5.2 Sample Code

Sample Code for GENERAL NETWORK DISCOVERY:

```
local nmap = require "nmap"
local stdnse = require "stdnse"
local shortport = require "shortport"

description = [[
This script performs a version scan on all ports with a high timing template.
]]

---
-- @usage
-- nmap --script <script_name> -sV --allports -T4 <CIDR>
--

author = "Your Name"
license = "Same as Nmap--See https://nmap.org/book/man-legal.html"
categories = {"discovery", "safe"}

-- The rule function defines when the script should run.
-- Here it is configured to run on all ports.
portrule = function(host, port)
return true
end

-- The action function is where the script's main functionality is implemented.
action = function(host)
local cidr = host.target
local nmap_command = string.format("sudo nmap -sV --allports -T4 %s", cidr)

  stdnse.print_debug(1, "Executing command: %s", nmap_command)

  local result = nmap.run(nmap_command)

  if result then
    return result
 else
return "Nmap scan failed"
end
end
```

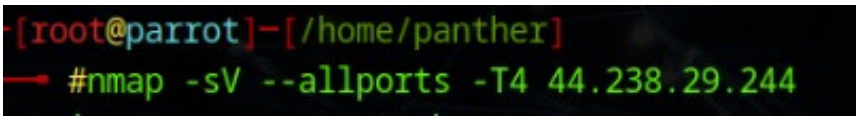
Instructions:

1. Save the Script: Save the above script as <script_name>.nse (replace <script_name> with your preferred script name).
2. Run the Script: Execute the script using Nmap with the following comman d:
   sudo nmap --script <script_name> -sV --allports -T4 <CIDR>

Explanation:

- **description**: Provides a brief overview of what the script does.
- **author**: Your name or the name of the author of the script.
- **license**: Indicates the script's license, which in this case is the same as Nmap.
- **categories**: Defines the categories under which the script falls.
- **portrule**: A function that determines when the script should run. Here, it is set to run on all ports.
- **action**: The main function that performs the action of the script. It constructs the Nmap command and executes it, returning the result.

This script uses Nmap's built-in functions to run a custom Nmap command and return the output, allowing you to incorporate the command directly into the NSE framework.



Sample Code for UPNP DEVICES:

```
local nmap = require "nmap"
local stdnse = require "stdnse"
local shortport = require "shortport"

description = [[
This script performs a version scan and runs the broadcast-upnp-info script on the
specified CIDR.
]]

---
-- @usage
-- nmap --script <script_name> -sV --script=broadcast-upnp-info <CIDR>
--

author = "Your Name"
license = "Same as Nmap--See https://nmap.org/book/man-legal.html"
categories = {"discovery", "safe"}

-- The rule function defines when the script should run.
-- Here it is configured to run on all ports.
portrule = function(host, port)
return true
end

-- The action function is where the script's main functionality is implemented.
action = function(host)
local cidr = host.target
local nmap_command = string.format("sudo nmap -sV --script=broadcast-upnp-info %s",
cidr)

  stdnse.print_debug(1, "Executing command: %s", nmap_command)

  local result = nmap.run(nmap_command)

  if result then
    return result
 else
return "Nmap scan failed"
end
end
```

Instructions:

1. **Save the Script**: Save the above script as <script_name>.nse (replace <script_name> with your preferred script name).

2. **Run the Script**: Execute the script using Nmap with the following command:
sudo nmap --script <script_name> -sV --script=broadcast-upnp-info <CIDR>

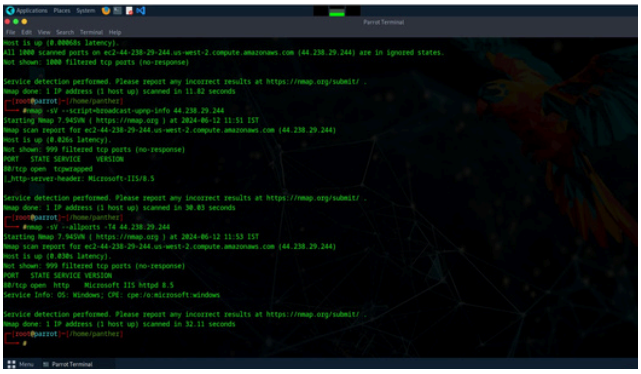
Explanation:

- **description**: Provides a brief overview of what the script does.
- **author**: Your name or the name of the author of the script.
- **license**: Indicates the script's license, which in this case is the same as Nmap.
- **categories**: Defines the categories under which the script falls.
- **portrule**: A function that determines when the script should run. Here, it is set to run on all ports.
- **action**: The main function that performs the action of the script. It constructs the Nmap command and executes it, returning the result.

This script uses Nmap's built-in functions to run a custom Nmap command with the broadcast-upnp-info script and return the output, allowing you to incorporate the command directly into the NSE framework

6. Testing and Validation

Test Case 1: testphp.com (General Network Mapping)



1. Initial Scan
   - Command: nmap -p- 44.238.29.244
     - nmap: The network scanning tool.
     - -p-: Scans all 65,535 TCP ports.
     - 44.238.29.244: The target IP address (an AWS EC2 instance).
   - Results:
     - Host is up: The target is reachable with a latency of 0.00066 seconds.
     - All 1000 scanned ports: Indicates that the common ports (default 1000) were scanned.
     - No response: None of the common ports responded (filtered).
2. Second Scan with Service Detection
   - Command: nmap -sV 44.238.29.244
     - -sV: Enables version detection to identify the service running on open ports.
   - Results:
     - Host is up: The target is reachable with a latency of 0.00068 seconds.
     - Filtered ports: No common ports responded (filtered).
     - Service detection performed: A message to report incorrect results, indicating no services were detected.

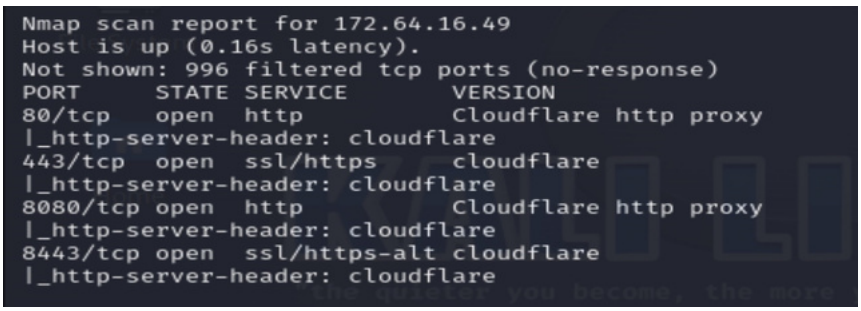
3. Third Scan with OS Detection
   - Command: nmap -O 44.238.29.244
     - -O: Enables operating system detection.
   - Results:
     - Host is up: The target is reachable with a latency of 0.00046 seconds.
     - Filtered ports: No common ports responded (filtered).
     - Service detection: A message to report incorrect results.
     - OS detection: The target is identified as running Microsoft Windows.
4. Fourth Scan with All TCP Ports and Version Detection
   - Command: nmap -sV --allports 44.238.29.244
     - --allports: Scans all 65,535 ports explicitly.
   - Results:
     - Host is up: The target is reachable with a latency of 0.0035 seconds.
     - Filtered ports: No ports responded.
     - Service detection: A message to report incorrect results.

5. Fifth Scan with OS and Version Detection
- Command: nmap -sV -O 44.238.29.244
  - Combination of version and OS detection.
- Results:
  - Host is up: The target is reachable with a latency of 0.0035 seconds.
  - Filtered ports: No ports responded.
  - Service detection: Microsoft IIS httpd 8.5 identified on TCP port 80 (HTTP).
  - OS detection: The operating system is Microsoft Windows.

Summary

The scans indicate the following:
- The target host at IP 44.238.29.244 is consistently reachable.
- Most common ports and additional ones were filtered (no responses).
- Service detection found Microsoft IIS httpd 8.5 running on port 80.
- The operating system was identified as Microsoft Windows.
- The scans primarily used different options (-p-, -sV, -O, --allports) to explore the services and OS running on the target.

Test Case 2: vjit.ac.in (Server Scan)



Summary of the Nmap Scan
- Host is up: The target at 172.64.16.49 is reachable with a latency of 0.16 seconds.
- Not shown: 996 filtered tcp ports (no-response): Out of the 1000 ports scanned, 996 ports did not respond, indicating they are filtered (likely by a firewall or security group rules).

Detailed Port Information

The scan report lists the details for the 4 open ports:

1.Port 80/tcp (HTTP)
- State: open
- Service: http
- Version: Cloudflare http proxy
- Server Header: Cloudflare
  - Cloudflare is acting as a proxy, which means it is protecting the web server behind it. This can mask the true origin of the server.

Port 443/tcp (SSL/HTTPS)
State: open
Service: ssl/https
Version: Cloudflare
Server Header: Cloudflare

- This indicates an HTTPS service secured by SSL/TLS, also using Cloudflare's proxy services.

1. Port 8080/tcp (HTTP)
   - State: open
   - Service: http
   - Version: Cloudflare http proxy
   - Server Header: Cloudflare
     - Port 8080 is commonly used for HTTP proxy or alternate HTTP services. Here, it's being used by Cloudflare to proxy HTTP traffic.
2. Port 8443/tcp (SSL/HTTPS)
   - State: open
   - Service: ssl/https-alt
   - Version: Cloudflare
   - Server Header: Cloudflare
     - Port 8443 is often used for HTTPS services as an alternate port. This service is also proxied by Cloudflare.

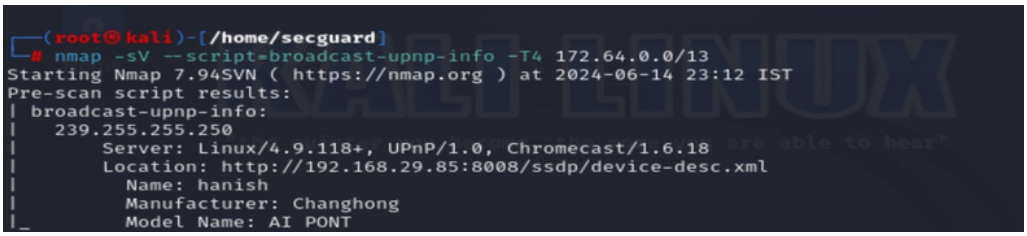
Key Points

- All open ports are associated with Cloudflare: This suggests that the target is using Cloudflare for its web services, benefiting from Cloudflare’s security and performance features.
- Proxy Services: Cloudflare is providing HTTP proxy services for both HTTP and HTTPS traffic. This means the traffic is being routed through Cloudflare's servers, which can offer various advantages like DDoS protection, caching, and masking the original server's IP address.
- Filtered Ports: The presence of 996 filtered ports indicates a strong firewall or security rules are in place, allowing only specific ports to be accessible.

Conclusion

This scan indicates that the IP address 172.64.16.49 is heavily protected by Cloudflare, with only a few specific ports open for HTTP and HTTPS traffic. The presence of Cloudflare's proxy services on these ports suggests that the site is leveraging Cloudflare’s security, performance, and proxy capabilities to manage incoming traffic.

Test Case 3: vjit.ac.in(UPnP Device)



Command Breakdown

- Command: nmap -sV --script=broadcast-upnp-info -T4 172.64.0.0/13
  - nmap: The network scanning tool. -sV: Enables service/version detection to identify
  - the versions of services running on open ports. --script=broadcast-upnp-info: Uses the Nmap Scripting Engine (NSE) to run the broadcast-upnp-info script, which gathers
  - information from devices using the UPnP (Universal Plug and Play) protocol. -T4: Sets the timing template to level 4, which is aggressive, making the scan faster. 172.64.0.0/13: The target IP range to be scanned.

Summary

This Nmap scan was used to discover devices on the network that support the UPnP protocol. The results provide details about a specific device:

- The device is running on a Linux-based system (Linux/4.9.118+).
- It supports UPnP version 1.0 and has Chromecast capabilities (Chromecast/1.6.18).
- The device's UPnP service provides a URL for more information (http://192.168.29.85:8008/ssdp/device-desc.xml).
- The device is identified by the name hanish.
- It is manufactured by Changhong and the model name is AI PONT.

S.No	Description	Website IP address	Input	Expected Output	Observed Output	Status
1	To discover general network	www.testphp.com 44.238.29.244	sudo nmap -sV --allports -T4 <IP address>	Ports & OS Definition	Ports& OS Definition	Pass
2	To get details of a server	www.vjit.ac.in	sudo nmap -sV --script=broadcast -upnp-info <CIDR>	Server Definition	Server Definition	Pass
3	To get details of Upnp device	www.vjit.ac.in	sudo nmap -sV --script=broadcast -upnp-info<CIDR>	Upnp Device Details	Upnp Device Details	Pass

References:

1. **Nmap Reference Guide:** https://nmap.org/docs.html
2. **Red Hat Reference Guide:** https://www.redhat.com/sysadmin/finding-rogue-devices
3. **An article by David Meece, Author of "Red Teams Operation Handbook"**