Intrusion Detection Systems

Objective

The objective of this project is to provide a comprehensive understanding of network intrusion detection by simulating and analyzing common types of network attacks on a Windows-based system. Specifically, you will execute a Ping of Death attack and a port scanning attack to observe how these activities impact a network. Additionally, you'll explore how a signature-based Intrusion Detection System (IDS) detects these attacks.

You will develop a deeper understanding of the techniques attackers use to disrupt network operations and the importance of effective intrusion detection systems in identifying and mitigating such threats in real time. This lab reinforces the necessity and value of maintaining by recognizing and responding to various forms of malicious activity to maintain a secure network environment.

Prerequisites

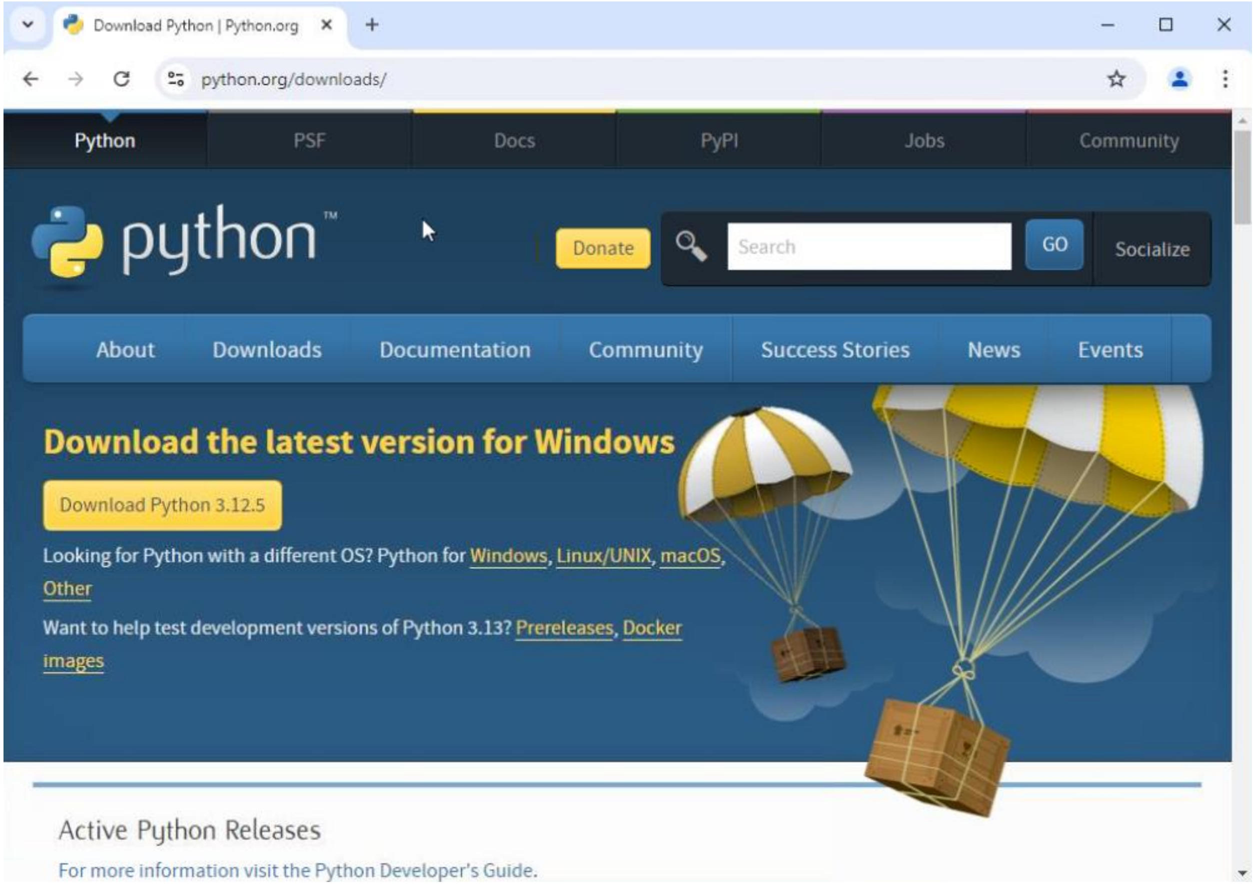
To successfully complete this project you'll need the following skills and software:

- Basic understanding of networking concepts (ICMP, TCP/IP)
- Python installed on your Windows machine
- Nmap installed on your Window Machine
- Port Scan Attacker and Detector is available on Machine

Lab Environment Configuration and Dependencies

Install the Latest Version of Python

Verify that Python 3.x is installed on your system. If Python is not already installed on your system, please download it by typing the following link into your browser: https://www.python.org/downloads/.



Download the latest version for Windows

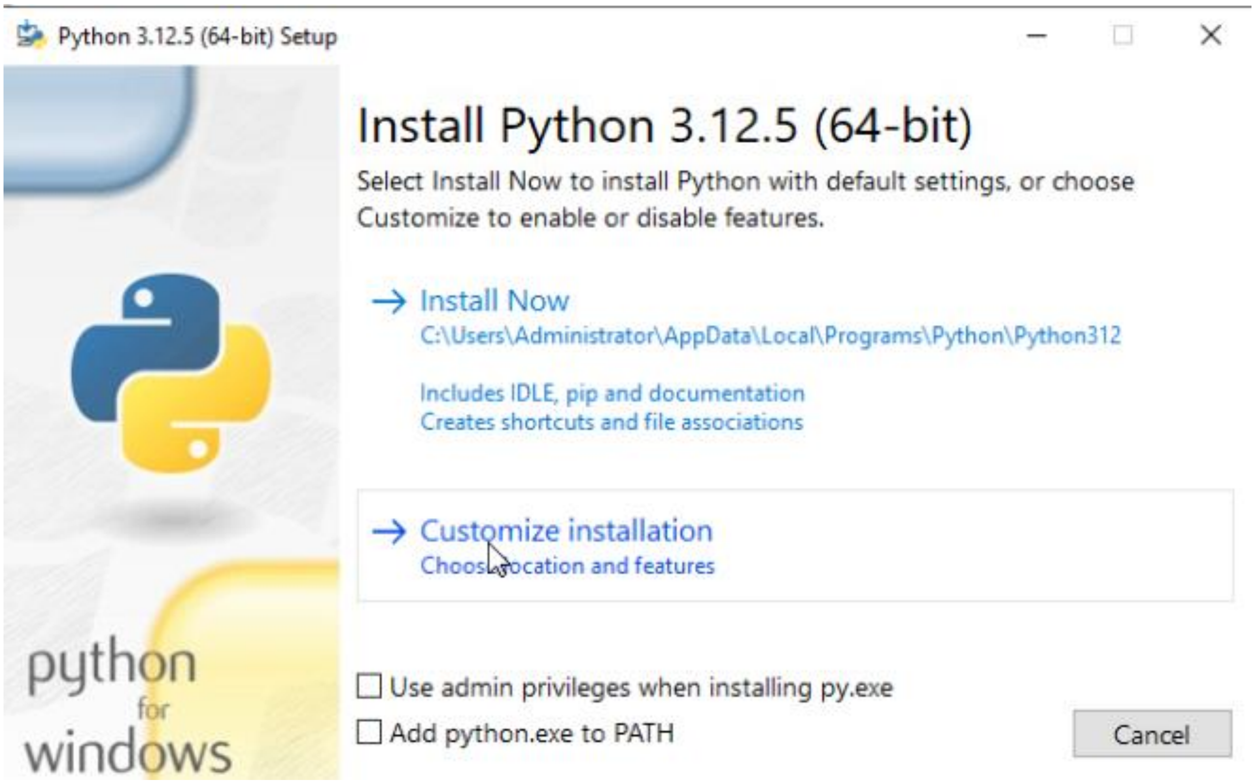
Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python 3.13? Prereleases, Docker images

Active Python Releases

For more information visit the Python Developer's Guide.

1. While Installing Python for Windows, select **Customize installation**.



Python 3.12.5 (64-bit) Setup

Install Python 3.12.5 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now
C:\Users\Administrator\AppData\Local\Programs\Python\Python312

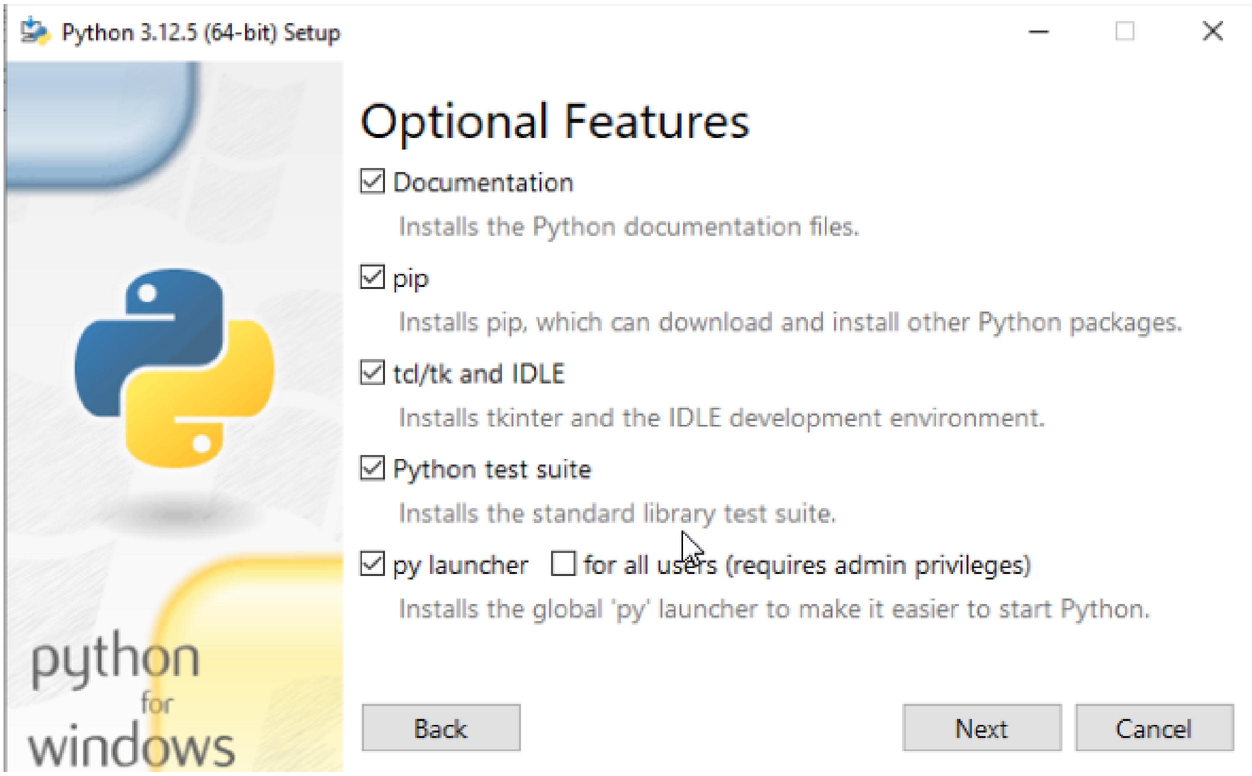
Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

□ Use admin privileges when installing py.exe
□ Add python.exe to PATH

2. On the **Optional Features** window, select the following options and click **Next**.

- Documentation
- pip
- tcl/tk and IDLE
- Python test suite
- py launcher



Optional Features

☑ Documentation
Installs the Python documentation files.

☑ pip
Installs pip, which can download and install other Python packages.

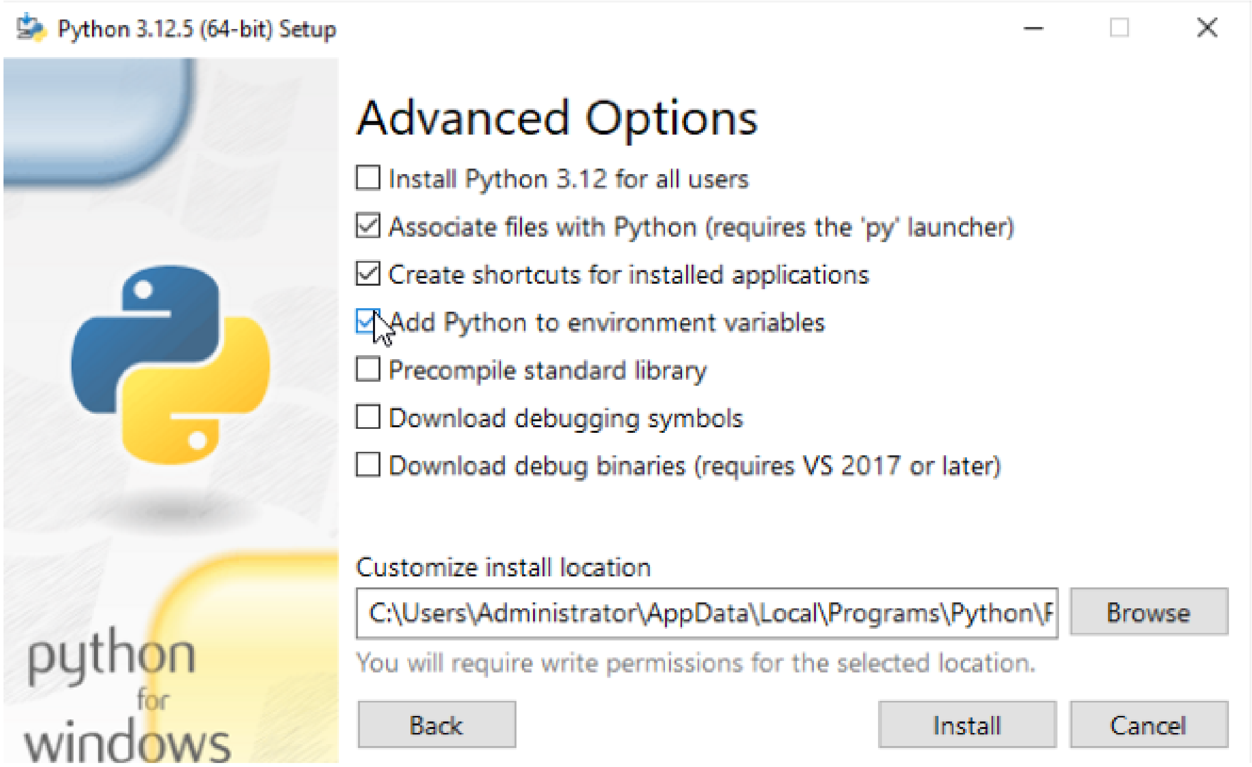
☑ tcl/tk and IDLE
Installs tkinter and the IDLE development environment.

☑ Python test suite
Installs the standard library test suite.

☑ py launcher □ for all users (requires admin privileges)
Installs the global 'py' launcher to make it easier to start Python.

3. On the **Advanced Options** window, select the following options and click **Install**.

- Associate files with Python (requires the 'py' launcher)
- Create shortcuts for installed applications
- Add Python to environment variables



Python 3.12.5 (64-bit) Setup

Advanced Options

☐ Install Python 3.12 for all users

☑ Associate files with Python (requires the 'py' launcher)

☑ Create shortcuts for installed applications

☑ Add Python to environment variables

☐ Precompile standard library

☐ Download debugging symbols

☐ Download debug binaries (requires VS 2017 or later)

Customize install location

C:\Users\Administrator\AppData\Local\Programs\Python\F

You will require write permissions for the selected location.

Install Scapy

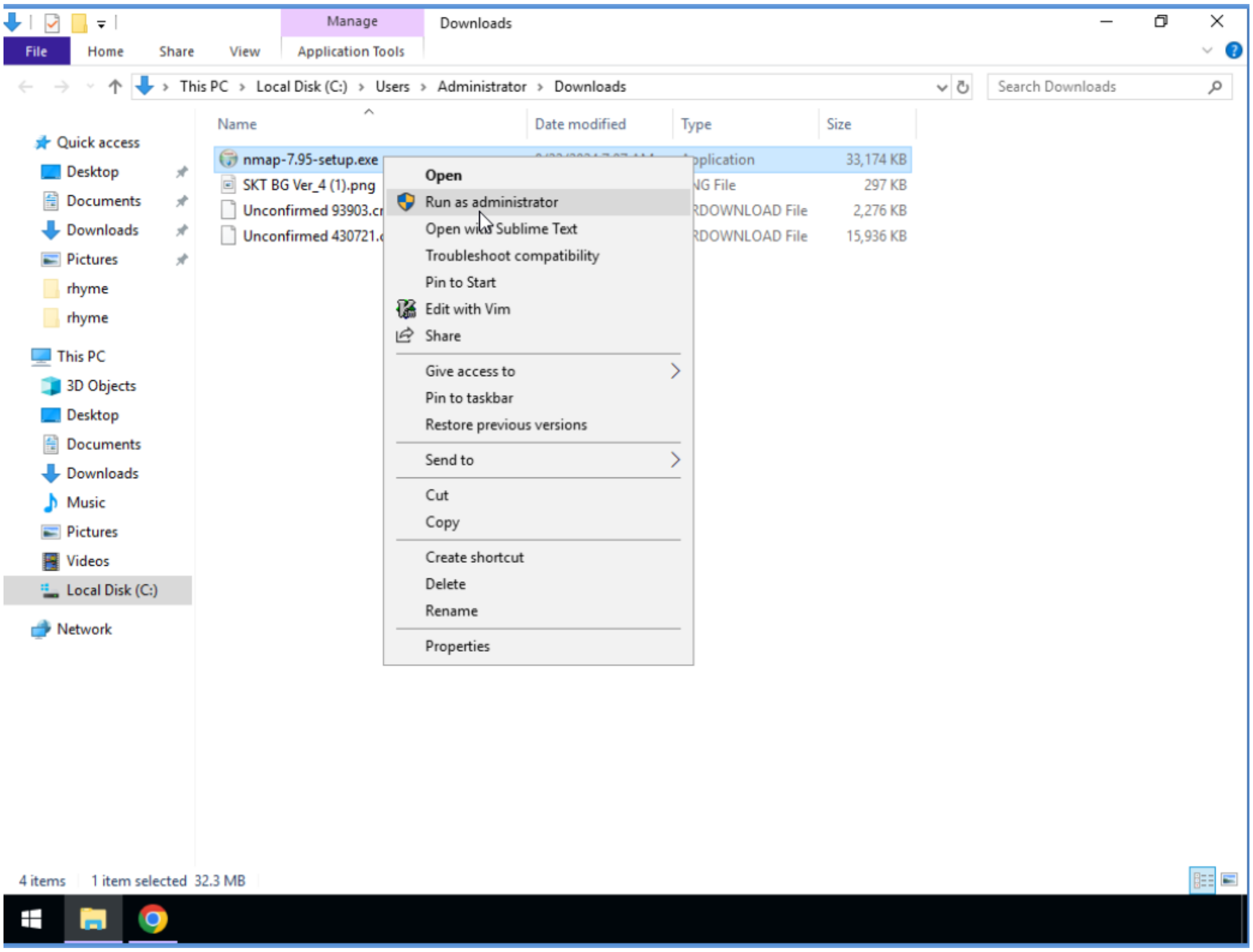
You'll need the Scapy library to use the **Sorst Scan Detecter** Python script. Scapy is a powerful Python library used for network packet manipulation and sending and receiving network packets.

- To install Scapy, open a command prompt using the `pip` command. Type the following command:

```
i.   pip install scapy
```

Install nmap for Port Scanning

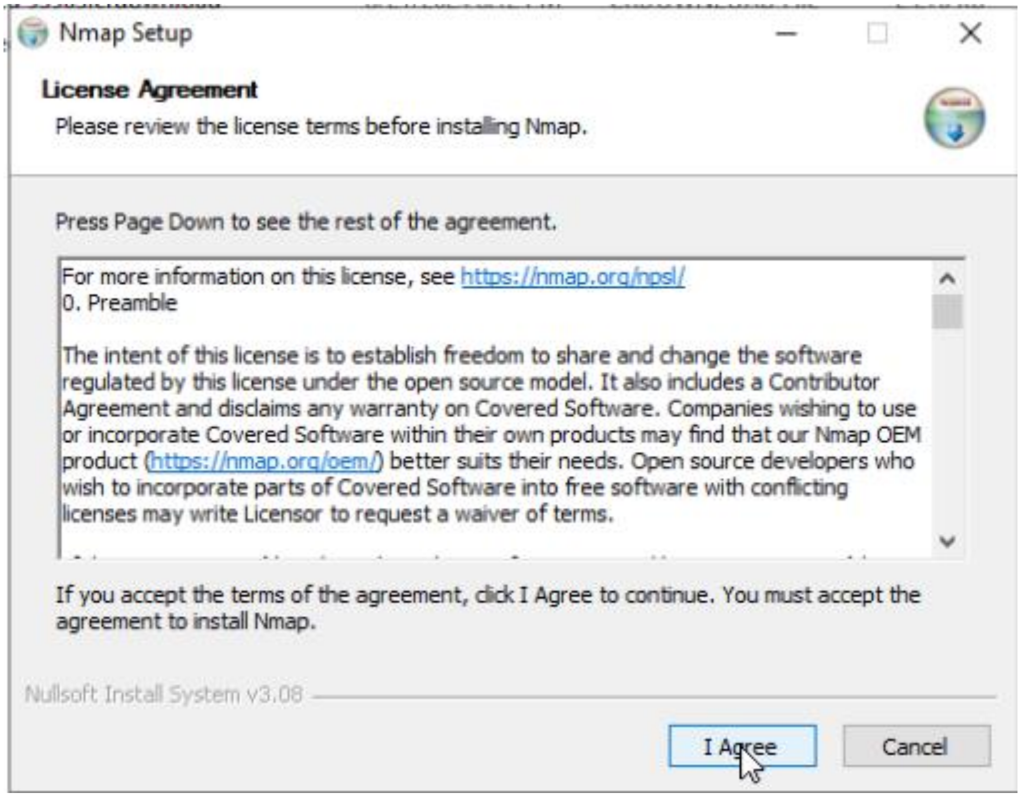
1. Download **nmap** from https://nmap.org/download.
2. Verify that you have administrator privileges for the computer.
3. Navigate the folder where you downloaded nmap.
4. Right-click the file name and install the application using administrative privildeges.



Name	Date modified	Type	Size
nmap-7.95-setup.exe		pplication	33,174 KB
SKT BG Ver_4 (1).png		IG File	297 KB
Unconfirmed 93903.cr		RDOWNLOAD File	2,276 KB
Unconfirmed 430721.c		RDOWNLOAD File	15,936 KB

4 items 1 item selected 32.3 MB

5. Install the application with all default options selected.



License Agreement

Please review the license terms before installing Nmap.

Press Page Down to see the rest of the agreement.

For more information on this license, see https://nmap.org/npsl/
0. Preamble

The intent of this license is to establish freedom to share and change the software regulated by this license under the open source model. It also includes a Contributor Agreement and disclaims any warranty on Covered Software. Companies wishing to use or incorporate Covered Software within their own products may find that our Nmap OEM product (https://nmap.org/oem/) better suits their needs. Open source developers who wish to incorporate parts of Covered Software into free software with conflicting licenses may write Licensor to request a waiver of terms.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Nmap.

Download the Port Scan Attacker and Detector

You also need to install the Port Scan Attacker and Detedtor Python files. You will download these files to the lab Workspace. The steps are as follows:

1. You can download these files on my GitHub project section . After successfully downloading the files, make sure the files are available on your desktop in the workstation environment during your session.

Port Scanning and Detection

Scenario: After mitigating a Ping of Death attack, you notice unusual network traffic patterns. Suspecting that an attacker might be scanning your network for open ports, you decide to perform a port scan to understand how an attacker might exploit your network.

After performing this process, you will verify if your intrusion detection system can detect such scanning activities.

What is Port Scanning?
Port scanning is a technique used by attackers to identify open ports and services running on a target machine. By scanning a range of ports, an attacker can determine which services are available and potentially vulnerable to exploitation. While port scanning is often used for legitimate network management, port scanning is also a common precursor to more serious attacks.

Perform a port scan:

1. In the search bar type **Command Prompt**. Right-click on it and select **Run as Administrator**.
2. In the command prompt window, navigate to the folder where `port_scan_attack.py` is located.

```
cd desktop
```

3. Run the the port scan script by typing the following command in the command prompt window:

```
python port_scan_attack.py
```

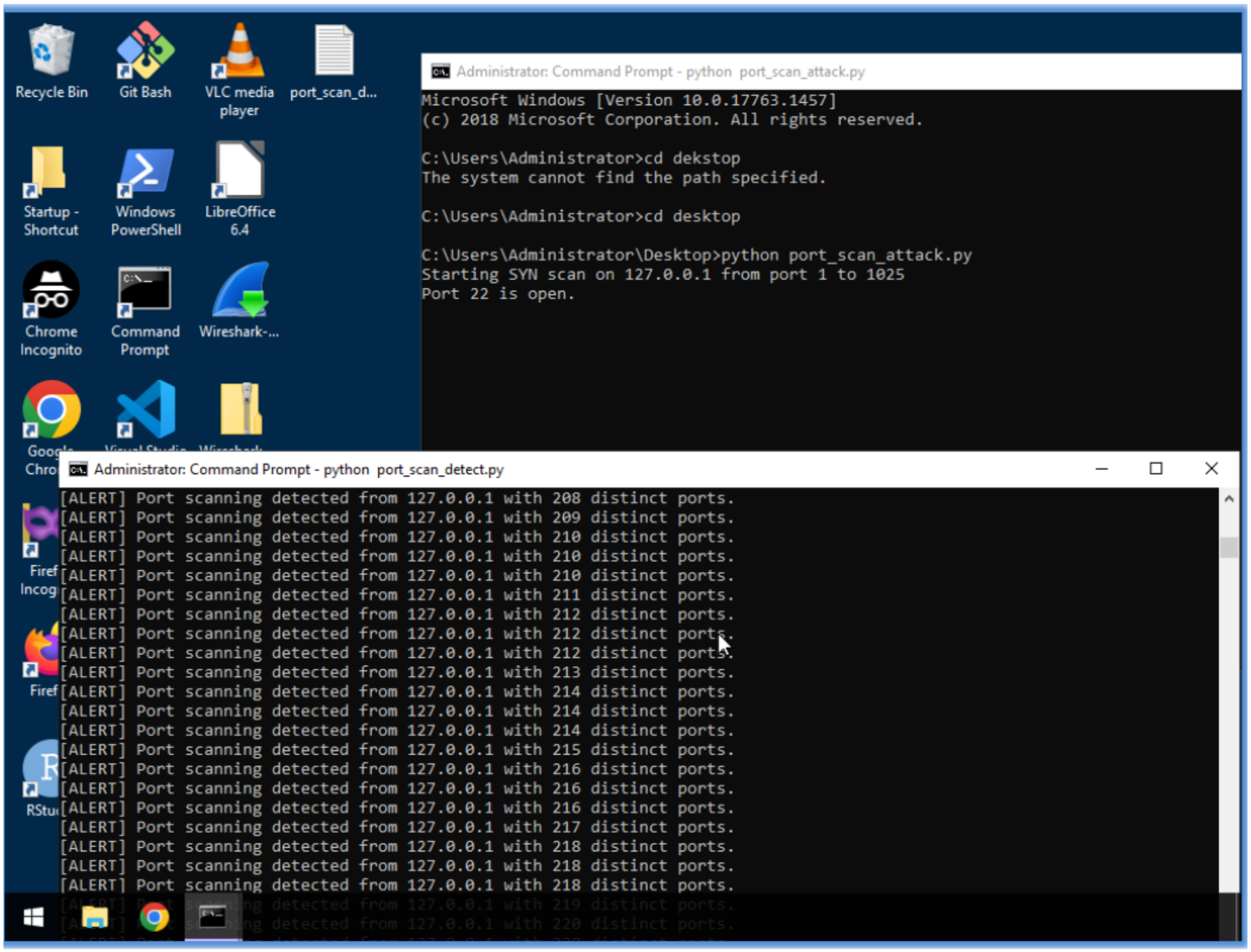
4. Open a second command prompt window.
5. In the command prompt, navigate to the folder where `port_scan_detect.py` is located.

```
cd desktop
```

5. Run the port scan script typing he following command in the command prompt window:

```
python port_scan_detect.py
```

6. Analyze the detection ouput. Review the detection output for any alerts generated by the script.



Administrator: Command Prompt - python port_scan_attack.py

```
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd dekstop
The system cannot find the path specified.

C:\Users\Administrator>cd desktop

C:\Users\Administrator\Desktop>python port_scan_attack.py
Starting SYN scan on 127.0.0.1 from port 1 to 1025
Port 22 is open.
```

Administrator: Command Prompt - python port_scan_detect.py

```
[ALERT] Port scanning detected from 127.0.0.1 with 208 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 209 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 210 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 210 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 210 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 211 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 212 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 212 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 212 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 213 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 214 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 214 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 214 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 215 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 216 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 216 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 216 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 217 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 218 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 218 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 218 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 219 distinct ports.
[ALERT] Port scanning detected from 127.0.0.1 with 220 distinct ports.
```

Scenario: In this project, you'll combine the power of Nmap, a widely used network scanning tool, with a custom Python script (`port_scan_detect.py`) to perform a local system scan and detect potential port scanning activities.

You will begin by launching **Nmap** with administrative privileges, then run the Python script to monitor the system. With **Nmap** running, you'll initiate a scan on the loopback IP address (`127.0.0.1`) and analyze the results.

Open Nmap

1. Select **Start** and locate the Nmap icon or search for the Namp icon on the desktop.
2. Right-click the Nmap icon and select **"Run as administrator"** to open Nmap with administrative privileges.

3. In the search bar type **Command Prompt**. Right-click on it and select **Run as Administrator**.
4. In the command prompt window, navigate to the folder where `port_scan_detect.py` is located using the following command:

```
cd desktop
```

5. Run the port scan script by typing the following command:

```
python port_scan_detect.py
```

Start the Nmap Scan

1. In the Nmap interface (Zenmap), type the loopback IP address `127.0.0.1` in the **Target** field.
2. From the available options, select the type of scan you want to perform such as **Intense scan** or **Quick scan**. Then select the **"Scan"** button to begin scanning the local system.

Review the Results

- Review the detection output for any alerts generated by the script.

Conclusion:

This Project demonstrates how to execute and detect network attacks on a Windows machine. By analyzing the alerts generated during the Ping of Death and port scanning activities, you gain insight into the importance of intrusion detection systems in maintaining network security. Through realistic scenarios, you've observed how these attacks can disrupt network operations and the critical role of detecting and mitigating them. Similarly, industry-standard Intrusion Detection Systems (IDS) operate using signature-based techniques to identify and respond to various types of attacks. These systems maintain signatures for a wide range of known threats, allowing them to recognize and mitigate malicious activities in real time, thereby safeguarding network environments from a multitude of security threats.