Universidade do Minho

Engenharia de Segurança

Vulnerabilities Mapping

Raphael J. S. Pinheiro
pg37160@alunos.uminho.pt

Vulnerability Management

- Area of Information Security
- Management of vulnerabilities found in:
    - pentests
    - bug bounty programs
    - user contribution
    - researches

Vulnerability Management

- Various Data Source
  - Nessus
  - Qualys
  - Acunetix
  - OpenVas
  - Nmap
  - Nexpose
  - so on

Vulnerability Representation

	Nessus	**Qualys**
ID	10669	10340
Name	A1Stats Multiple Script Traversal Arbitrary File Access	Drummon Miles A1Stats Directory Traversal Vulnerability
Categories	infos	Remote Discovery, Patch Available, Exploit Available
Family	CGI abuses	CGI
CVE	CVE-2001-0561	CVE-2001-0561
CVSS Score	Medium / CVSS Base Score : 5.0	7.5
Bugtraq ID	2705	2705

Problem

How to map vulnerabilities from different sources?

Solution

- Compare the attributes:
  - Title
  - CVE
  - References
- Calculate the similarity

Solution

- 2 reference datasets:
    - Known matches
    - Known unique vulnerabilities (non-matches)

Build Reference Datasets

- Matches:
  - 141 entries
  - N to N

- Not Matches:
  - 67 entries/scanner
  - 134 entries total
  - Vulns that do not have mappings

Build Test DB

- Actual DB: ~150MB (csv files) and ~220k entries
- Test DB: <1MB (csv files) and 407 entries

Comparing Attributes

- Python FuzzyWuzzy
- Title:
  - Token Sort Ratio (doesn’t ignore duplicates, but ignores order)
  - Never empty
- CVE:
  - Token Set Ratio (ignores duplicates)
  - May be empty
- Refs:
  - Token Set Ratio (ignores duplicates)
  - May be empty

Calculating Similarity

```
Similarity = (TitleRatio + 2*CVERatio + 2*RefsRatio) / (1 + 2*[0|1] + 2*[0|1])

Example:

Similarity = (0.65856 + 2*0.72341 + 2*0) / (1 + 2 + 0) = 0.70179
```

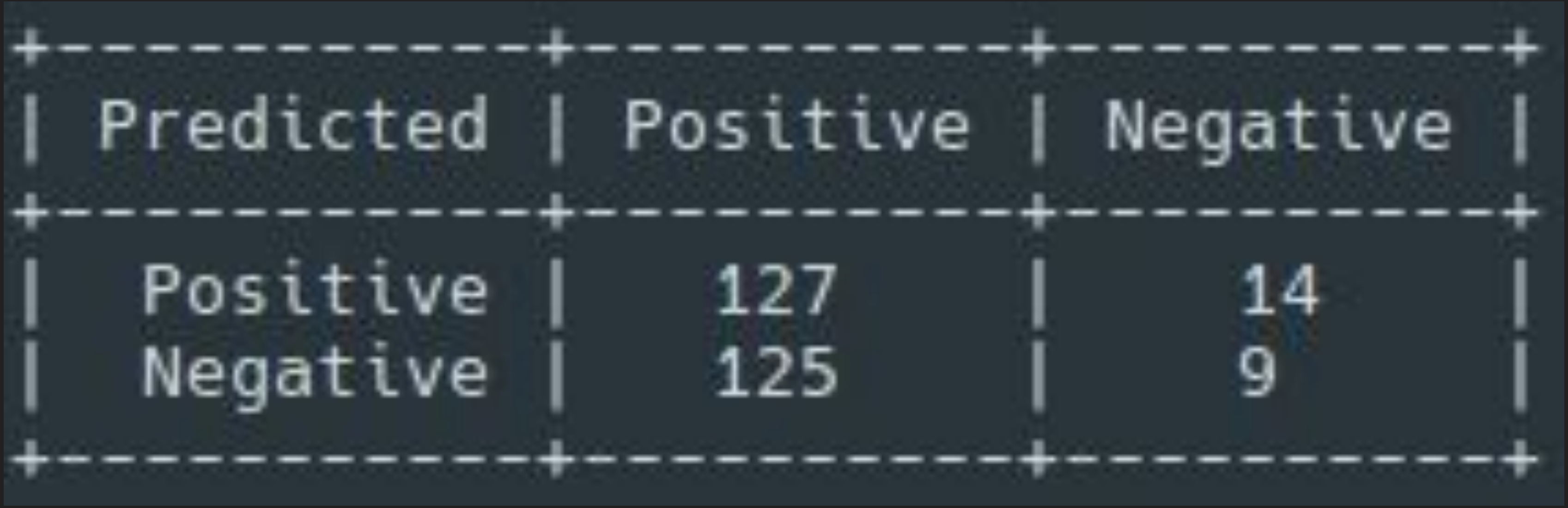
[0|1] => If CVE/Refs were empty, it doesn’t count

Matches Results

- True Positives:
    - For each entry *E* on the DB, do:
        - s = similarity(P, E)
    - if E == Q and s > 0.65:
        - True Positive

- False Positives:
    - For each entry *E* on the DB, do:
        - s = similarity(P, E)
    - if E == Q and s < 0.65:
        - False Positive

Confusion Matrix



```
+-----------+----------+----------+
| Predicted | Positive | Negative |
+-----------+----------+----------+
|  Positive |   127    |    14    |
|  Negative |   125    |    9     |
+-----------+----------+----------+
```

Accuracy, Precision, Recall and F_1 Score

- Accuracy = Correct Classification / Total Entries = 0.4945
- Precision = TP / (TP + FP) = 127 / (127 + 14) = 0.9007
- Recall = TP / (TP + FN) = 127 / (127 + 125) = 0.5040
- F_1 Score = 2*(Precision * Recall)/(Precision + Recall) = 0.6463

Future Work

- Graphs:
  - ROC (Receiver Operating Characteristic)
  - AUC (Area Under the Curve)
- Adjust Similarity Formula
- Use more attributes to compose the formula
- Run it through the actual DB (~220k entries)
- Add other sources (OpenVAS, Acunetix, etc)