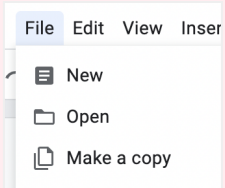
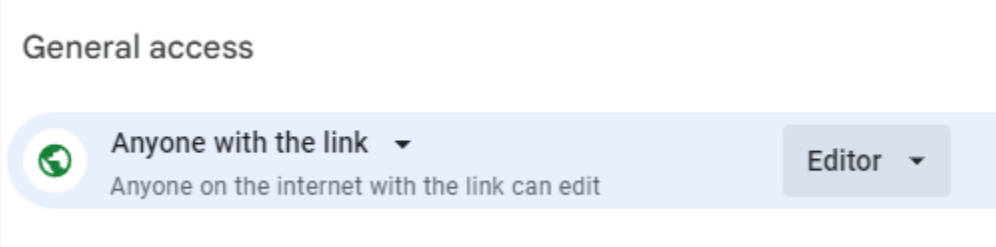
Instructions for Copying and Sharing this Document

📣 DELETE THIS BOX BEFORE SUBMITTING!!

Step 1: **Click** "File -> Make a Copy" to make a copy of this document that you can edit.



Step 2: **Change** the Share settings to "Anyone with Link -> Editor". This will allow our graders to leave comments on your submission.



CYB101 Project 3

(🔗 [Instructions Page])

👤 Student Name: Jonathan Siegel

✉️ Student Email: jsiegel9310@sdsu.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?

Put it in a virus checker or see if my operating systems virus detector can detect it.

📣 **Shoutouts:** Share appreciation for anyone who helped you out with this project or made your day a little better!

Michael Jackson

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

[Insert Screenshot Here]

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -a x86 --platform windows -p windows/messagebox TEXT="Virus Exec
uted" -f exe -o messageVirus.exe
No encoder specified, outputting raw payload
Payload size: 267 bytes
Final size of exe file: 73802 bytes
Saved as: messageVirus.exe

┌──(kali㉿kali)-[~]
└─$
```

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`**	osx	**`-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

[Insert Screenshot Here]

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -a x86 --platform windows -p windows/messagebox TEXT="Virus Exec
uted" -f exe -o messageVirus.exe
No encoder specified, outputting raw payload
Payload size: 267 bytes
Final size of exe file: 73802 bytes
Saved as: messageVirus.exe

┌──(kali㉿kali)-[~]
└─$ msfvenom -a x86 --platform windows \
  -p windows/messagebox TEXT="Virus Executed" \
  -f raw > messageBox
No encoder specified, outputting raw payload
Payload size: 267 bytes


┌──(kali㉿kali)-[~]
└─$ msfvenom -c messageBox -a x86 --platform windows \
  -p windows/speak_pwned -f exe -o pwnedVirus.exe
Adding shellcode from messageBox to the payload
No encoder specified, outputting raw payload
Payload size: 833 bytes
Final size of exe file: 73802 bytes
Saved as: pwnedVirus.exe

┌──(kali㉿kali)-[~]
└─$
```

Notes (Optional)**:**

Project Question #2: In a few words, what does the payload **`windows/speak_pwned`** do?

This is a payload to the virus, or essentially the payload itself. Once the system runs the payload it ultimately runs the virus as well.

Step 3: Encrypted Virus

Screenshot #3: The commands and output of creating your encrypted virus file

[Insert Screenshot Here]

```
kali@kali: ~
File  Actions  Edit  View  Help
-e: command not found

┌──(kali㉿kali)-[~]
└─$ msfvenom -a x86 --platform Windows \
  -p windows/messagebox TEXT="Encrypted Virus" \
  -e x86/shikata_ga_nai -i 3 -f python -o messageEncrypted
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 294 (iteration=0)
x86/shikata_ga_nai succeeded with size 321 (iteration=1)
x86/shikata_ga_nai succeeded with size 348 (iteration=2)
x86/shikata_ga_nai chosen with final size 348
Payload size: 348 bytes
Final size of python file: 1722 bytes
Saved as: messageEncrypted

┌──(kali㉿kali)-[~]
└─$ msfvenom -c messageEncrypted -a x86 \
  --platform windows -p windows/speak_pwned -f exe -o pyVirus.exe
Adding shellcode from messageEncrypted to the payload
No encoder specified, outputting raw payload
Payload size: 2298 bytes
Final size of exe file: 73802 bytes
Saved as: pyVirus.exe

┌──(kali㉿kali)-[~]
└─$
```

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.

[Insert Screenshot Here]

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?

> **Stretch Challenge #2:** A screenshot showing the results of uploading one of the virus files to the [VirusTotal website](#).
>
> **[Insert Screenshot Here]**
>
> **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?

Submission Checklist

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

Reflection

- [x] ~~Reflection Question #1 answered above~~
- [x] ~~Reflection Question #2 answered above~~
- [x] ~~Shoutouts Completed~~

Required Challenge Screenshots and Questions

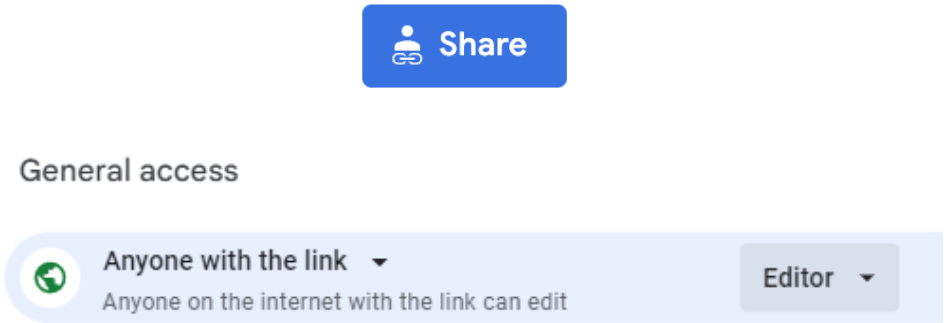
- [x] ~~Screenshot #1~~
- [x] ~~Project Question #1~~
- [x] ~~Screenshot #2~~
- [x] ~~Project Question #2~~
- [x] ~~Screenshot #3~~
- [x] ~~Project Question #3~~

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	0	+1 bonus
Stretch Question #1	0	+1 bonus
Screenshot showing VirusTotal.com results	0	+1 bonus
Stretch Question #2	0	+1 bonus
Total Possible Points (Part A + Part B)	**20**	**20** (+4)

Grader Feedback