triads

CVSS Encoding: Modulo



- Need a way to group similar CVSS vectors based on vector features, not just overall score.
  - Ex. differentiating between “low” importants and “high” importants
- Solution: modulo
- Magnitude ordering (most to least important): C, I, A, AV, PR, AC, UI, S
  - The ordering is informed by Impact in our dataset, but you could use any ordering

Further reading: https://en.wikipedia.org/wiki/Mixed_radix

CWE

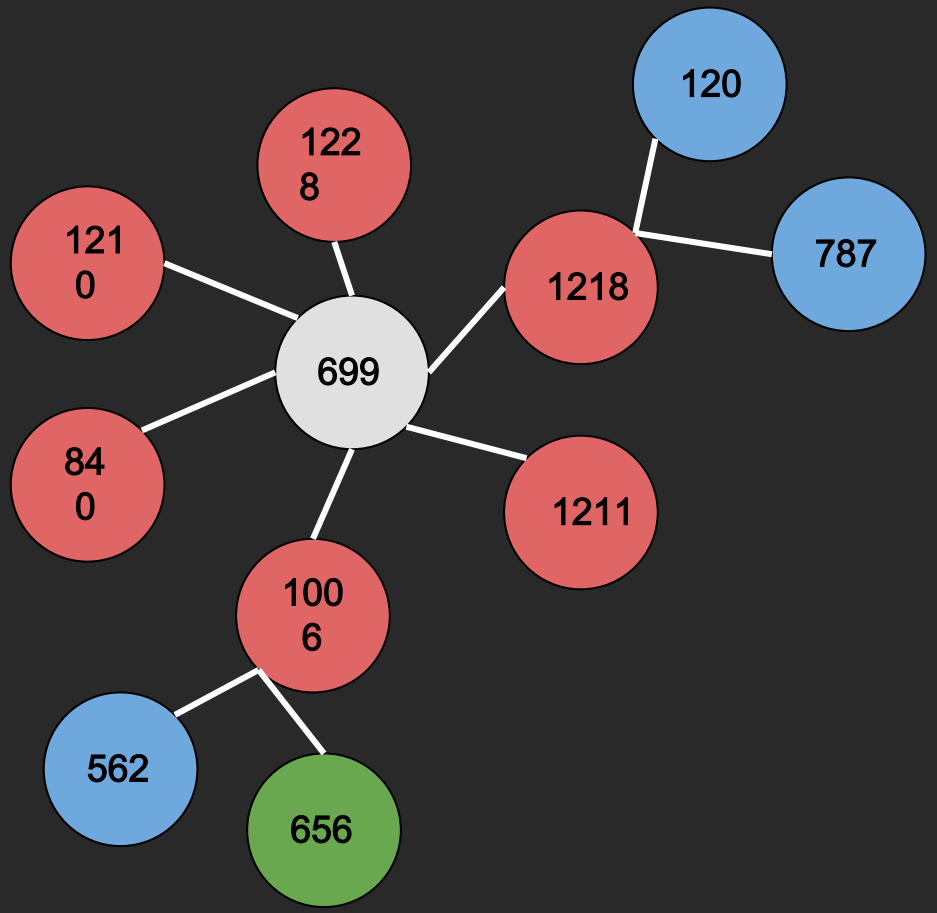
Unique information represented with CWE

```
================== TOP 10 CWES ==================
1. CWE-476: 579
2. CWE-416: 566
3. CWE-20: 375
4. CWE-79: 359
5. CWE-400: 338
6. CWE-125: 327
7. CWE-119: 289
8. CWE-200: 274
9. CWE-787: 215
10. CWE-401: 155
```

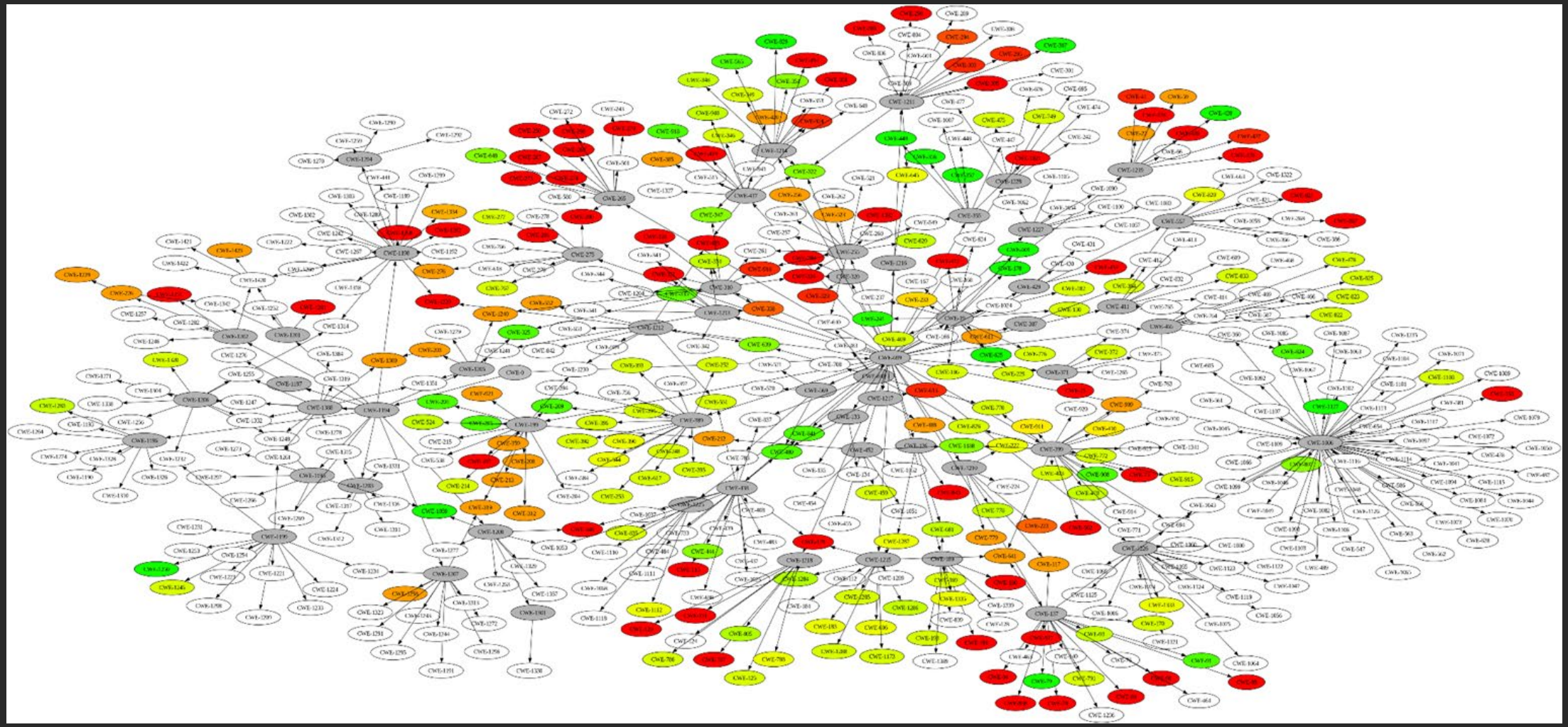
CWEs have *some* relationship, how do we discover it?

```
=============== TOP (5) COMPONENTS TO CWE ===============
1. kernel: {
  "CWE-476": 438,
  "CWE-416": 279,
  "CWE-125": 121
}
2. Mozilla: {
  "CWE-120": 56,
  "CWE-416": 39,
  "CWE-1021": 29
}
3. chromium-browser: {
  "CWE-416": 89,
  "CWE-843": 23,
  "CWE-122": 15
}
4. OpenJDK: {
  "CWE-770": 14,
  "CWE-20": 10,
  "CWE-248": 5
}
5. QEMU: {
  "CWE-125": 13,
  "CWE-476": 12,
  "CWE-835": 9
}
```


CWE "Space"



- CWE is **already** in a graph form
- The distance between two CWEs is just graph distance
- Views mean the graph isn't acyclic, but that's not insurmountable



Component Encoding: Buckets

Are components subject to a specific set of vulnerabilities and weaknesses?

- Every flaw has a list of impacted component(s)
- There is an inherent relationship present in component lists, so we don't want to split them up
- Similar lists should go together
  - Ex. [kernel], [kernel, usb], …
- Solution: buckets


```
"['kernel']": {
   "bucket": 1,
   "unique_count": 603,
   "comp_list": [
     [
       "kernel"
     ],
     [
       "kernel",
       "kvm"
     ],
     [
       "kernel",
       "kvm",
       "nvmx"
     ],
     ...
   ],
   "graphing_space": [
     1.0,
     1.0016445182724252,
     1.0032890365448506,
     ...
   ]
},
```

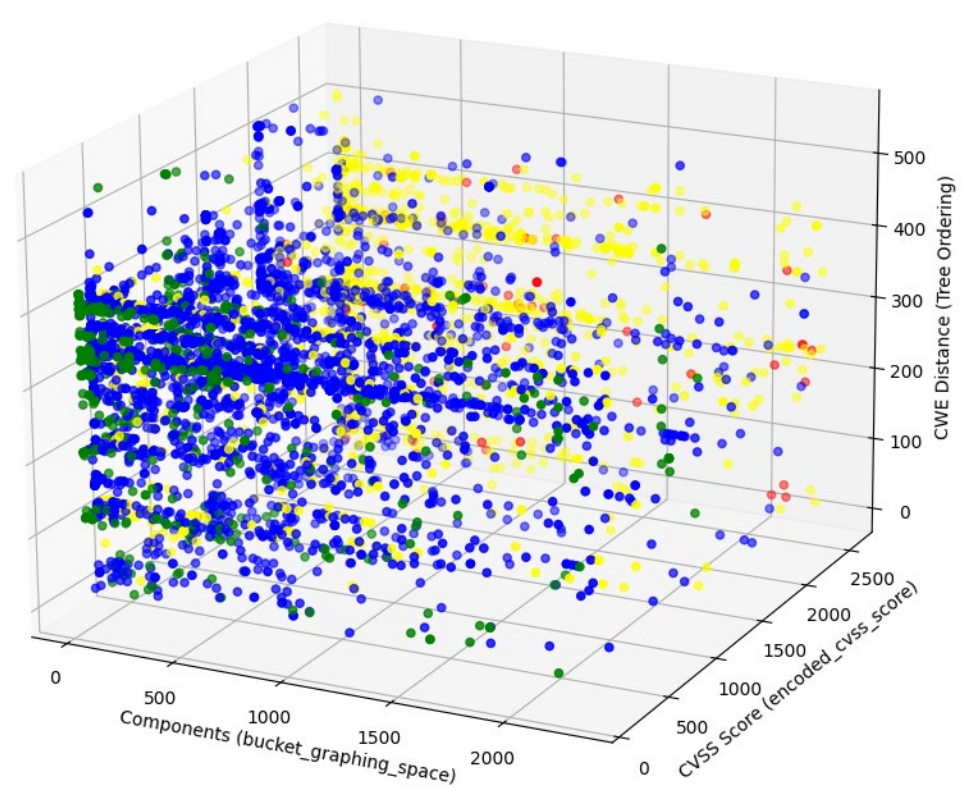

Vulnerability Impact Cluster Analysis

Experimental clustering of vulnerabilities based on metadata to identify the boundaries between vulnerabilities of different severity.

Our Dataset

- Limited to a specific date range
  - 2020 - 2024
- Some CWE IDs not in 699 or 1194 views, need to be remapped? (What about chains?)
- Inherent skews in the data:
  - Over-representation of kernel flaws
  - Internal workflow processes may result in over-abundance of specific metadata (ex. CVSS = 5.5)
- Result:
  - Going from a total 14,278 flaws collected to a filtered, useable total of 7,415

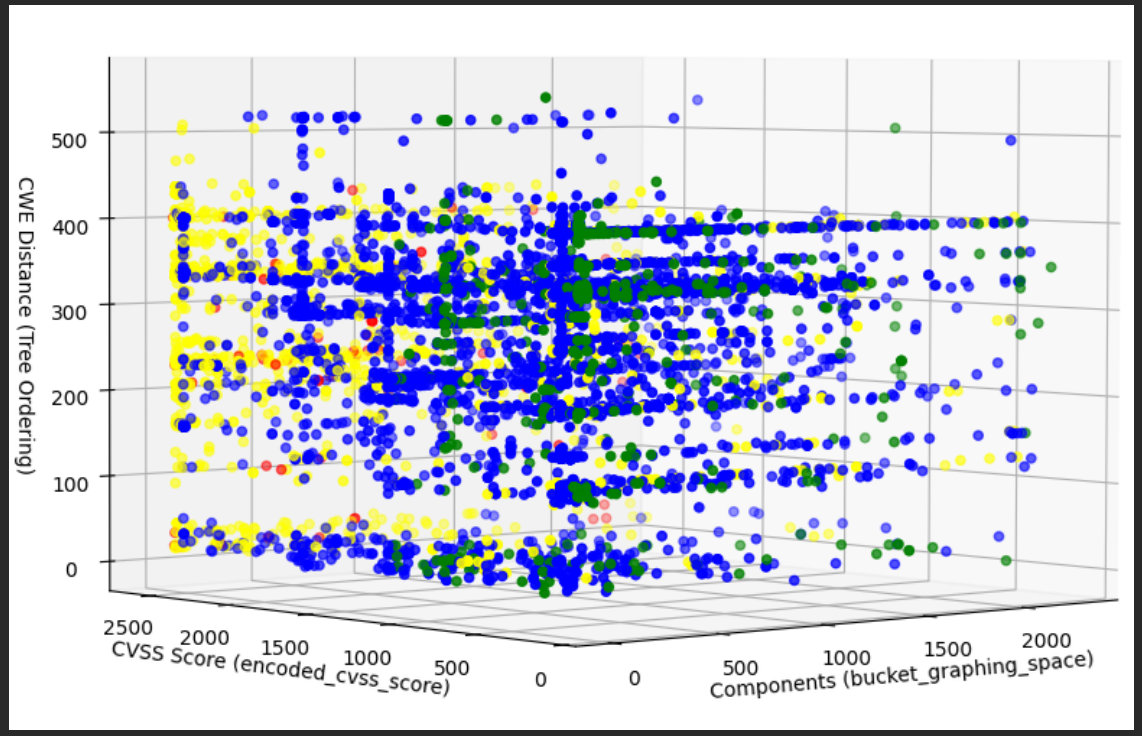
Clustering - Components



Notice how the component buckets create varying densities which reflect the prevalence of a given product.

Ex. kernel bucket is located at Components Axis = 1.01.99.

Clustering - CWE

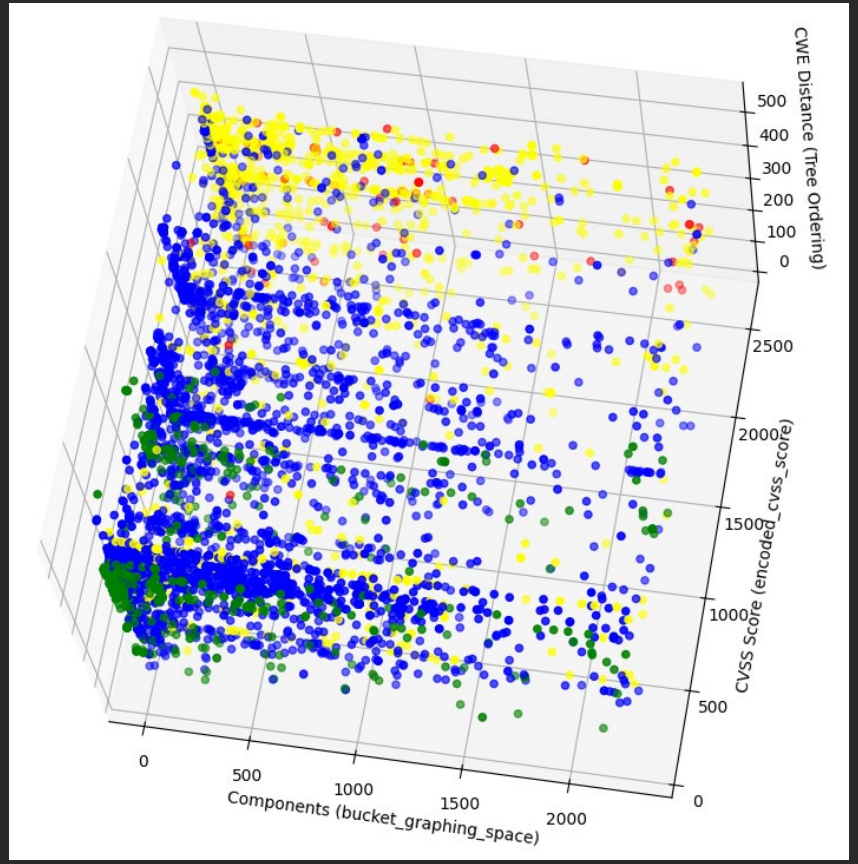


The CWE Distance Axis maps the 699 view followed by the 1194 view using a DFS approach.

Both views have IDs that overlap (range-wise), as such, banding is less clear.

CWE relationship to clusters becomes more obvious with further analysis (more on this later)

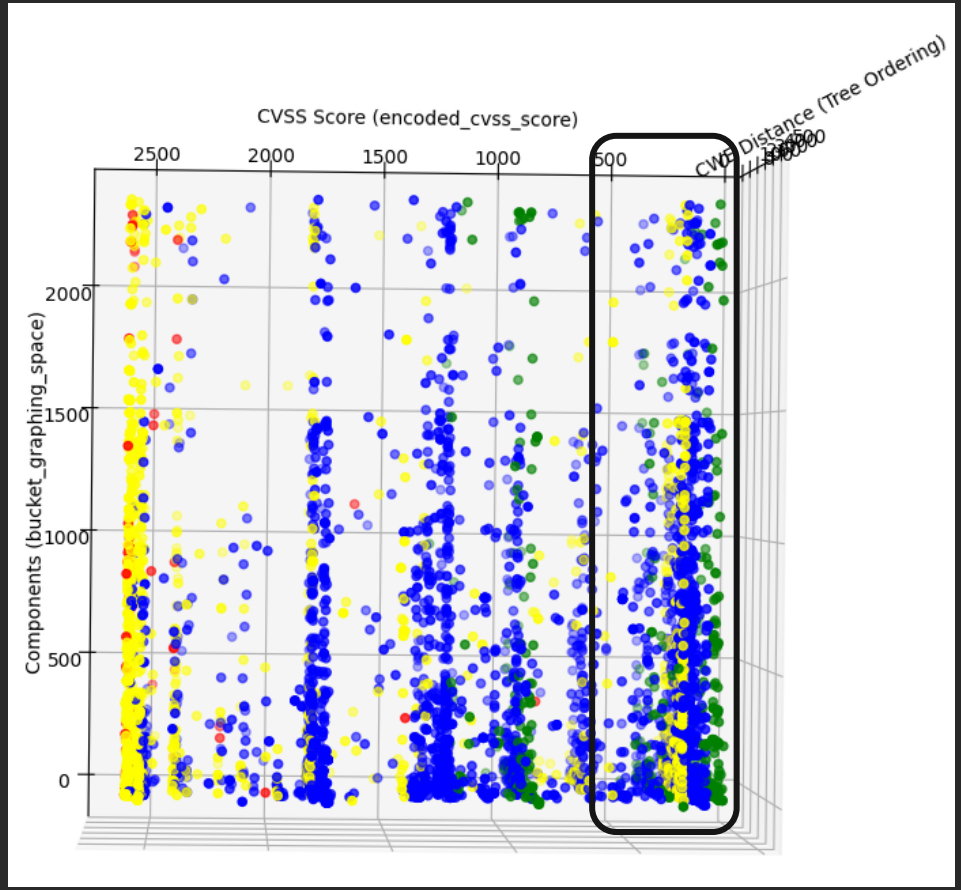
Clustering - CVSS



Notice the banding along the CVSS Score Axis, caused by how modulo works (hence there are repeated potential ranges where no CVSS scores will be present).

The banding also groups flaws by similarity rather than severity alone (more on this later).

Clustering - “Band 1” Analysis



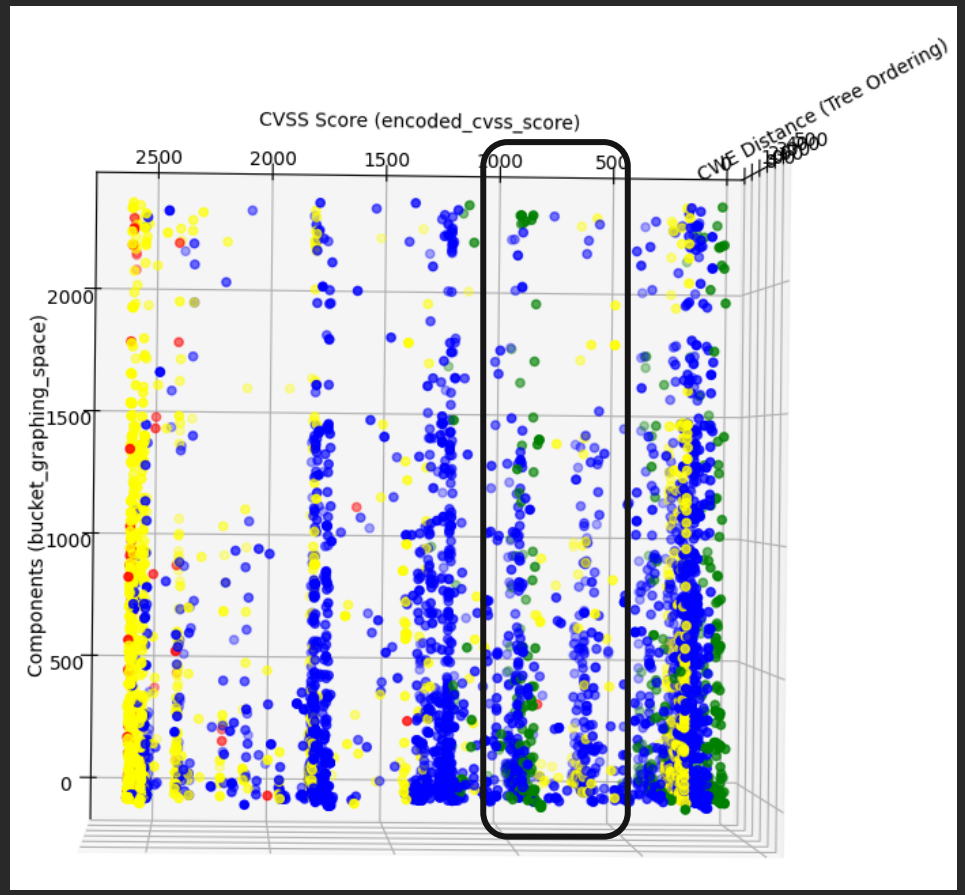
“DOS-ville”

1. LOWS:
   a. CIA: NNL
   b. CVSS: 3.3
   c. CWE: 401

2. MODERATES:
   a. CIA: NNH
   b. CVSS: 5.5
   c. CWE: 366

3. IMPORTANTS:
   a. CIA: NNH
   b. CVSS: 7.5
   c. CWE: 401

Clustering - “Band 2” Analysis



“CONF - ville”

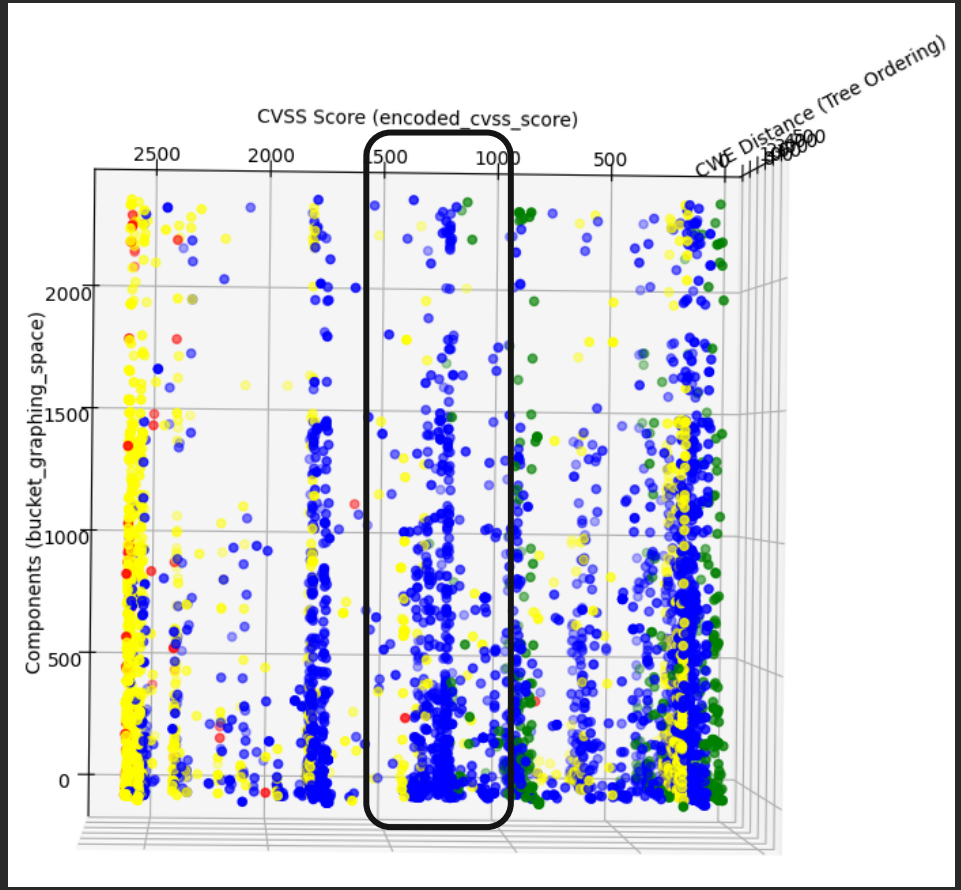
1. LOWS:
    a. CIA: LNN
    b. CVSS: 3.3
    c. CWE: 296

2. MODERATES:
    a. CIA: LNN
    b. CVSS: 5.3
    c. CWE: 296

3. IMPORTANTS:
    a. CIA: 7.5
    b. CVSS: NHN
    c. CWE: 112

4. CRITICALS:
    a. CIA: 9.1
    b. CVSS: NHH
    c. CWE: 223

Clustering - “Band 3” Analysis



“Mixed Bag”

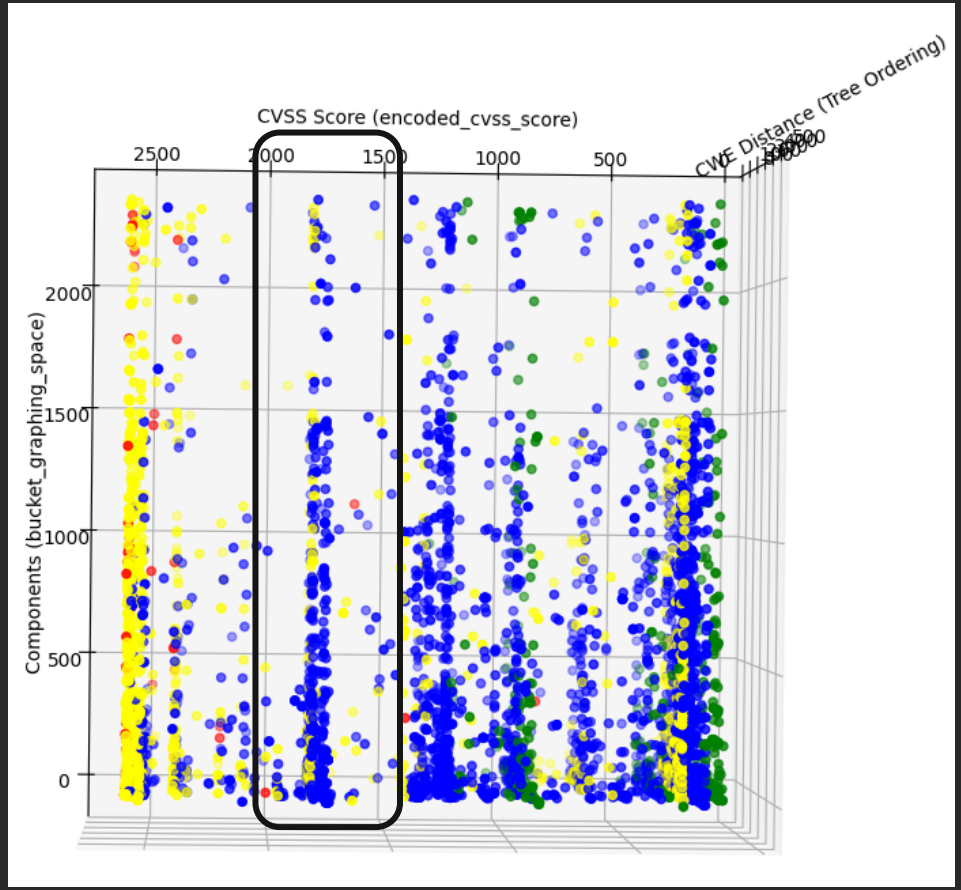
1. LOWS:
    a. CIA: LLN
    b. CVSS: 3.6
    c. CWE: 231

2. MODERATES:
    a. CIA: LLN
    b. CVSS: 6.1
    c. CWE: 231

3. IMPORTANTS:
    a. CIA: LLH
    b. CVSS: 8.6
    c. CWE: 338

4. CRITICALS:
    a. CIA: LLH
    b. CVSS: 9.1
    c. CWE: 369

Clustering - “Band 4” Analysis



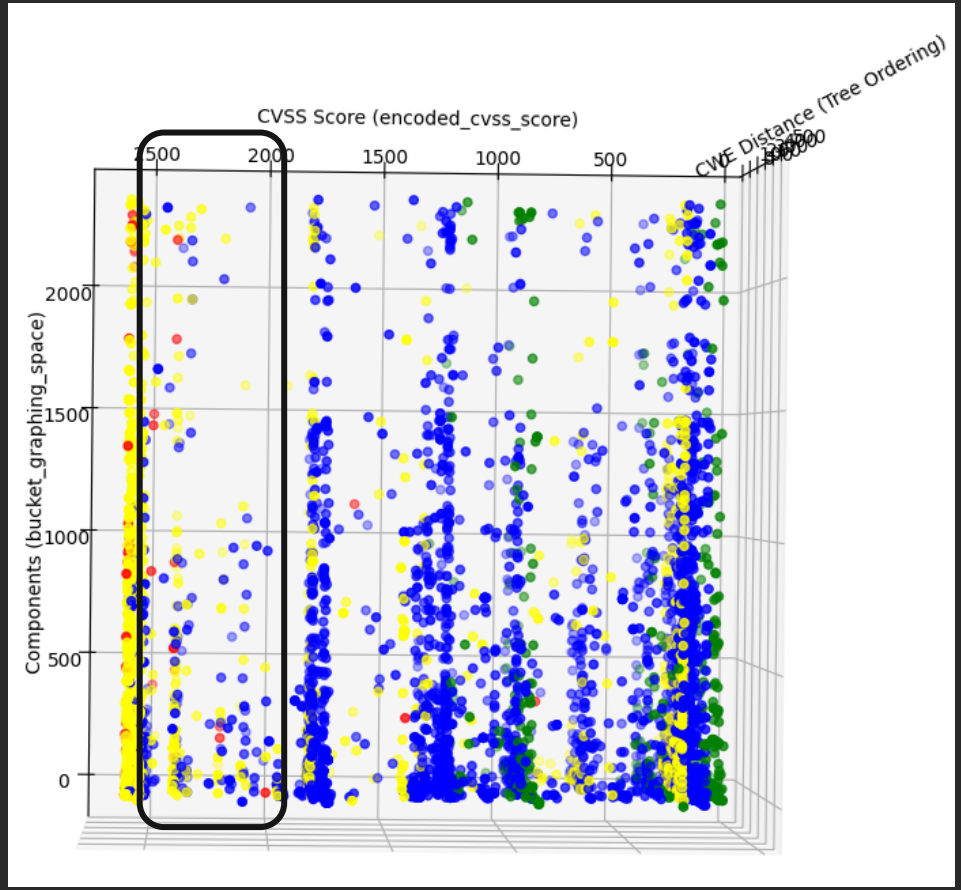
“PII-ville”

1. MODERATES:
    a. CIA: HNN
    b. CVSS: 5.5
    c. CWE: 296

2. IMPORTANTS:
    a. CIA: HNN
    b. CVSS: 7.5
    c. CWE: 255

3. CRITICALS:
    a. CIA: LHL
    b. CVSS: 9.9
    c. CWE: 136

Clustering - “Band 5” Analysis



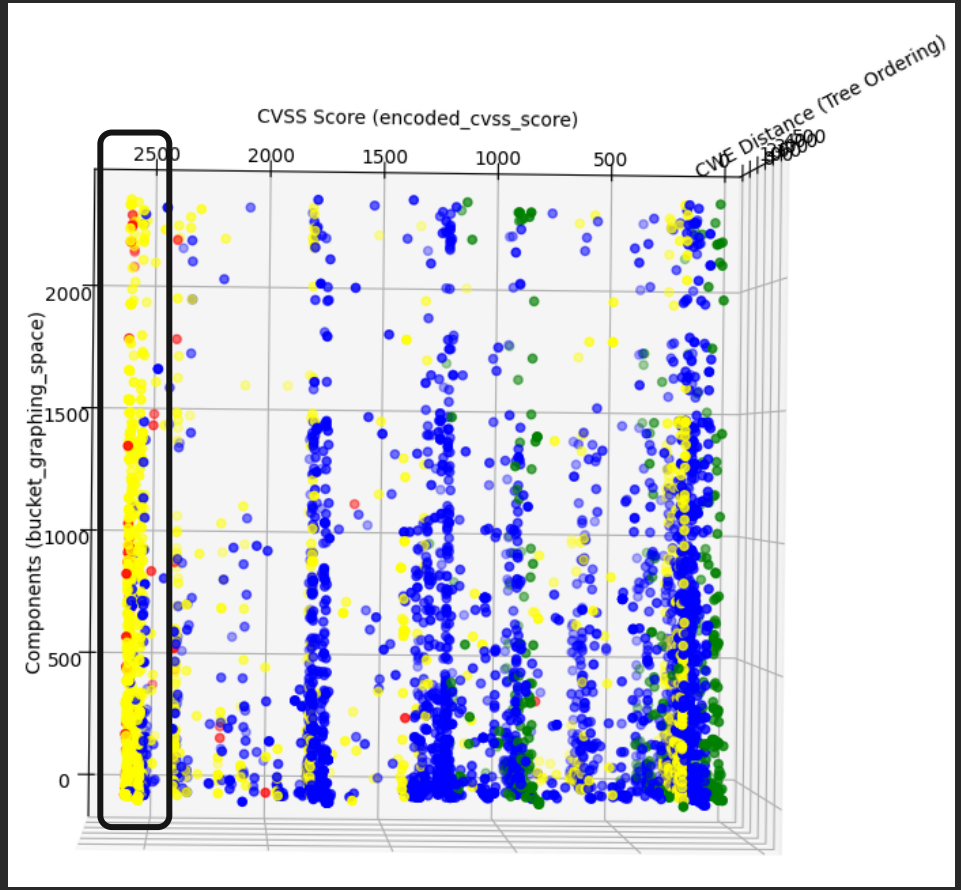
“DATA MANIPULATION -ville”

1. MODERATES:
   a. CIA: HHN
   b. CVSS: 6.8
   c. CWE: 231

2. IMPORTANTS:
   a. CIA: HHN
   b. CVSS: 8.1
   c. CWE: 231

3. CRITICALS:
   a. CIA: HHN
   b. CVSS: 10.0
   c. CWE: 208

Clustering - “Band 6” Analysis



“WORST CASE-ville”

1. MODERATES:
   a. CIA: HHH
   b. CVSS: 6.7
   c. CWE: 401

2. IMPORTANTS:
   a. CIA: HHH
   b. CVSS: 8.8
   c. CWE: 401

3. CRITICALS:
   a. CIA: HHH
   b. CVSS: 9.8
   c. CWE: 401

Clustering - Key Takeaways

CIA Trends to CWE Explanations

- CVSS Indexing (using our modulo approach) organizes similar flaws together, despite different Impact levels
  - Ex. the 0-500 CVSS index range primarily contains flaws that affect Availability (hence “DOS-ville”)
  - Ex. the 2000-2500 CVSS index range primarily contains flaws that affect Confidentiality and Integrity (hence “Data Manipulation-ville”)
- Cross-referencing with CWE, we are able to extrapolate a potential explanation for some of the CIA trends we see in the different clusters
  - Ex. CWE-401 (“Missing Release of Memory after Effective Lifetime”) is one of the most prominent weaknesses in the “DOS-ville” cluster
  - Ex. CWE-231 (“Improper Handling of Extra Values”) is one of the most prominent weaknesses in the “Data Manipulation-ville” cluster

Analysis Validation via Vulnerability Metadata

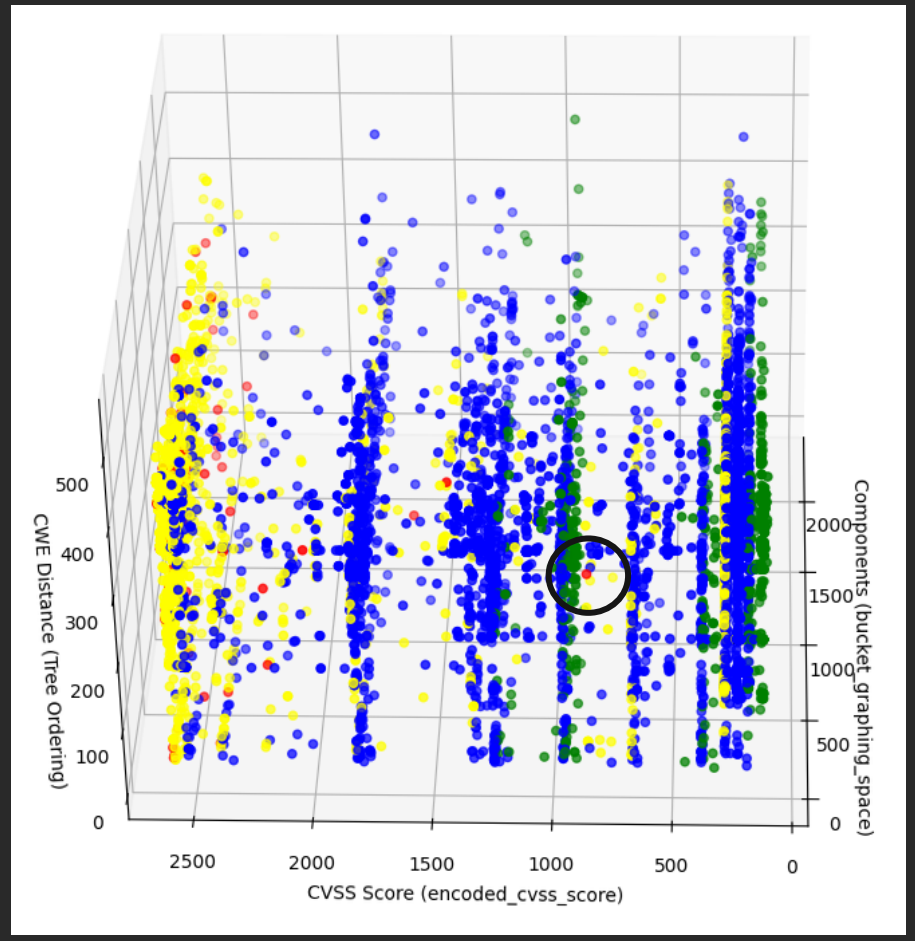
Having identified impact clusters, we experiment with locating new vulnerabilities in those clusters to validate initial analysis of new vulnerabilities.

Future Research

Refining these findings and keeping the data up to date

- Are the assigned CVSS/CWE "unusual"?
  - Making sure we avoid regression to the mean
- Are there past flaws we should double-check?
  - Sometimes flaws are just outliers. Those can be interesting case studies!
- When working on a new vulnerability, based on the metadata, what neighborhood does it appear to be in? What are some similar flaws?
- How can we improve our methodology? Are there better search techniques or data structures?

Future Research- Outlier Example



- The only CRITICAL flaw in Band 2.
- CVE-2024 -40896.
- Affects libxml2.
- Allows for XML External Entity (XXE) attacks.
- The rare situation where you can have a CIA triad be: NHH.

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat