Computer Science Department
University of Crete

CS-455: Internet Attacks
Spring 2022 - 2023

Assignment 7
Vulnerability Assessment

Uploaded: 17/05/2023
Deadline: 25/06/2023

Introduction

In this assignment, you will scan your PC (or a VM) for vulnerabilities and report related CVEs, CWEs and exploits. To initiate the process, you will have to download all known CVEs and exploits into a database and then run certain queries. In each question think of ways to verify the correctness of your results and record the output.

Questions

1. **[20%] Step 1: Install a MySQL database and import all CVEs**
    - Download site: https://www.cve.org/Downloads#current-format
        - Size: 280MB (1.3GB unzipped), ~200K CVEs
    - References: https://github.com/CVEProject/cve-schema
    - You may import a small number of CVEs for a start to test some functionality and later import all CVEs to test the robustness of your implementation
    - You may use MySQL Workbench to view the contents of the database
    - The queries listed below may be demonstrated inside the Workbench or via a python front-end
    - Alternative ways to download CVEs / query online CVE databases
        - https://nvd.nist.gov/developers
    - Report: Describe the table created and show evidence the import was successful and verified (e.g., no CVEs are missing)

2. **[20%] Step 2: Download known exploits and import them into the db**
    - Use python scripts to extract data from the downloaded files (e.g., .xlsx) and then import it into the database
    - https://www.exploit-db.com/

- https://gitlab.com/exploit-database/exploitdb/
- Report: Describe the table created and show evidence the import was successful and verified (e.g., no exploits are missing)

3. **[20%] Step 3: Scan for installed apps and import this data into the db**
   - Assume the target is your operating system
   - Use python scripts to extract data from your OS and then import it into the database
   - Alternatively, you may extract information about programs installed in a (Windows/Linux) VM and import that data into the db running on your host OS. In this scenario you have the option to install vulnerable applications in the VM.
   - Report: Describe the table created and show evidence the import was successful and verified (e.g., no installed apps are missing)

4. **[20%] Step 4: Return results for the following queries:**
   - Return all CVEs based on a CWE id
   - Return all CVEs with cvss_v3 > x (e.g., 7)
   - Return all CVEs related to a product (e.g., Firefox)
     - Only CVEs with CPEs should be considered
   - Return all CVEs published in a particular time range
   - Return all software installed in the target OS
   - Return all software installed in the target OS in a particular time range
   - Return all vulnerable software installed, together with the relevant CVEs and exploits
     - Color code the output
     - https://www.cvedetails.com/cvss-score-distribution.php
     - Alternatively, use a simpler color coding (i.e., less colors). See for example, slide 17 in 2223-CS455-17-Pentesting-Reporting.pdf
       1. Critical: Red
       2. High: Orange
       3. Moderate: Yellow
       4. Low: Green
       5. Informational: Blue
     - If your system is secure no results will be returned
     - You will have to install some dummy apps (hot OS or VM) to demonstrate the required functionality
   - Report: Record the query and sample output (in some cases the output will be long). In all cases record totals (e.g., #CVEs returned). Describe the table created and show evidence the results are valid (e.g., querying NVD returns the same results)

5. **[20%] Step 5: OWASP A02_2021: Cryptographic Failures**
    - Consider the following list of relevant CWEs/CVEs
        - https://owasp.org/Top10/A02_2021-Cryptographic_Failures/
        - Only 29 CWEs (out of 400 – 600) are relevant
    - Run the Q4 queries only against the specific range of CVEs
    - This way, you will be able to identify apps running on your system, susceptible to cryptography-related vulnerabilities
    - Report: Record the query and sample output (in some cases the output will be long). In all cases record totals (e.g., #CVEs returned). Describe the table created and show evidence the results are valid (e.g., querying NVD returns the same results)

6. **[10%] BONUS: Automation**
    - Write a python script to periodically update the database with CVEs and exploits
        - For example, daily or weekly
    - Write a script to periodically scan your system for vulnerabilities (based on CVEs stored in the database)
    - Scripts: submit separately
    - Report: Include a README for the script as well as evidence of their correct behavior

References

- Open-Source CVE Monitoring & Management: Cutting Through the Vulnerability Storm
    - https://www.youtube.com/watch?v=cCzb0lewVj4&ab_channel=TheLinuxFoundation
- 7 Ways To Generate a List of Installed Programs in Windows
    - https://helpdeskgeek.com/how-to/generate-a-list-of-installed-programs-in-windows/

Submission

- Submit a pdf document including all the information requested above.
- Submissions will be done through eLearn.
- This assignment is an individual creative process and students must submit their own work. You are not allowed, under any circumstances, to copy another person's work. You must also ensure that your work won't be accessible to others.
- You are encouraged to post any questions you may have in the eLearn forum. If, however you believe that your question contains part of the solution or spoilers for the other students you can communicate directly with the course staff at hy455@csd.uoc.gr.

Disclaimer

It is **ILLEGAL** to attempt any type of attack against a target (individual, network, public/private entity, etc.) without authorization / explicit permission. Such actions are **prohibited and punished by law and university policies**.