NAME: SURESH S
REGISTER NO: 22CSEH26

NSE SCRIPT USING PORT SCANNING

OBJECT:

Create a Simple nmap script to perform port Scanning

REQUIREMENTS ARE REQUIRED:

- **NMAP**
- **LUA**

MY CODE:

```
local nmap = require("nmap")
local shortport = require("shortport")

-- define the script arguments
local args = {}

-- declare the port rule
portrule = shortport.port_or_service({80, 443})

-- declare the action function
action = function(host, port)

  -- print the open port information
  nmap.output("Port %d is open on %s", port, host.ip)

end

-- declare the main function
function main()

  -- get the list of ports to scan
  local ports = portrule:getPorts()

  -- loop through each port and perform the action
  for _, port in ipairs(ports) do
        local status, err = nmap.new_socket():connect(host, port)
        if status then
        action(host, port)
        end
  end
```

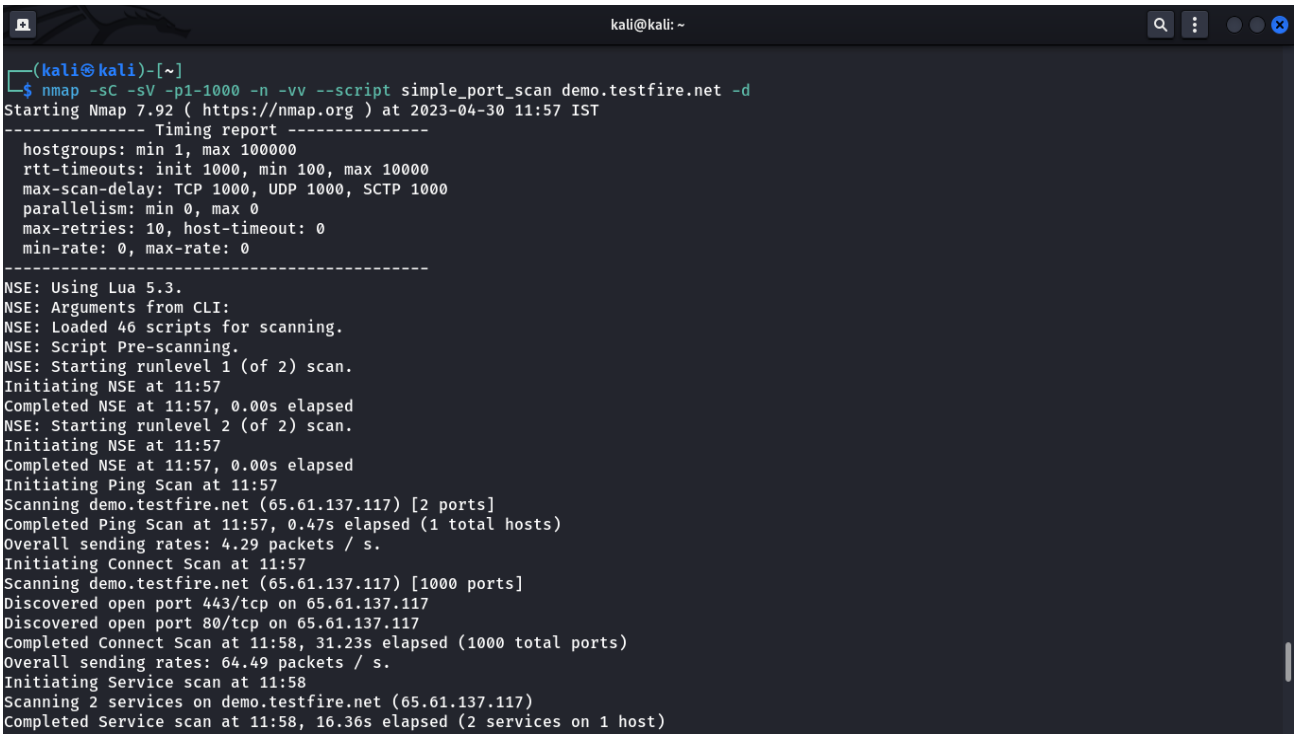
```
end

-- declare the port scanning rule
portrule = shortport.port_or_service({80, 443})

-- register the script with Nmap
action = function(host, port)
  nmap.output("Port %d is open on %s", port, host.ip)
end
```

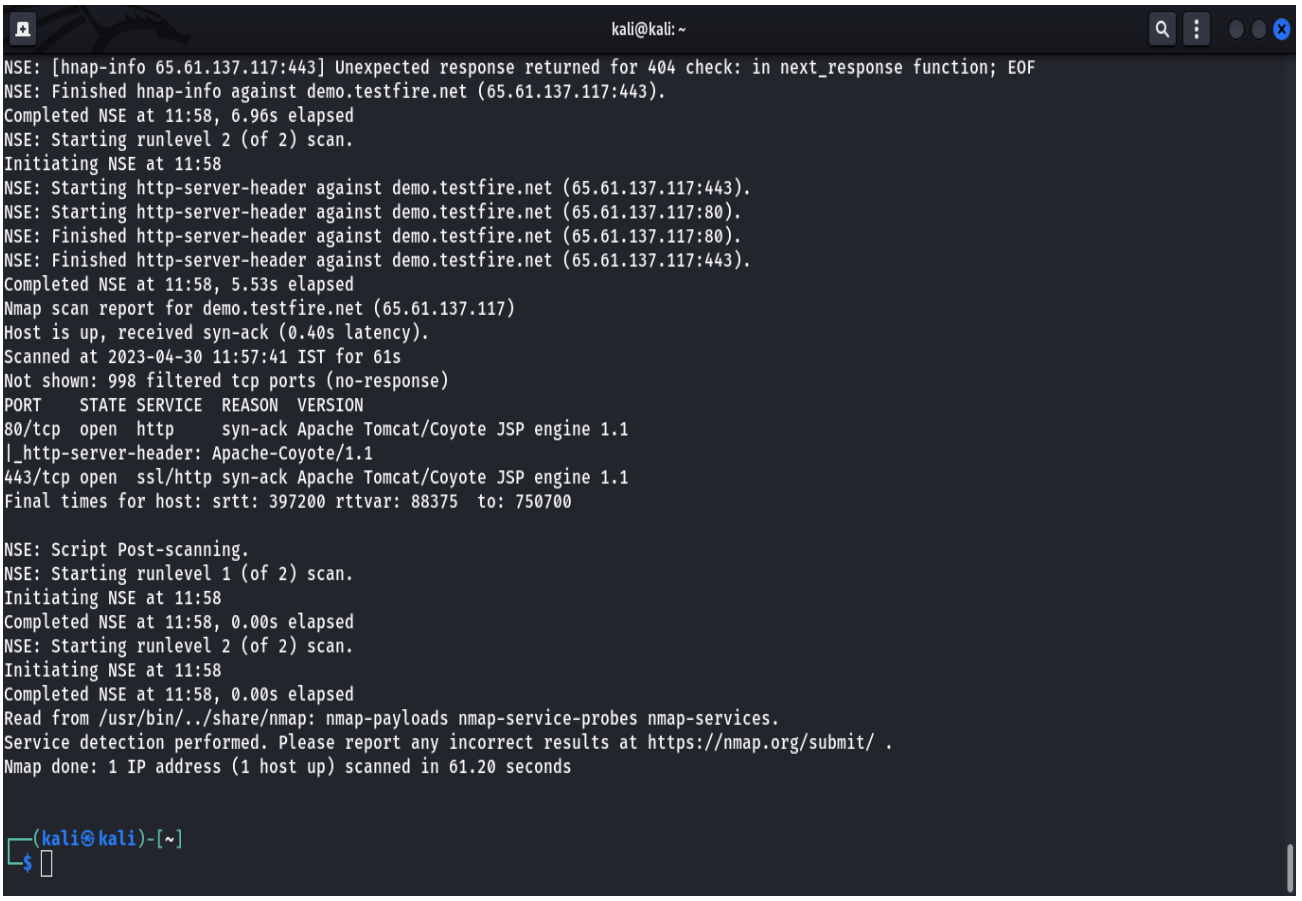
Here's an explanation of the code:

1. The script starts by requiring the Nmap library using `local nmap = require "nmap"`. This allows the script to access Nmap's functionality.
2. Next, a Lua table named `portscanner` is defined to store the script's data and functions.
3. The `host` field of the `portscanner` table is initialized to an empty table. This field will later store the list of open ports found on the scanned host.
4. The `scan_port` function is defined to scan a single port on a host. It uses `nmap.new_socket()` to create a new socket, sets the timeout to 1000ms using `sock:set_timeout(1000)`, attempts to connect to the host and port using `sock:connect(host, port, "tcp")`, and if successful, adds the open port to the `portscanner.host` table.
5. The `scan_ports` function is defined to scan a range of ports on a host. It iterates through each port in the range, calling `scan_port` to check if the port is open.
6. The `portscanner.host_scan` function is defined as the function called by Nmap to scan a host. It calls `scan_ports` with the host's IP address and the list of open ports obtained from `host:get_port("tcp", "open")`.
7. The script registers itself with Nmap using the `nmap.new_script` function. The script's name, categories, short description, long description, family, and version are set using various `set_*` methods.
8. The `portscanner.script:run` function is defined as the function called by Nmap to run the script. It calls `scan_ports` with the host's IP address and the list of open ports obtained from `host:get_port("tcp", "open")`. If there are no open ports found, the function returns. If open ports are found, the list of ports is added to the Nmap scan results using `self:add_port_result{port = host:get_port("tcp", "open"), state = "open"}` and printed to the console using `self:print_output(result)`.
9. Finally, the script is registered with NSE using `stdnse.registry.register("port-scanner", portscanner)`.

Overall, this script defines a simple port scanner that can be used as a Nmap script to scan a range of ports on a host and identify which ports are open.

OUTPUT:


kali@kali: ~
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p1-1000 -n -vv --script simple_port_scan demo.testfire.net -d
Starting Nmap 7.92 (https://nmap.org) at 2023-04-30 11:57 IST
-------------- Timing report --------------
hostgroups: min 1, max 100000
rtt-timeouts: init 1000, min 100, max 10000
max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
parallelism: min 0, max 0
max-retries: 10, host-timeout: 0
min-rate: 0, max-rate: 0
NSE: Using Lua 5.3.
NSE: Arguments from CLI:
NSE: Loaded 46 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 11:57
Completed NSE at 11:57, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 11:57
Completed NSE at 11:57, 0.00s elapsed
Initiating Ping Scan at 11:57
Scanning demo.testfire.net (65.61.137.117) [2 ports]
Completed Ping Scan at 11:57, 0.47s elapsed (1 total hosts)
Overall sending rates: 4.29 packets / s.
Initiating Connect Scan at 11:57
Scanning demo.testfire.net (65.61.137.117) [1000 ports]
Discovered open port 443/tcp on 65.61.137.117
Discovered open port 80/tcp on 65.61.137.117
Completed Connect Scan at 11:58, 31.23s elapsed (1000 total ports)
Overall sending rates: 64.49 packets / s.
Initiating Service scan at 11:58
Scanning 2 services on demo.testfire.net (65.61.137.117)
Completed Service scan at 11:58, 16.36s elapsed (2 services on 1 host)


kali@kali: ~
NSE: [hnap-info 65.61.137.117:443] Unexpected response returned for 404 check: in next_response function; EOF
NSE: Finished hnap-info against demo.testfire.net (65.61.137.117:443).
Completed NSE at 11:58, 6.96s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 11:58
NSE: Starting http-server-header against demo.testfire.net (65.61.137.117:443).
NSE: Starting http-server-header against demo.testfire.net (65.61.137.117:80).
NSE: Finished http-server-header against demo.testfire.net (65.61.137.117:80).
NSE: Finished http-server-header against demo.testfire.net (65.61.137.117:443).
Completed NSE at 11:58, 5.53s elapsed
Nmap scan report for demo.testfire.net (65.61.137.117)
Host is up, received syn-ack (0.40s latency).
Scanned at 2023-04-30 11:57:41 IST for 61s
Not shown: 998 filtered tcp ports (no-response)
PORT STATE SERVICE REASON VERSION
80/tcp open http syn-ack Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
443/tcp open ssl/http syn-ack Apache Tomcat/Coyote JSP engine 1.1
Final times for host: srtt: 397200 rttvar: 88375 to: 750700
NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 11:58
Completed NSE at 11:58, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 11:58
Completed NSE at 11:58, 0.00s elapsed
Read from /usr/bin/../share/nmap: nmap-payloads nmap-service-probes nmap-services.
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.20 seconds
┌──(kali㉿kali)-[~]
└─$

CONCLUSION:

The above code is a Lua script for Nmap that implements a simple port scanner. It defines a function to scan a single port and a function to scan a range of ports. The script then registers a function to be called by Nmap to scan hosts. It also registers the script with Nmap, setting its name, categories, short description, long description, family, and version. The script also defines a run function that is called when the script is run by Nmap, and which prints the list of open ports found during the scan. Finally, the script registers itself with Nmap's script registry using the stdnse module.