**Advanced Executive Program in Cybersecurity**

Virtual Internship Project Problem Statement



**Malware Analyst**

**Problem statement:**

You are working as a malware analyst for El Banco Bank, where your primary responsibility is to secure the bank's assets by examining, identifying, and understanding malware, such as viruses, worms, bots, rootkit, ransomware, and Trojan horse. These types of malware can infect systems by exploiting vulnerabilities and cause them to behave in unexpected ways.

**Background of the problem statement:**

El Banco Bank is one of the fastest growing banks in Europe with more than 1200 branches across the country and manages €200 billion in assets.

Handling millions of dollars of banking transactions per day, its customers hugely depend upon the security of their banking data. The recent surge in cyber-attacks and data breaches has become a significant issue for every organization.

According to the latest reports, 51% of cyberattacks are due to various malware, such as viruses, rootkit, trojan horse, and ransomware.

**Expected deliverables:**

**TASK 1:**

As a malware analyst, you have to examine suspicious files or URLs and detect any malware threats. You have been provided a list of files that you need to examine and verify if these files are real and do not contain anything malicious. You can check the digital signatures of the files to verify if it is authentic and hasn’t been tampered with.

For the following applications, determine the Signer Name and the Digest Algorithm used in the digital signatures. If the digital signature is not available, leave the fields blank.

|  |  |  |
| --- | --- | --- |
|  | **Name of Signer** | **Digest Algorithm** |
| **Virtualbox** | Oracle Corporation | Sha1, Sha256 |
| **LibreOffice** | The Document Foundation | Sha256 |
| **OWASP ZAP** | NA | NA |
| **Wireshark** | Wireshark Foundation | Sha256 |

**TASK 2:**

If the digital signature of the files is not available, you can still verify the integrity of the file by comparing the hash values provided for the original files. For those files that cannot be verified using digital signatures, use the following resources to compare the SHA256 values of the files and determine if the given digest and the calculated digest value match.

By comparing the files' hash values, you are able to determine the integrity of the files and be assured that the downloaded files are authentic and haven't been tampered with.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Given Digest value** | **Calculated Digest value** | **Match?** |
| **Virtualbox** | 2fb44977d3329e55e8b61  408ab4af5239ecd3d80c5990fb5cd6bd0c91a854d62 | 2FB44977D3329E55E8B61408AB4AF5239ECD3D80C5990FB5CD6BD0C91A854D62 | YES |
| **LibreOffice** | Sha256:  A7251bca75603c3075477d090e17931dbeff32eb18808421d540fe5b21497a5f  Sha1:  8a7b6d2077e3148480f460f27ef4630570ad0524 | Sha256:  A7251BCA75603C3075477D090E17931DBEFF32EB18808421D540FE5B21497A5F | YES |
| **OWASP ZAP** | Df49ffbd14cf82cde5ac06902615e40cbfce1576f866436366708c0845eb9ec6 | DF49FFBD14CF82CDEAC06902615E40CBFCE1576F8664366708C0845EB9EC6 | YES |
| **Wireshark** | B621718ffe64748590ea9568fbbed0f3d86b0939906dc9f7fe064e20ce385492 | B621718FFE64748590EA9568FBBED0F3D86B0939906DC9F7FE064E20CE385492 | YES |

**Resources for SHA256 values:**

1. <https://raw.githubusercontent.com/zaproxy/zap-admin/master/ZapVersions-2.11.xml>
2. <https://www.virtualbox.org/download/hashes/6.1.30/SHA256SUMS>
3. <https://download.documentfoundation.org/libreoffice/stable/7.2.3/win/x86_64/LibreOffice_7.2.3_Win_x64.msi.mirrorlist>
4. https://www.wireshark.org/download/SIGNATURES-3.6.0.txt

**TASK 3:**

Analyzing files to understand the associated threats is an increasingly important skill for malware analysts. Analyzing malware could be a daunting task. Fortunately, there are many tools and resources at your disposal that could help you make this task a little bit easier.

Your next task is to determine if the files are malicious or not.

**Link to download the malwares:** https://github.com/Simplilearn-Edu/Advanced-Executive-Program-in-Cybersecurity

|  |  |
| --- | --- |
| **File** | **Malware?** |
| 1. **Keylogger** | NO |
| 1. **Ransomware** | NO |
| 1. **Exeinfope** | NO |

**Link for analyzing malicious files:** <https://www.virustotal.com/>

**TASK 4:**

Another important task for a malware analyst is to perform**a vulnerability** assessment to**identify** the most critical**vulnerabilities** for correction. This will reduce the risk of**hackers** exploiting the applications.

Your organization uses GLPI, an open-source IT Asset Management, issue tracking system, and service desk system written on PHP. GLPI uses a barcode plugin used for printing barcodes and QR codes.

|  |  |  |
| --- | --- | --- |
|  | **Version** | **Link** |
| **GLPI** | 9.5.5 | <https://glpi-project.org/> |
| **Barcode GLPI plugin** | 2.6.0 | <https://github.com/pluginsGLPI/barcode> |

Use the NVD database to search for vulnerabilities in GLPI and third-party plugins (minimum 5 vulnerabilities) and suggest a fix or a workaround.

**Link for NVD Database:** <https://nvd.nist.gov/>

|  |  |  |  |
| --- | --- | --- | --- |
| **CVE** | **Description** | **CVSS Severity** | **Remediation** |
| CVE-2021-43778CVE-2023-28838CVE-2023-28852CVE-2023-23610CVE-2022-21720 | the barcode plugin installed are vulnerable to a path traversal vulnerability.  a SQL Injection vulnerability allow users with access rights to statistics or reports to extract all data from database and, in some cases, write a webshell on the server.  a user with dashboard administration rights may hack the dashboard form to store malicious code that will be executed when other users will use the related dashboard.  vulnerable to Improper Privilege Management. Any user having access to the standard interface can export data of almost any GLPI item type, even those on which user is not allowed to access (including assets, tickets, users, ...)  an entity administrator is capable of retrieving normally inaccessible data via SQL injection | **Base Score:** [**5.0 MEDIUM**](https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator?name=CVE-2021-43778&vector=(AV:N/AC:L/Au:N/C:P/I:N/A:N)&version=2.0&source=NIST)  **Vector:**  (AV:N/AC:L/Au:N/C:P/I:N/A:N)  **CVSS 3.x Severity and Metrics:**  **NIST:** NVD  **Base Score:** [**8.1 HIGH**](https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?name=CVE-2023-28838&vector=AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N&version=3.1&source=NIST)  **Vector:**  CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N  **Base Score:** [**4.8 MEDIUM**](https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?name=CVE-2023-28852&vector=AV:N/AC:L/PR:H/UI:R/S:C/C:L/I:L/A:N&version=3.1&source=GitHub,%20Inc.)  **Vector:**  CVSS:3.1/AV:N/AC:L/PR:H/UI:R/S:C/C:L/I:L/A:N  **Base Score:** [**6.5 MEDIUM**](https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?name=CVE-2023-23610&vector=AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N&version=3.1&source=GitHub,%20Inc.)  **Vector:**  CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N  **Base Score:** [**4.9 MEDIUM**](https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?name=CVE-2022-21720&vector=AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N&version=3.1&source=GitHub,%20Inc.)  **Vector:**  CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N | As a workaround, delete the `front/send.php`  File. **References to Advisories, Solutions, and Tools** <https://github.com/pluginsGLPI/barcode/commit/428c3d9adfb446e8492b1c2b7affb3d34072ff46>  a patch for this issue. As a workaround, remove `Assistance > Statistics` and `Tools > Reports` read rights from every user.   | **Hyperlink** | **Resource** | | --- | --- | | <https://github.com/glpi-project/glpi/releases/tag/10.0.7> | **Patch**  **Release Notes** |   <https://github.com/glpi-project/glpi/releases/tag/9.5.13>   | **Hyperlink** | **Resource** | | --- | --- | | <https://github.com/glpi-project/glpi/security/advisories/GHSA-6565-hm87-24hf> |  |   <https://github.com/glpi-project/glpi/releases/tag/9.5.7> |