An Investigation into Vulnerability Databases

Bachelor Thesis
(final presentation)

31st March 2020

Brian Schweigler
Used Terms

**provider**
organization and people behind a vulnerability database

**vulnerability report**
a vulnerability, an exploit, or a security-related document submitted to a provider

**severity**
potential impact of a vulnerability
libraries with security flaws
Gecko HTML renderer / Australis XUL GUI / ...
libraries with security flaws
mpg123 / ffmpeg / ...

?
... but how can we leverage vulnerability databases?
Our Idea

- software
- vulnerability statistics
  - shows library with security flaws
- dependency information
  - shares library with security flaws

- library
- application

- FFmpeg
- Windows Media Player
What Do We Need to Know?

RQ1

What are the affiliations between the different vulnerability database providers, and what data do they collect?
First, we need data!
First, we need data!

LOTS of data.
## Evaluated Providers

<table>
<thead>
<tr>
<th>Provider</th>
<th>Free</th>
<th>Founded in</th>
<th>Owner</th>
<th>Source</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE</td>
<td>yes</td>
<td>1999</td>
<td>Mitre</td>
<td>open</td>
<td>simple, unbiased overview</td>
</tr>
<tr>
<td>Exploit-DB</td>
<td>yes</td>
<td>2004</td>
<td>Offensive Security</td>
<td>open</td>
<td>exploit and app download</td>
</tr>
<tr>
<td>NVD</td>
<td>yes</td>
<td>2000</td>
<td>NIST</td>
<td>closed</td>
<td>lists affected software</td>
</tr>
<tr>
<td>RAPID7</td>
<td>no</td>
<td>2000</td>
<td>RAPID7</td>
<td>closed</td>
<td>evaluates package use</td>
</tr>
<tr>
<td>Snyk</td>
<td>yes</td>
<td>2006</td>
<td>Snyk Ltd.</td>
<td>closed</td>
<td>evaluates package use</td>
</tr>
</tbody>
</table>
Data Collection

- Vendor db
- Servers
- Interface
- Kotlin code (web-scraping & regex matching)
- Unified db
Data Processing

Python scripts applied to vulnerability descriptions for ...

- tokenizing strings
- counting word frequencies in descriptions
- grouping by dates
- removing stop-words
- plotting (140+)
Exploit-DB requires a valid HTTP header for requests
User-Agent: <product> / <product-version>

Exploit-DB requires regular IP address changes

character set encoding issues
(who needs special characters?!)
## Our Dataset

<table>
<thead>
<tr>
<th>Provider</th>
<th>First Report</th>
<th>Cut-off date</th>
<th># Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE</td>
<td>1999</td>
<td>12.12.2019</td>
<td>156,828</td>
</tr>
<tr>
<td>NVD</td>
<td>1999</td>
<td>03.12.2019</td>
<td>133,477</td>
</tr>
<tr>
<td>Exploit-DB</td>
<td>1988</td>
<td>01.11.2019</td>
<td>44,539</td>
</tr>
<tr>
<td>RAPID7</td>
<td>2004</td>
<td>18.12.2019</td>
<td>30,849</td>
</tr>
<tr>
<td>Snyk</td>
<td>2005</td>
<td>17.11.2019</td>
<td>4,605</td>
</tr>
</tbody>
</table>
What Do We Need to Know?

RQ1
What are the affiliations between the different vulnerability database providers, and what data do they collect?
Vulnerability Provider Affiliations

 CVE

 Exploit-DB

 NVD

 Snyk

 RAPID7
## Supported Feature Sets

<table>
<thead>
<tr>
<th>Provider</th>
<th>ID</th>
<th>CWE</th>
<th>Author</th>
<th>Title</th>
<th>Description</th>
<th>Reference</th>
<th>Date</th>
<th>Score</th>
<th>Special</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>NVD</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Exploit-DB</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>RAPID7</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Snyk</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
What Do We Need to Know?

**RQ1**
What are the affiliations between the different vulnerability database providers, and what data do they collect?

**RQ2**
Are the vulnerability scoring systems used consistently?
“The Common Vulnerability Scoring Standard (CVSS) provides a way to capture principal characteristics of a vulnerability and produce a numerical score [1-10] reflecting its severity.”

— FIRST (Forum of Incident Response and Security Teams)
CVSS Details

multiple versions are in use (CVSS V2, CVSS V3, CVSS V3.1)
# Scoring

<table>
<thead>
<tr>
<th>Provider</th>
<th>CVSS V2</th>
<th>CVSS V3</th>
<th>V2 Average</th>
<th>V3 Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE</td>
<td>X</td>
<td>X</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>NVD</td>
<td>✔️</td>
<td>✔️</td>
<td>6.2</td>
<td>7.4</td>
</tr>
<tr>
<td>Exploit-DB</td>
<td>X</td>
<td>X</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>RAPID7</td>
<td>✔️</td>
<td>X</td>
<td>6.3</td>
<td>n/a</td>
</tr>
<tr>
<td>Snyk</td>
<td>X</td>
<td>✔️</td>
<td>n/a</td>
<td>6.6</td>
</tr>
</tbody>
</table>
What Do We Need to Know?

RQ1
What are the affiliations between the different vulnerability database providers, and what data do they collect?

RQ2
Are the vulnerability scoring systems used consistently?

RQ3
Can we see trends based on the collected data?
Determination of Affected Software

200 most frequent non-stop words found in the descriptions: (filtered by software names)

Acrobat, Android, Apache, Apple, Chrome, Chromium*, FFmpeg*, Flash, Firefox, Java, Linux, Thunderbird, Windows, Wordpress
... and then we created some plots!
Trends - Shared Code (Thunderbird / Firefox)
Trends - Shared Code (FFmpeg / Android)
Trends - Related Software (Microsoft / Linux)
Trends - Related Software (Microsoft / Linux)
Future Work

machine learning to find corresponding patterns

improved NLP on descriptions
Summary

RQ1: Affiliations & Data

RQ2: Scoring

RQ3: Trends