**CYB101 Project 5 (🔗** [**Instructions Page**](https://courses.codepath.org/courses/cyb101/unit/5#!projects)**)**

👤 Student Name: Sila Tamang

✉️ Student Email: stamang3@uco.edu

**Reflection (Required)**

| **🤔 Reflection Question #1:** If I had to **explain “how is malware detected?” in 3 emojis,** they would be…  (Feel free to put other comments about your experience this unit here, too!) |
| --- |
| 😇🧐🤔 |

| **👾Reflection Question #2:** If someone sent you an unknown file, how would you go about checking if it contains a virus? |
| --- |
| If someone sends me an unknown file, here are following steps I will do to check if it contains a virus:   1. Use an antivirus software: The first and most important step is to have an updated antivirus software installed on the computer. Most antivirus software automatically scans every file that I download and will flag any file that contains a virus. 2. Scan the file with the antivirus software: Using an antivirus software installed, I can right-click on the file and select the option to scan the file. This will launch my antivirus software, which will scan the file for viruses. 3. Check the file extension: Most viruses are spread through executable files, which have extensions like .exe, .bat, .vbs, .js, .wsf, .msi, and .scr. If the file I received has one of these extensions, it is more likely to be a virus. However, some viruses can also be hidden in other file types, such as image or document files, so it is always best to scan the file with my antivirus software. 4. Use online virus scanners: I can also use online virus scanners to check if a file contains a virus. Some popular online virus scanners include VirusTotal and Jotti's malware scan. Simply upload the file to the website, and it will be scanned by multiple antivirus engines. 5. Use a sandbox: If I am still unsure about the file's safety, I can use a sandbox to run the file in a virtual environment. This will allow me to test the file without actually running it on my computer. Some popular sandbox tools include Sandboxie and Windows Sandbox. |

| **📣 Shoutouts:** Share appreciation for anyone who helped you out with this project or made your day a little better! |
| --- |
| The CodePath Team and team 52 mentor |

**Required Challenge Screenshots (Required)**

Use the answer boxes below to paste in your screenshots from completing the project. Clarifying notes are optional.

(You don’t need any screenshots for **Part 1** or **Part 2.**)

#### **Step 1: Simple Message Virus**

| **Screenshot #1:** The commands and output of creating your message virus file |
| --- |
|  |
| **Notes** (Optional)**:** |

| **Project Question #1:** Fill in blanks in the **msfvenom** command to create the following virus:   * Payload: the (fictional) macOS/messagebox payload with a message of “OOF” * Target: an x86 architecture laptop running macOS * Virus File: a osx-app file named appleVirus ending in the .app extension | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **msfvenom -a** | x86 | **--platform** | | macOS | **-p** | macOS/messagebox TEXT=”OOF” |
|  |  | **-f** | app | | **-o** | appleVirus.app |

#### **Step 2: Multi-Payload Virus**

| **Screenshot #2:** The commands and output of creating your multi-payload virus file |
| --- |
|  |
| **Notes** (Optional)**:** |

#### 

| **Project Question #2:** In a few words, what does the payload **windows/speak\_pwned** do? |
| --- |
| The payload windows/speak\_pwned for msfvenom is designed to create a Windows executable that when executed will cause the system to speak a message "You have been pwned!" using the built-in Windows Speech API (SAPI). This payload is typically used for demonstration or educational purposes to showcase the potential impact of a successful exploit. |

#### **Step 3: Encrypted Virus**

| **Screenshot #3:** The commands and output of creating your encrypted virus file |
| --- |
|  |
| **Notes** (Optional)**:** |

#### 

| **Project Question #3:** MSFVenom’s encoder **x86/shikata\_ga\_nai** is a… (Fill in the blank) | | |
| --- | --- | --- |
| **“polymorphic** | **XOR** | **additive feedback encoder”** |

#### 

**Stretch Challenge (Optional)**

| **Stretch Challenge #1:** A screenshot showing the results of using **vt-cli** to evaluate at least one virus file. |
| --- |
|  |
| **Notes** (Optional)**:** |

#### 

| **Stretch Question #1:** Was **vt-cli** able to detect your file? Based on what you’ve learned this unit, what do you think is the reason why or why not? |
| --- |
| **No, vt-cli was not able to detect the file. If the virus is new or unknown, it may not be included in the virus database used by vt-cli. vt-cli relies on virus signatures and behavioral patterns to detect viruses, so if the virus is not included in its database, it may not be detected.** |

#### 

| **Stretch Challenge #2:** A screenshot showing the results of uploading one of the virus files to the [VirusTotal website](http://virustotal.com/). |
| --- |
|  |
| **Notes** (Optional)**:** |

#### 

| **Stretch Question #2:** Was VirusTotal able to detect your file? Based on what you’ve learned this unit, what do you think is the reason why or why not? |
| --- |
| **Yes, VirusTotal was able to detect the file. VirusTotal uses multiple antivirus engines to scan files, which increases the chances of detecting malware. Each antivirus engine has its own set of detection techniques and signatures, so a file that is missed by one engine may be detected by another.** |

**Submission Checklist**

**👉***Check off each of the features you have completed.* ***You will only be graded on the features you check off.***

**Reflection**

* ~~Reflection Question #1 answered above~~
* ~~Reflection Question #2 answered above~~
* ~~Shoutouts Completed~~

**Required Challenge Screenshots and Questions**

* ~~Screenshot #1~~
* ~~Project Question #1~~
* ~~Screenshot #2~~
* ~~Project Question #2~~
* ~~Screenshot #3~~
* ~~Project Question #3~~

**Stretch Challenge**

* ~~Screenshot showing~~ **~~vt-cli~~** ~~results~~
* ~~Stretch Question #1~~
* ~~Screenshot showing VirusTotal.com results~~
* ~~Stretch Question #2~~

**Submit your work!**

| Step 1: **Click** the Share button at the top of your screen double check that anyone with the link can edit. (This allows our grading team to input your grade below!)      Step 2: **Copy** the link to this document.    Step 3: **Submit** the link on the portal. |
| --- |

**Grader Comments**

| *Once your project has been assessed, our graders will leave feedback for you in this space. Please do not delete.* **Grading Rubric**  | Reflection Questions | Total Received Points | Total Possible | | --- | --- | --- | | Reflection Question #1 answered above | 2 | 2 | | Reflection Question #2 answered above | 2 | 2 | | **PART A TOTAL** | 4 | **4** | | Required Challenge Screenshots | Total Received Points | Total Possible | | Screenshot #1 | 4 | 4 | | Project Question #1 | 2 | 2 | | Screenshot #2 | 4 | 4 | | Project Question #2 | 1 | 1 | | Screenshot #3 | 4 | 4 | | Project Question #3 | 1 | 1 | | **PART B TOTAL** | **16** | **16** | | Stretch Challenge | Total Received Points | Total Possible | | Screenshot showing **vt-cli** results | 1 | +1 bonus | | Stretch Question #1 | 1 | +1 bonus | | Screenshot showing VirusTotal.com results | 1 | +1 bonus | | Stretch Question #2 | 1 | +1 bonus | | **Total Possible Points (Part A + Part B)** | **4** | **20** (+4) | |  | **24** |  |   **Grader Feedback** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |