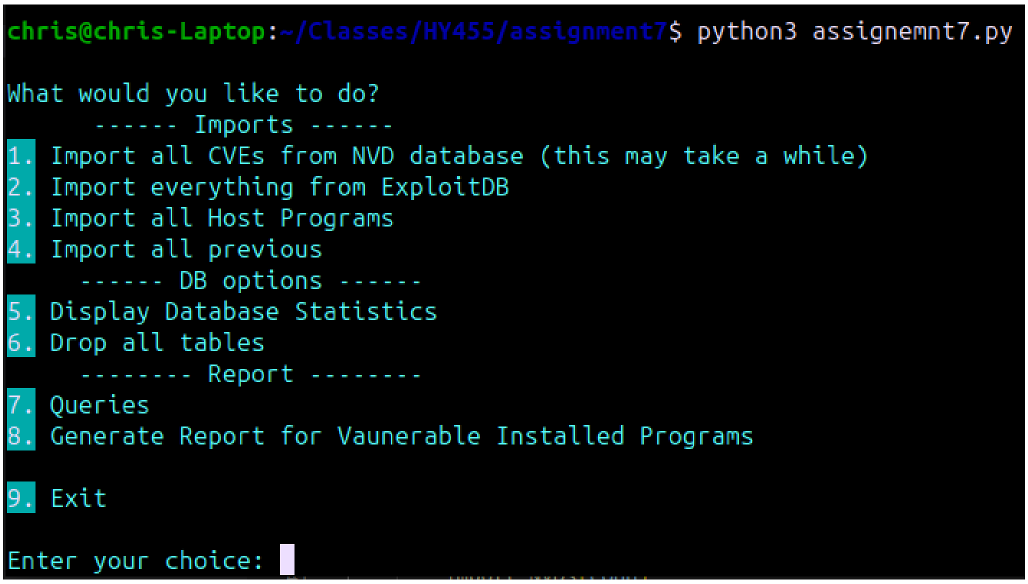
HY455 Cyber Security
Assignment 7 - Vulnerability Assessment
Chris Papastamos (csd4569)

For the purpose of this assignment I created a tool which implements the functionality needed by the assignment's questions.

This tool consists of 5 python files in which I am going to explain along with the process of completing the questions' tasks. The main file of the tool is called assignment7.py and when ran it greets the User with a menu displaying the available options. This menu can be seen in the screenshot below:



```
chris@chris-Laptop:~/Classes/HY455/assignment7$ python3 assignemnt7.py

What would you like to do?
     ------ Imports ------
1. Import all CVEs from NVD database (this may take a while)
2. Import everything from ExploitDB
3. Import all Host Programs
4. Import all previous
    ------ DB options ------
5. Display Database Statistics
6. Drop all tables
    -------- Report --------
7. Queries
8. Generate Report for Vaunerable Installed Programs

9. Exit

Enter your choice:
```

Question 1

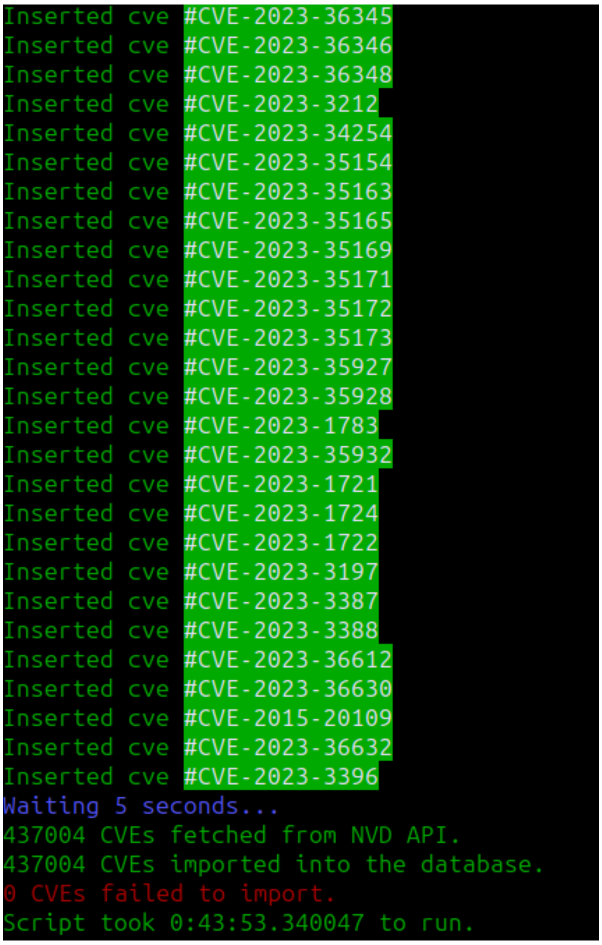
The first step for this question was to setup a mySQL database in my local computer. That was pretty straight forward as all I needed to do was to install mySQL using apt and set a password for the root user. Also I needed to create a database which I called cve_db.

In order to import all the CVEs in the database I created earlier, I had to write a python script (option 1 in the main manu) which will query the NVD API and store the results in the database. The NVD API had some limitations so I had to implement the following rules for my queries:

1. Only query for a 120 day window at a time.
2. Wait 10 seconds after each query because otherwise the API returns 403 replies
3. Request 2000 results per page each time
4. If TotalResults > startIndex + resultsPerPage increment the startIndex and request the same dates from the API

The script for this functionality can be found in the NVD_to_mySQL.py file. The main function of the script is the import_NVDs() function which calls the fetch_and_import_cves().

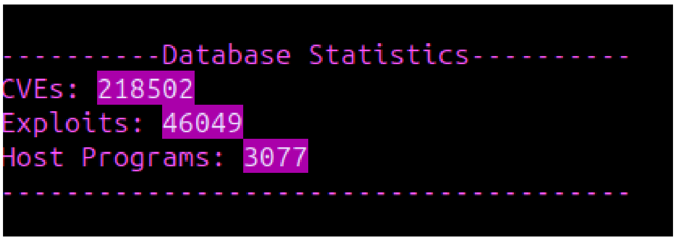
When this script runs, the database is populated with 218.502 CVEs. I have configured the script to run from 1/1/1990 until the date time that the script is executed



```
Inserted cve #CVE-2023-36345
Inserted cve #CVE-2023-36346
Inserted cve #CVE-2023-36348
Inserted cve #CVE-2023-3212
Inserted cve #CVE-2023-34254
Inserted cve #CVE-2023-35154
Inserted cve #CVE-2023-35163
Inserted cve #CVE-2023-35165
Inserted cve #CVE-2023-35169
Inserted cve #CVE-2023-35171
Inserted cve #CVE-2023-35172
Inserted cve #CVE-2023-35173
Inserted cve #CVE-2023-35927
Inserted cve #CVE-2023-35928
Inserted cve #CVE-2023-1783
Inserted cve #CVE-2023-35932
Inserted cve #CVE-2023-1721
Inserted cve #CVE-2023-1724
Inserted cve #CVE-2023-1722
Inserted cve #CVE-2023-3197
Inserted cve #CVE-2023-3387
Inserted cve #CVE-2023-3388
Inserted cve #CVE-2023-36612
Inserted cve #CVE-2023-36630
Inserted cve #CVE-2015-20109
Inserted cve #CVE-2023-36632
Inserted cve #CVE-2023-3396
Waiting 5 seconds...
437004 CVEs fetched from NVD API.
437004 CVEs imported into the database.
0 CVEs failed to import.
Script took 0:43:53.340047 to run.
```

In the screenshot here we can observe the ending of a ran of the import script. When done, the script informs the user how many CVEs are fetched and imported in the database as well as how many CVEs failed to import and the time it ran (~44 min in my case).

As you can see the database holds 218.502 CVEs (If we request the database statistics from the main menu, while the script reports 437.004.That is due to the fact that the CVE API returns all versions of each CVE, which means that the CVE is inserted multiple times in the database, while only the last version is finally in the database.

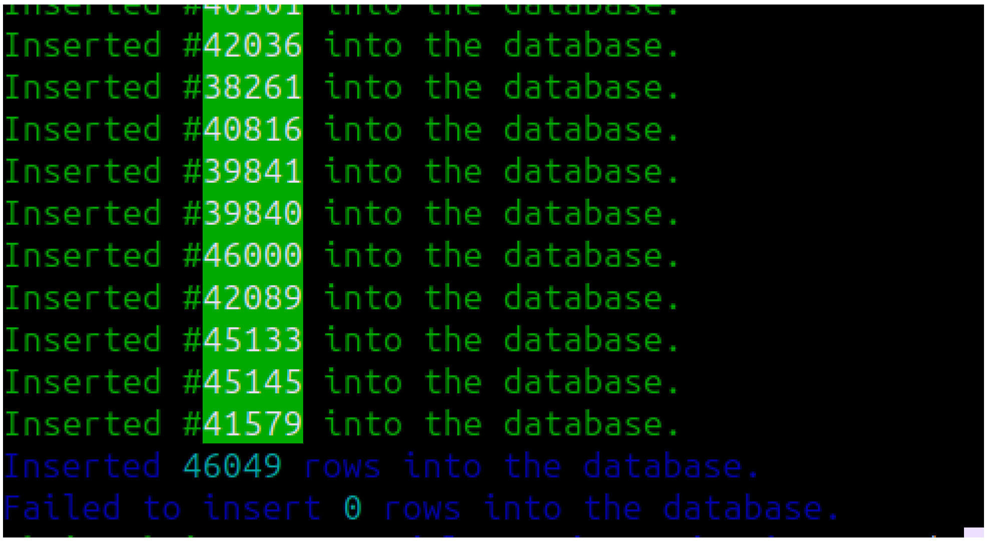


```
----------Database Statistics----------
CVEs: 218502
Exploits: 46049
Host Programs: 3077
---------------------------------------
```

Question 2

For the second question of the assignment I cloned the exploitdb repository to my computer and made a python script (option 2 in the main menu) to extract all contents of the .csv file containing the exploits and insert them in the database. The script can be found in the exploitdv_to_mySQL.py file.

When the script runs it returns the statistics of the ran, which ensures us that no exploit went missing during the import:

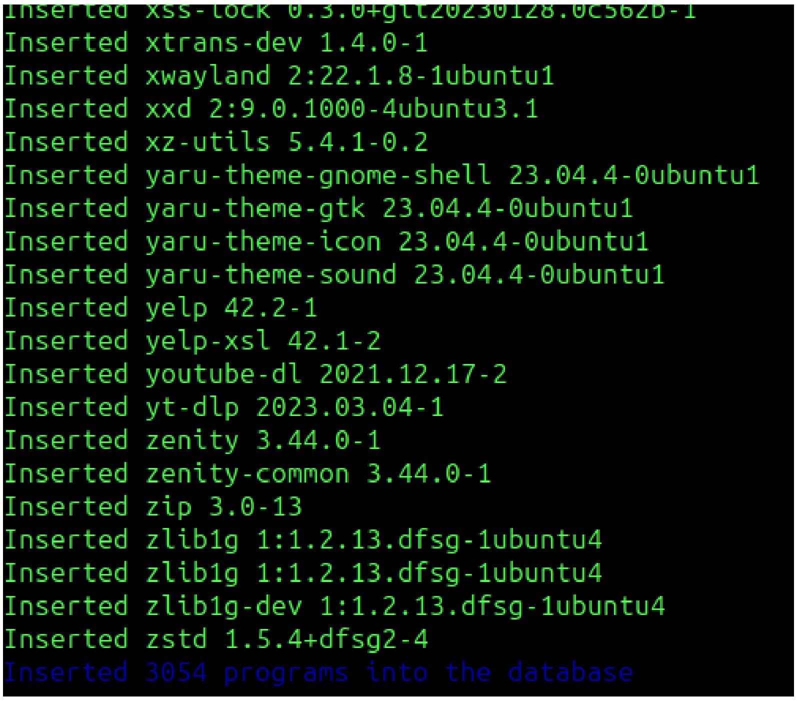


```
Inserted #42036 into the database.
Inserted #38261 into the database.
Inserted #40816 into the database.
Inserted #39841 into the database.
Inserted #39840 into the database.
Inserted #46000 into the database.
Inserted #42089 into the database.
Inserted #45133 into the database.
Inserted #45145 into the database.
Inserted #41579 into the database.
Inserted 46049 rows into the database.
Failed to insert 0 rows into the database.
```

Question 3

For this question I created a script (option 3 in the main menu) which runs the command "dpkg-query -W -f='${Package} ${Version} ${Status}\n" which returns a line containing the package name, the version and some status (that I hoped it contained the date). The script collects all that data and inserts all installed packages in the database.

As you can see in the script for this question (located in the file programs_to_mySQL.py) the database has a column for the installation date but unfortunately I was unable to figure out a way to collect this information and add it to the database (the result is this column to be Null for all entries).



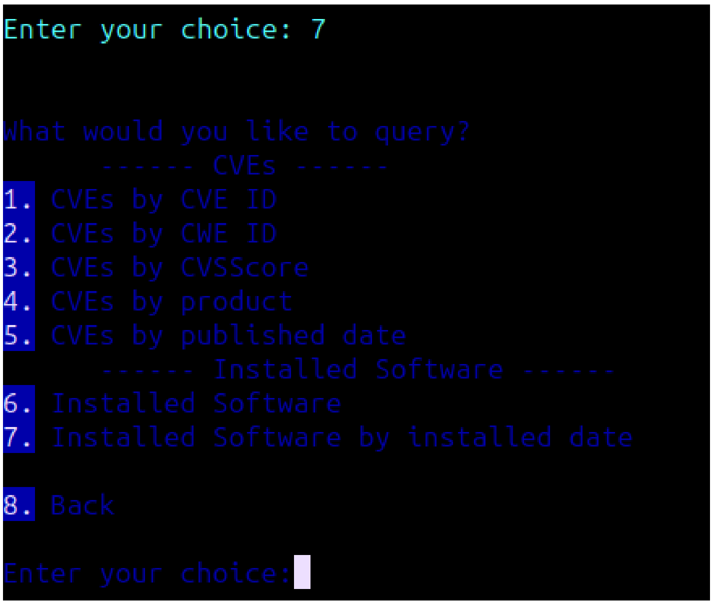
```
Inserted xss-lock 0.3.0+git20230128.0c562b-1
Inserted xtrans-dev 1.4.0-1
Inserted xwayland 2:22.1.8-1ubuntu1
Inserted xxd 2:9.0.1000-4ubuntu3.1
Inserted xz-utils 5.4.1-0.2
Inserted yaru-theme-gnome-shell 23.04.4-0ubuntu1
Inserted yaru-theme-gtk 23.04.4-0ubuntu1
Inserted yaru-theme-icon 23.04.4-0ubuntu1
Inserted yaru-theme-sound 23.04.4-0ubuntu1
Inserted yelp 42.2-1
Inserted yelp-xsl 42.1-2
Inserted youtube-dl 2021.12.17-2
Inserted yt-dlp 2023.03.04-1
Inserted zenity 3.44.0-1
Inserted zenity-common 3.44.0-1
Inserted zip 3.0-13
Inserted zlib1g 1:1.2.13.dfsg-1ubuntu4
Inserted zlib1g 1:1.2.13.dfsg-1ubuntu4
Inserted zlib1g-dev 1:1.2.13.dfsg-1ubuntu4
Inserted zstd 1.5.4+dfsg2-4
Inserted 3054 programs into the database
```

The end of output of this script can be seen in this screenshot. There were 3054 packages in my system and all of them (and their versions) were added in the database.

Question 4

For this question I had to make the tool execute the queries requested by the assignemt. The queries are split in the following categories: NVD API queries (query 1-5), Database Queries (query 6,7), combination of both (main menu option 8).

When one selects the 7th option from the main menu, a secondary menu pops up which can be seen in the screenshot below.

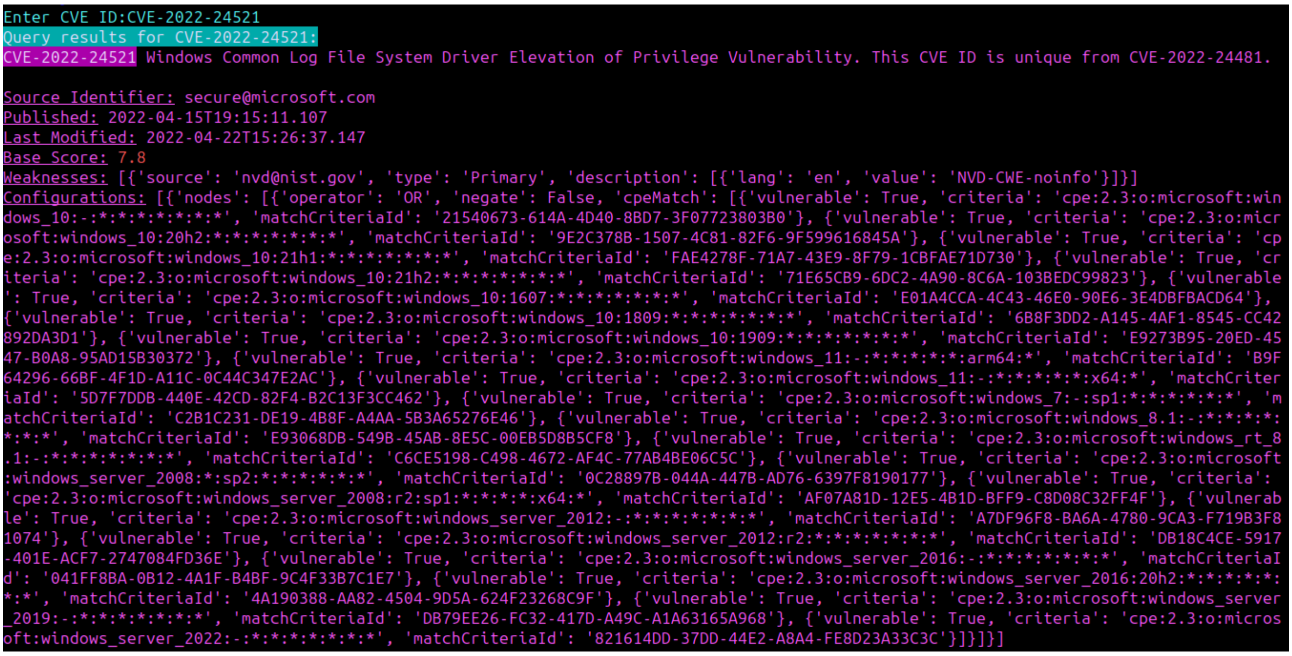


```
Enter your choice: 7


What would you like to query?
      ------ CVEs ------
1. CVEs by CVE ID
2. CVEs by CWE ID
3. CVEs by CVSScore
4. CVEs by product
5. CVEs by published date
      ------ Installed Software ------
6. Installed Software
7. Installed Software by installed date

8. Back

Enter your choice:
```

All queries are explained in the next page

Query 1: CVEs by CVE ID

For this query the script requires a CVE ID (like CVE-2022-24521) and then it queries the NVD API with the appropriate parameters to find the desired CVE. The CVE is then displayed with all its fields as well as a color coded severity (base) score.

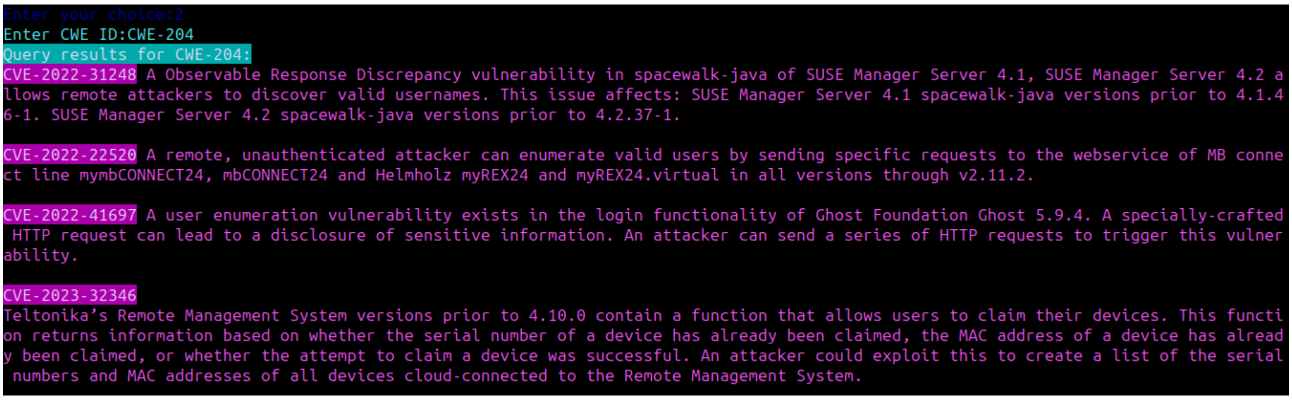


```
Enter CVE ID:CVE-2022-24521
Query results for CVE-2022-24521:
CVE-2022-24521 Windows Common Log File System Driver Elevation of Privilege Vulnerability. This CVE ID is unique from CVE-2022-24481.

Source Identifier: secure@microsoft.com
Published: 2022-04-15T19:15:11.107
Last Modified: 2022-04-22T15:26:37.147
Base Score: 7.8
Weaknesses: [{'source': 'nvd@nist.gov', 'type': 'Primary', 'description': [{'lang': 'en', 'value': 'NVD-CWE-noinfo'}]}]
Configurations: [{'nodes': [{'operator': 'OR', 'negate': False, 'cpeMatch': [{'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:win
dows_10:-:*:*:*:*:*:*:*', 'matchCriteriaId': '21540673-614A-4D40-8BD7-3F07723803B0'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:micr
osoft:windows_10:20h2:*:*:*:*:*:*:*', 'matchCriteriaId': '9E2C378B-1507-4C81-82F6-9F599616845A'}, {'vulnerable': True, 'criteria': 'cp
e:2.3:o:microsoft:windows_10:21h1:*:*:*:*:*:*:*', 'matchCriteriaId': 'FAE4278F-71A7-43E9-8F79-1CBFAE71D730'}, {'vulnerable': True, 'cr
iteria': 'cpe:2.3:o:microsoft:windows_10:21h2:*:*:*:*:*:*:*', 'matchCriteriaId': '71E65CB9-6DC2-4A90-8C6A-103BEDC99823'}, {'vulnerable
': True, 'criteria': 'cpe:2.3:o:microsoft:windows_10:1607:*:*:*:*:*:*:*', 'matchCriteriaId': 'E01A4CCA-4C43-46E0-90E6-3E4DBFBACD64'},
{'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_10:1809:*:*:*:*:*:*:*', 'matchCriteriaId': '6B8F3DD2-A145-4AF1-8545-CC42
892DA3D1'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_10:1909:*:*:*:*:*:*:*', 'matchCriteriaId': 'E9273B95-20ED-45
47-B0A8-95AD15B30372'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_11:-:*:*:*:*:*:arm64:*', 'matchCriteriaId': 'B9F
64296-66BF-4F1D-A11C-0C44C347E2AC'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_11:-:*:*:*:*:*:x64:*', 'matchCriter
iaId': '5D7F7DDB-440E-42CD-82F4-B2C13F3CC462'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_7:-:sp1:*:*:*:*:*:*', 'm
atchCriteriaId': 'C2B1C231-DE19-4B8F-A4AA-5B3A65276E46'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_8.1:-:*:*:*:*:
*:*:*', 'matchCriteriaId': 'E93068DB-549B-45AB-8E5C-00EB5D8B5CF8'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_rt_8
.1:-:*:*:*:*:*:*:*', 'matchCriteriaId': 'C6CE5198-C498-4672-AF4C-77AB4BE06C5C'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft
:windows_server_2008:*:sp2:*:*:*:*:*:*', 'matchCriteriaId': '0C28897B-044A-447B-AD76-6397F8190177'}, {'vulnerable': True, 'criteria':
'cpe:2.3:o:microsoft:windows_server_2008:r2:sp1:*:*:*:*:x64:*', 'matchCriteriaId': 'AF07A81D-12E5-4B1D-BFF9-C8D08C32FF4F'}, {'vulnerab
le': True, 'criteria': 'cpe:2.3:o:microsoft:windows_server_2012:-:*:*:*:*:*:*:*', 'matchCriteriaId': 'A7DF96F8-BA6A-4780-9CA3-F719B3F8
1074'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_server_2012:r2:*:*:*:*:*:*:*', 'matchCriteriaId': 'DB18C4CE-5917
-401E-ACF7-2747084FD36E'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_server_2016:-:*:*:*:*:*:*:*', 'matchCriteriaI
d': '041FF8BA-0B12-4A1F-B4BF-9C4F33B7C1E7'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_server_2016:20h2:*:*:*:*:*:
*:*', 'matchCriteriaId': '4A190388-AA82-4504-9D5A-624F23268C9F'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:microsoft:windows_server
_2019:-:*:*:*:*:*:*:*', 'matchCriteriaId': 'DB79EE26-FC32-417D-A49C-A1A63165A968'}, {'vulnerable': True, 'criteria': 'cpe:2.3:o:micros
oft:windows_server_2022:-:*:*:*:*:*:*:*', 'matchCriteriaId': '821614DD-37DD-44E2-A8A4-FE8D23A33C3C'}]}]}]
```

This query could also be executed in my mySQL database as all CVEs are also stored locally.

Query 2: CVEs by CWE ID

For this query I used the NVD API as well as all I had to do is to use the cweID parameter in the URL to get all CVEs associated with the given CWE id. The output of this query looks like this:

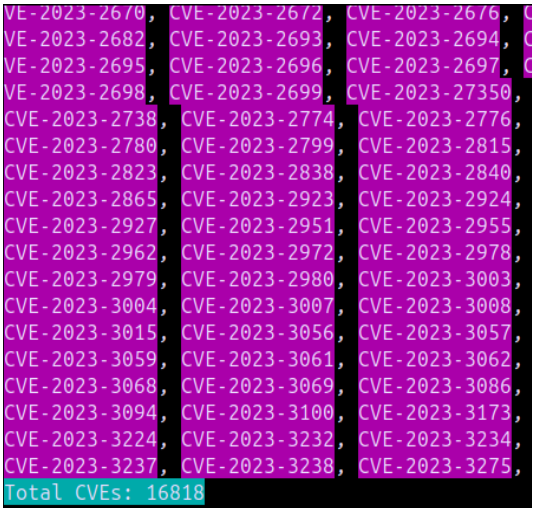


```
Enter your choice:2
Enter CWE ID:CWE-204
Query results for CWE-204:
CVE-2022-31248 A Observable Response Discrepancy vulnerability in spacewalk-java of SUSE Manager Server 4.1, SUSE Manager Server 4.2 a
llows remote attackers to discover valid usernames. This issue affects: SUSE Manager Server 4.1 spacewalk-java versions prior to 4.1.4
6-1. SUSE Manager Server 4.2 spacewalk-java versions prior to 4.2.37-1.

CVE-2022-22520 A remote, unauthenticated attacker can enumerate valid users by sending specific requests to the webservice of MB conne
ct line mymbCONNECT24, mbCONNECT24 and Helmholz myREX24 and myREX24.virtual in all versions through v2.11.2.

CVE-2022-41697 A user enumeration vulnerability exists in the login functionality of Ghost Foundation Ghost 5.9.4. A specially-crafted
 HTTP request can lead to a disclosure of sensitive information. An attacker can send a series of HTTP requests to trigger this vulner
ability.

CVE-2023-32346
Teltonika’s Remote Management System versions prior to 4.10.0 contain a function that allows users to claim their devices. This functi
on returns information based on whether the serial number of a device has already been claimed, the MAC address of a device has alread
y been claimed, or whether the attempt to claim a device was successful. An attacker could exploit this to create a list of the serial
 numbers and MAC addresses of all devices cloud-connected to the Remote Management System.
```

Query 3: CVEs by CVSScore

For this query the user is required to give a float number for the search score. The script then queries the database for CVEs with grater or equal base score than the requested one.

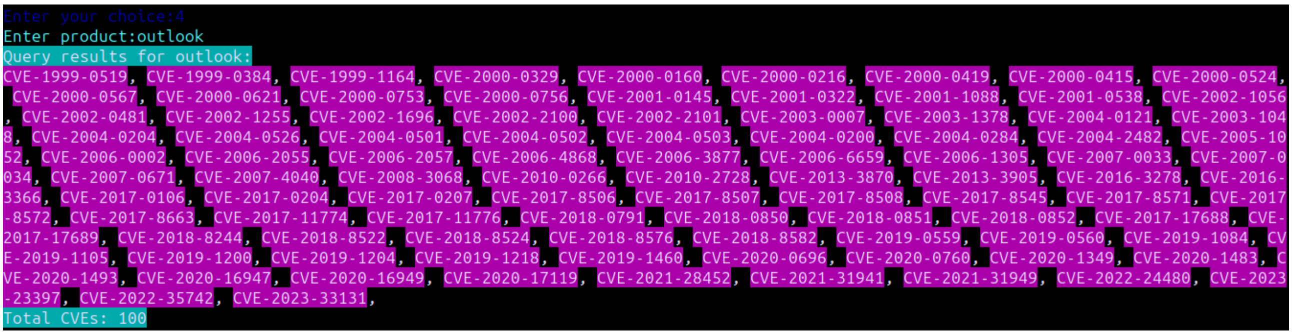
The result of the query can be seen in the screenshot on the right. The CVEs are displayed in a short format to prevent the flooding of the output (although they already flood the output).



```
VE-2023-2670, CVE-2023-2672, CVE-2023-2676, C
VE-2023-2682, CVE-2023-2693, CVE-2023-2694, C
VE-2023-2695, CVE-2023-2696, CVE-2023-2697, C
VE-2023-2698, CVE-2023-2699, CVE-2023-27350,
CVE-2023-2738, CVE-2023-2774, CVE-2023-2776,
CVE-2023-2780, CVE-2023-2799, CVE-2023-2815,
CVE-2023-2823, CVE-2023-2838, CVE-2023-2840,
CVE-2023-2865, CVE-2023-2923, CVE-2023-2924,
CVE-2023-2927, CVE-2023-2951, CVE-2023-2955,
CVE-2023-2962, CVE-2023-2972, CVE-2023-2978,
CVE-2023-2979, CVE-2023-2980, CVE-2023-3003,
CVE-2023-3004, CVE-2023-3007, CVE-2023-3008,
CVE-2023-3015, CVE-2023-3056, CVE-2023-3057,
CVE-2023-3059, CVE-2023-3061, CVE-2023-3062,
CVE-2023-3068, CVE-2023-3069, CVE-2023-3086,
CVE-2023-3094, CVE-2023-3100, CVE-2023-3173,
CVE-2023-3224, CVE-2023-3232, CVE-2023-3234,
CVE-2023-3237, CVE-2023-3238, CVE-2023-3275,
Total CVEs: 16818
```

Query 4: CVEs by product

For the implementation of this query I first had to find a way to create the CPE string for the API call. After some research I came up with the string “cpe:2,5:*:*:{product}:{version}”. If the version was not specified, the version variable would be set as another *.

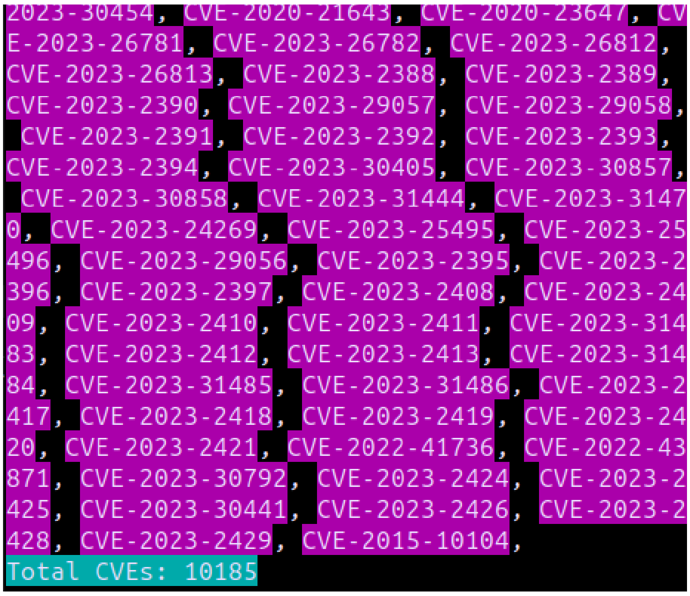


```
Enter your choice:4
Enter product:outlook
Query results for outlook:
CVE-1999-0519, CVE-1999-0384, CVE-1999-1164, CVE-2000-0329, CVE-2000-0160, CVE-2000-0216, CVE-2000-0419, CVE-2000-0415, CVE-2000-0524,
 CVE-2000-0567, CVE-2000-0621, CVE-2000-0753, CVE-2000-0756, CVE-2001-0145, CVE-2001-0322, CVE-2001-1088, CVE-2001-0538, CVE-2002-1056
, CVE-2002-0481, CVE-2002-1255, CVE-2002-1696, CVE-2002-2100, CVE-2002-2101, CVE-2003-0007, CVE-2003-1378, CVE-2004-0121, CVE-2003-104
8, CVE-2004-0204, CVE-2004-0526, CVE-2004-0501, CVE-2004-0502, CVE-2004-0503, CVE-2004-0200, CVE-2004-0284, CVE-2004-2482, CVE-2005-10
52, CVE-2006-0002, CVE-2006-2055, CVE-2006-2057, CVE-2006-4868, CVE-2006-3877, CVE-2006-6659, CVE-2006-1305, CVE-2007-0033, CVE-2007-0
034, CVE-2007-0671, CVE-2007-4040, CVE-2008-3068, CVE-2010-0266, CVE-2010-2728, CVE-2013-3870, CVE-2013-3905, CVE-2016-3278, CVE-2016-
3366, CVE-2017-0106, CVE-2017-0204, CVE-2017-0207, CVE-2017-8506, CVE-2017-8507, CVE-2017-8508, CVE-2017-8545, CVE-2017-8571, CVE-2017
-8572, CVE-2017-8663, CVE-2017-11774, CVE-2017-11776, CVE-2018-0791, CVE-2018-0850, CVE-2018-0851, CVE-2018-0852, CVE-2017-17688, CVE-
2017-17689, CVE-2018-8244, CVE-2018-8522, CVE-2018-8524, CVE-2018-8576, CVE-2018-8582, CVE-2019-0559, CVE-2019-0560, CVE-2019-1084, CV
E-2019-1105, CVE-2019-1200, CVE-2019-1204, CVE-2019-1218, CVE-2019-1460, CVE-2020-0696, CVE-2020-0760, CVE-2020-1349, CVE-2020-1483, C
VE-2020-1493, CVE-2020-16947, CVE-2020-16949, CVE-2020-17119, CVE-2021-28452, CVE-2021-31941, CVE-2021-31949, CVE-2022-24480, CVE-2023
-23397, CVE-2022-35742, CVE-2023-33131,
Total CVEs: 100
```

Query 5: CVEs by published date

For this querry I used the fields pubStartDate and pubEndDate to query the NVD API. Also I made sure to pull all the TotalResults as NVD allows only 2000 results per page.

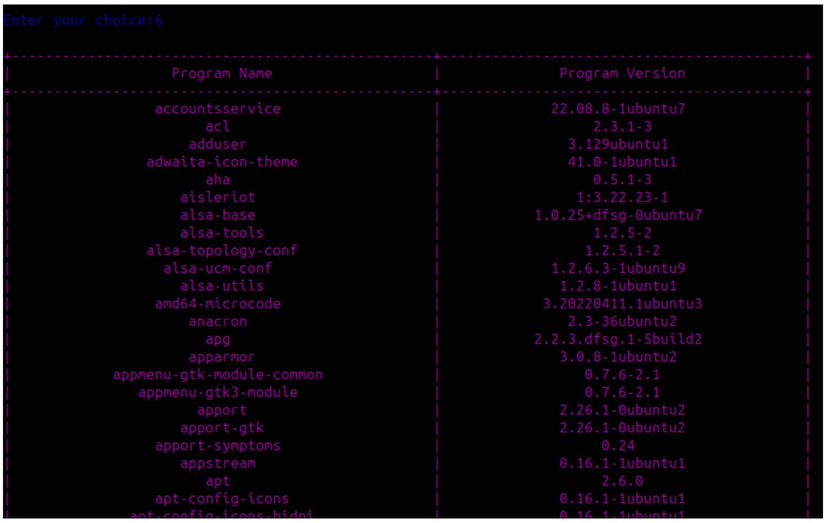
As the end date the user can also use the keyword “now” to pull all CVEs until the time the script is executed. The result looks like the previous two queries.



```
2023-30454, CVE-2020-21643, CVE-2020-23647, CV
E-2023-26781, CVE-2023-26782, CVE-2023-26812,
CVE-2023-26813, CVE-2023-2388, CVE-2023-2389,
CVE-2023-2390, CVE-2023-29057, CVE-2023-29058,
 CVE-2023-2391, CVE-2023-2392, CVE-2023-2393,
CVE-2023-2394, CVE-2023-30405, CVE-2023-30857,
 CVE-2023-30858, CVE-2023-31444, CVE-2023-3147
0, CVE-2023-24269, CVE-2023-25495, CVE-2023-25
496, CVE-2023-29056, CVE-2023-2395, CVE-2023-2
396, CVE-2023-2397, CVE-2023-2408, CVE-2023-24
09, CVE-2023-2410, CVE-2023-2411, CVE-2023-314
83, CVE-2023-2412, CVE-2023-2413, CVE-2023-314
84, CVE-2023-31485, CVE-2023-31486, CVE-2023-2
417, CVE-2023-2418, CVE-2023-2419, CVE-2023-24
20, CVE-2023-2421, CVE-2022-41736, CVE-2022-43
871, CVE-2023-30792, CVE-2023-2424, CVE-2023-2
425, CVE-2023-30441, CVE-2023-2426, CVE-2023-2
428, CVE-2023-2429, CVE-2015-10104,
Total CVEs: 10185
```

Query 6: Installed Software

This query was as simple as pulling all data from the installed programs table of my database. I used the library “tabulate” to display a better looking table of the contents.



```
Enter your choice:6

+------------------------------------------+------------------------------------------+
|               Program Name               |             Program Version              |
+------------------------------------------+------------------------------------------+
|            accountsservice               |            22.08.8-1ubuntu7              |
|                  acl                     |                2.3.1-3                   |
|                adduser                   |             3.129ubuntu1                 |
|          adwaita-icon-theme              |             41.0-1ubuntu1                |
|                  aha                     |                0.5.1-3                   |
|               aisleriot                  |              1:3.22.23-1                 |
|               alsa-base                  |        1.0.25+dfsg-0ubuntu7              |
|               alsa-tools                 |                1.2.5-2                   |
|          alsa-topology-conf              |               1.2.5.1-2                  |
|             alsa-ucm-conf                |           1.2.6.3-1ubuntu9               |
|               alsa-utils                 |            1.2.8-1ubuntu1                |
|            amd64-microcode               |        3.20220411.1ubuntu3               |
|                anacron                   |             2.3-36ubuntu2                |
|                  apg                     |         2.2.3.dfsg.1-5build2             |
|               apparmor                   |            3.0.8-1ubuntu2                |
|       appmenu-gtk-module-common          |               0.7.6-2.1                  |
|          appmenu-gtk3-module             |               0.7.6-2.1                  |
|                apport                    |            2.26.1-0ubuntu2               |
|               apport-gtk                 |            2.26.1-0ubuntu2               |
|            apport-symptoms               |                 0.24                     |
|               appstream                  |            0.16.1-1ubuntu1               |
|                  apt                     |                 2.6.0                    |
|            apt-config-icons              |            0.16.1-1ubuntu1               |
|         apt-config-icons-hidpi           |            0.16.1-1ubuntu1               |
```

Query 7: Installed Software by date

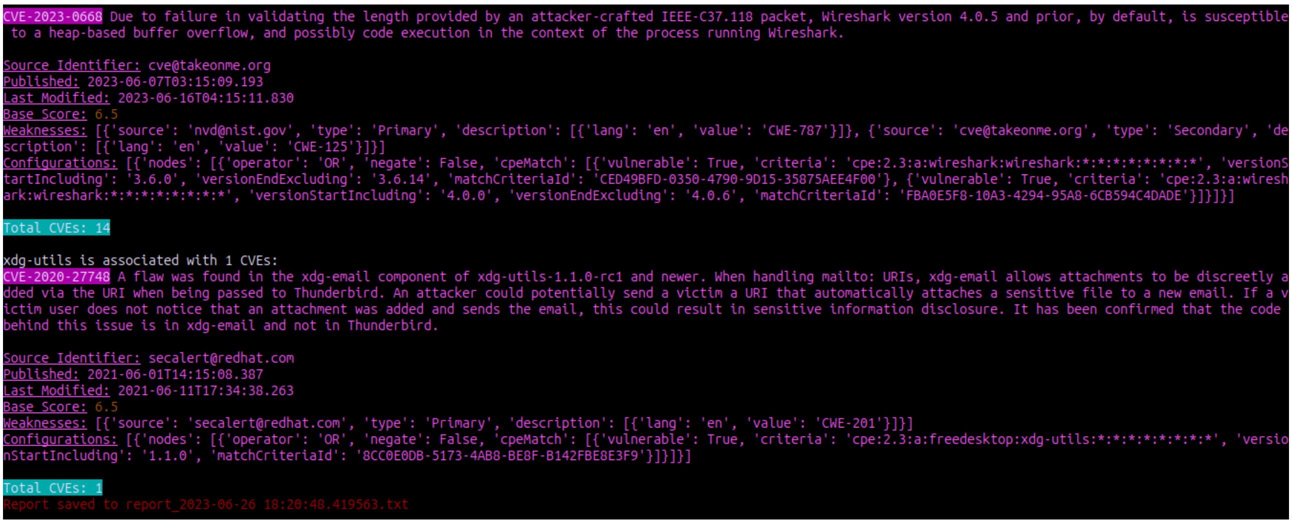
I didn’t manage to complete the functionality of this query as I was unable to collect the published date of each installed program through *dpkg*. If I was able to get my hands on this information it would be as simple as inserting it together with the version and the package name (the column at the database exists) and then querying for the requested dates

Main Menu Opt8: Generate Report for Vulnerable Installed Programs

This was the biger part of this assignment and takes a lot time to run. For the implementation of this report I had to go through these steps: First get all the installed programs, then querry the NVD API for each program using its CPE (name and version) and if a CVE is related to the program and its version, collect it to a dictionary.

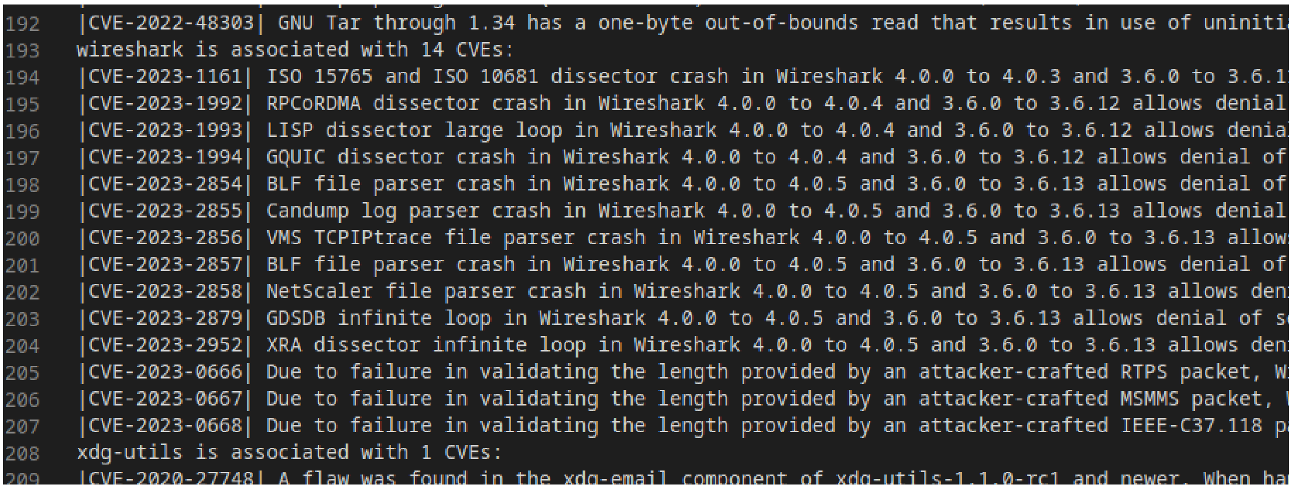
After the search is complete and if there are any CVEs at all, the script saves the report at a file named “report_{datetime}.txt”. Also the output is displayed in the terminal with each cve having its base score color coded based on the severity of the score.

The ending of the output of the report can be seen in the screenshot below:



```
CVE-2023-0668 Due to failure in validating the length provided by an attacker-crafted IEEE-C37.118 packet, Wireshark version 4.0.5 and prior, by default, is susceptible
 to a heap-based buffer overflow, and possibly code execution in the context of the process running Wireshark.

Source Identifier: cve@takeonme.org
Published: 2023-06-07T03:15:09.193
Last Modified: 2023-06-16T04:15:11.830
Base Score: 6.5
Weaknesses: [{'source': 'nvd@nist.gov', 'type': 'Primary', 'description': [{'lang': 'en', 'value': 'CWE-787'}]}, {'source': 'cve@takeonme.org', 'type': 'Secondary', 'de
scription': [{'lang': 'en', 'value': 'CWE-125'}]}]
Configurations: [{'nodes': [{'operator': 'OR', 'negate': False, 'cpeMatch': [{'vulnerable': True, 'criteria': 'cpe:2.3:a:wireshark:wireshark:*:*:*:*:*:*:*:*', 'versionS
tartIncluding': '3.6.0', 'versionEndExcluding': '3.6.14', 'matchCriteriaId': 'CED49BFD-0350-4790-9D15-35875AEE4F00'}, {'vulnerable': True, 'criteria': 'cpe:2.3:a:wiresh
ark:wireshark:*:*:*:*:*:*:*:*', 'versionStartIncluding': '4.0.0', 'versionEndExcluding': '4.0.6', 'matchCriteriaId': 'FBA0E5F8-10A3-4294-95A8-6CB594C4DADE'}]}]}]

Total CVEs: 14

xdg-utils is associated with 1 CVEs:
CVE-2020-27748 A flaw was found in the xdg-email component of xdg-utils-1.1.0-rc1 and newer. When handling mailto: URIs, xdg-email allows attachments to be discreetly a
dded via the URI when being passed to Thunderbird. An attacker could potentially send a victim a URI that automatically attaches a sensitive file to a new email. If a v
ictim user does not notice that an attachment was added and sends the email, this could result in sensitive information disclosure. It has been confirmed that the code
behind this issue is in xdg-email and not in Thunderbird.

Source Identifier: secalert@redhat.com
Published: 2021-06-01T14:15:08.387
Last Modified: 2021-06-11T17:34:38.263
Base Score: 6.5
Weaknesses: [{'source': 'secalert@redhat.com', 'type': 'Primary', 'description': [{'lang': 'en', 'value': 'CWE-201'}]}]
Configurations: [{'nodes': [{'operator': 'OR', 'negate': False, 'cpeMatch': [{'vulnerable': True, 'criteria': 'cpe:2.3:a:freedesktop:xdg-utils:*:*:*:*:*:*:*:*', 'versio
nStartIncluding': '1.1.0', 'matchCriteriaId': '8CC0E0DB-5173-4AB8-BE8F-B142FBE8E3F9'}]}]}]

Total CVEs: 1
Report saved to report_2023-06-26 18:20:48.419563.txt
```

In addition here is a part of the file produced by the script. I would give the report file with the other submitted files but I think it is wiser not to give away all of my computer’s vulnerabilities (although after this long output I am going to take my measures so that this list is reduced)



```
 |CVE-2022-48303| GNU Tar through 1.34 has a one-byte out-of-bounds read that results in use of uniniti
wireshark is associated with 14 CVEs:
 |CVE-2023-1161| ISO 15765 and ISO 10681 dissector crash in Wireshark 4.0.0 to 4.0.3 and 3.6.0 to 3.6.1
 |CVE-2023-1992| RPCoRDMA dissector crash in Wireshark 4.0.0 to 4.0.4 and 3.6.0 to 3.6.12 allows denial
 |CVE-2023-1993| LISP dissector large loop in Wireshark 4.0.0 to 4.0.4 and 3.6.0 to 3.6.12 allows denia
 |CVE-2023-1994| GQUIC dissector crash in Wireshark 4.0.0 to 4.0.4 and 3.6.0 to 3.6.12 allows denial of
 |CVE-2023-2854| BLF file parser crash in Wireshark 4.0.0 to 4.0.5 and 3.6.0 to 3.6.13 allows denial of
 |CVE-2023-2855| Candump log parser crash in Wireshark 4.0.0 to 4.0.5 and 3.6.0 to 3.6.13 allows denial
 |CVE-2023-2856| VMS TCPIPtrace file parser crash in Wireshark 4.0.0 to 4.0.5 and 3.6.0 to 3.6.13 allow
 |CVE-2023-2857| BLF file parser crash in Wireshark 4.0.0 to 4.0.5 and 3.6.0 to 3.6.13 allows denial of
 |CVE-2023-2858| NetScaler file parser crash in Wireshark 4.0.0 to 4.0.5 and 3.6.0 to 3.6.13 allows den
 |CVE-2023-2879| GDSDB infinite loop in Wireshark 4.0.0 to 4.0.5 and 3.6.0 to 3.6.13 allows denial of s
 |CVE-2023-2952| XRA dissector infinite loop in Wireshark 4.0.0 to 4.0.5 and 3.6.0 to 3.6.13 allows den
 |CVE-2023-0666| Due to failure in validating the length provided by an attacker-crafted RTPS packet, W
 |CVE-2023-0667| Due to failure in validating the length provided by an attacker-crafted MSMMS packet,
 |CVE-2023-0668| Due to failure in validating the length provided by an attacker-crafted IEEE-C37.118 p
xdg-utils is associated with 1 CVEs:
 |CVE-2020-27748| A flaw was found in the xdg-email component of xdg-utils-1.1.0-rc1 and newer. When ha
```

Also the credentials to the database I used are replaced by dummy ones so if you intend to run my tool, you need to set it up considering your configuration’s credentials