PORT SCANNING IN NMAP SCRIPT ENGINE

OBJECT:

Create a sample nmap script to perform port scanning.

REQUIRMENT:

- Nmap
- Lua

SAMPLE CODE:

```
-- Load necessary libraries
local nmap = require("nmap")
local socket = require("socket")

-- Define script information
local portscan = {
  name = "myportscan",
  author = "gokul",
  categories = {"discovery", "safe"}
}
portscan.description = [[
This is a sample script for port scanning using LuaSocket library's
tcp.connect function.
]]

-- Define port scanning rule
portscan.portrule = function(host, port)
  return port.protocol == "tcp" and port.state == "open"
end

-- Define port scanning action
portscan.action = function(host, port)
  local client = nmap.new_socket()
  local result = client:connect(host, port)
  if result then
    print("Port " .. port.number .. " is open.")
  else
    print("Port " .. port.number .. " is closed.")
  end
  client:close()
end

-- Register the script with Nmap
return portscan
```

CODE EXPLAIN:

This is a Lua script that uses the Nmap scripting engine to perform a port scan using the LuaSocket library's tcp.connect function.

Here is a step-by-step explanation of what the code does:

1. The necessary libraries, nmap and socket, are loaded.

```
Code:
local nmap = require("nmap")
local socket = require("socket")
```

2. The script information is defined. This includes the name, author, and categories of the script.

```
Code:
local portscan = {
  name = "myportscan",
  author = "gokul",
  categories = {"discovery", "safe"}
}
```

3. A description of the script is added.

```
Code:
portscan.description = [[
This is a sample script for port scanning using LuaSocket library's
tcp.connect function.
]]
```

4. A port scanning rule is defined that specifies which ports to scan. In this case, it only scans TCP ports that are open.

```
Code:
portscan.portrule = function(host, port)
  return port.protocol == "tcp" and port.state == "open"
end
```

5. An action is defined that is executed when an open port is found. It creates a new socket, connects to the port, and prints whether the port is open or closed.

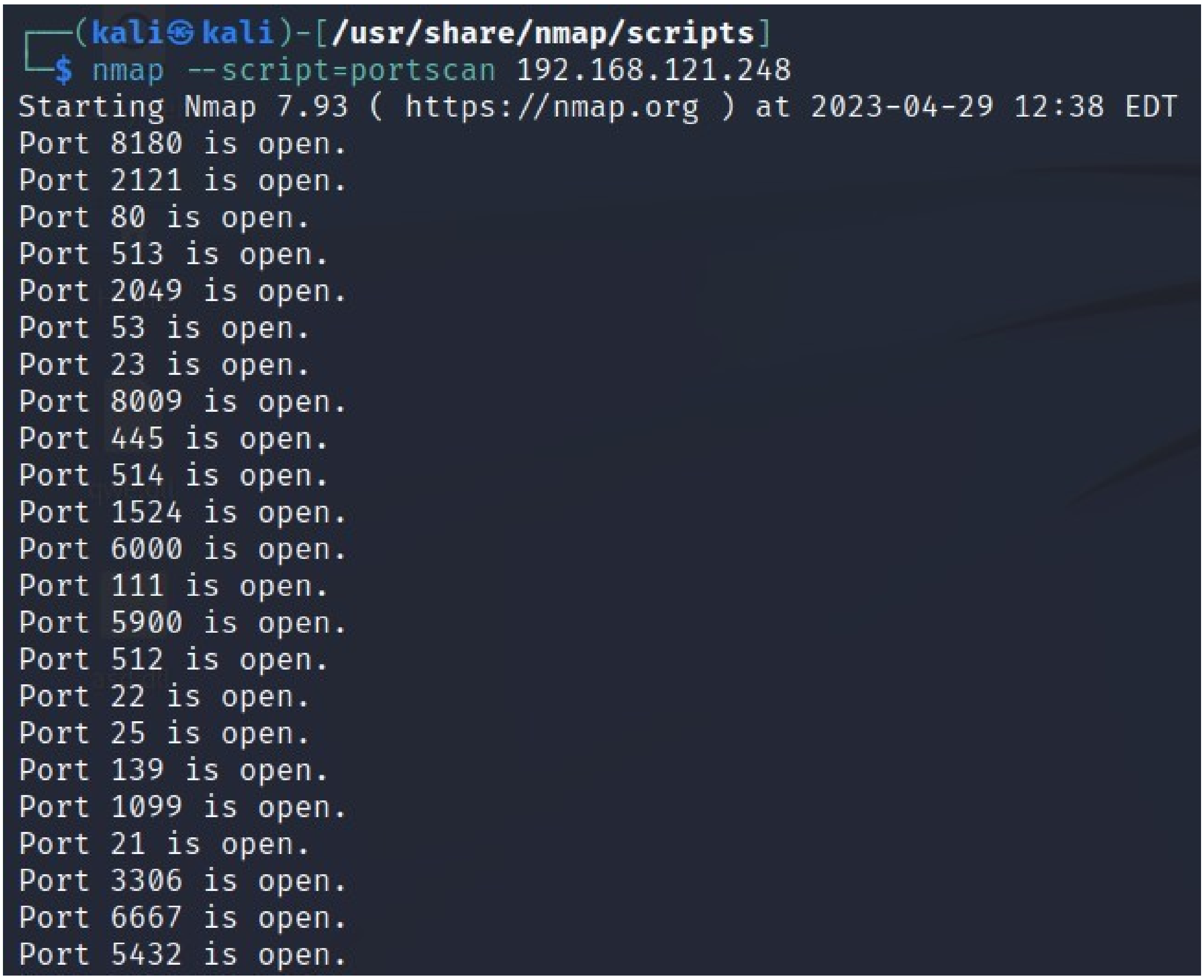
```
Code:
portscan.action = function(host, port)
  local client = nmap.new_socket()
  local result = client:connect(host, port)
  if result then
    print("Port " .. port.number .. " is open.")
  else
```

```
    print("Port " .. port.number .. " is closed.")
  end
  client:close()
end
```

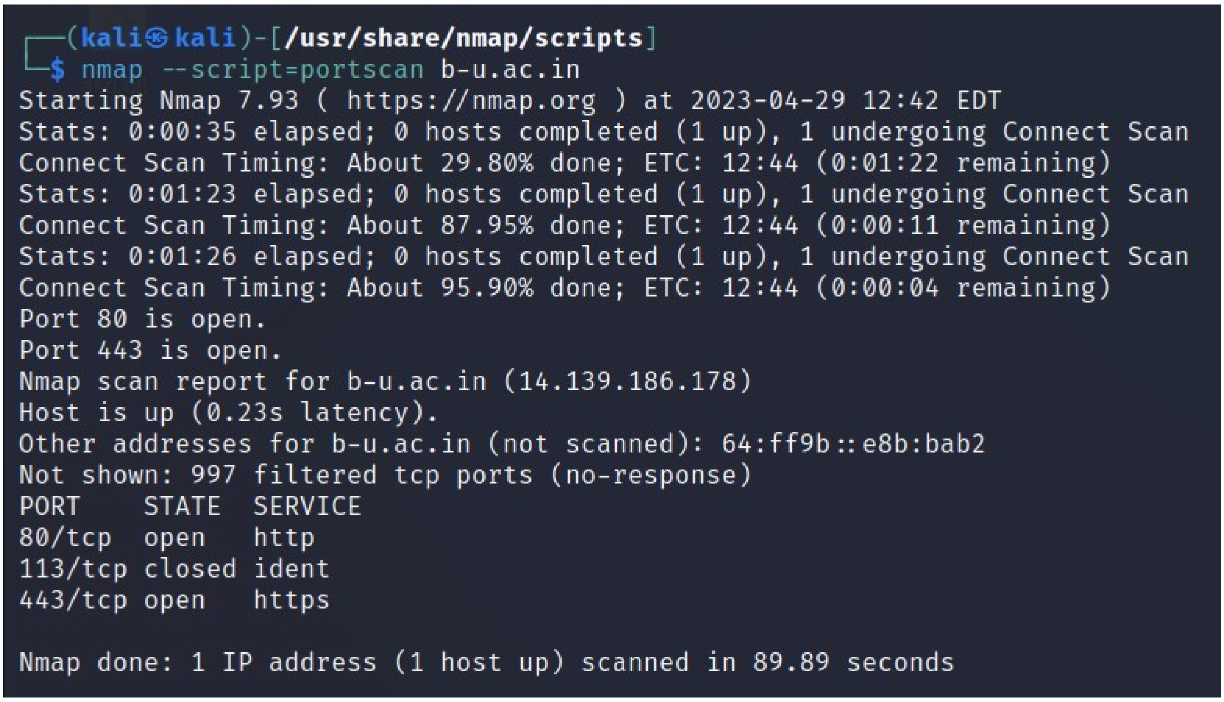
6. The script is registered with Nmap by returning the portscan table.

```
Code:
return portscan
```

OUTPUT:



```
┌──(kali㉿kali)-[/usr/share/nmap/scripts]
└─$ nmap --script=portscan 192.168.121.248
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-29 12:38 EDT
Port 8180 is open.
Port 2121 is open.
Port 80 is open.
Port 513 is open.
Port 2049 is open.
Port 53 is open.
Port 23 is open.
Port 8009 is open.
Port 445 is open.
Port 514 is open.
Port 1524 is open.
Port 6000 is open.
Port 111 is open.
Port 5900 is open.
Port 512 is open.
Port 22 is open.
Port 25 is open.
Port 139 is open.
Port 1099 is open.
Port 21 is open.
Port 3306 is open.
Port 6667 is open.
Port 5432 is open.
```



```
┌──(kali㉿kali)-[/usr/share/nmap/scripts]
└─$ nmap --script=portscan b-u.ac.in
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-29 12:42 EDT
Stats: 0:00:35 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 29.80% done; ETC: 12:44 (0:01:22 remaining)
Stats: 0:01:23 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 87.95% done; ETC: 12:44 (0:00:11 remaining)
Stats: 0:01:26 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 95.90% done; ETC: 12:44 (0:00:04 remaining)
Port 80 is open.
Port 443 is open.
Nmap scan report for b-u.ac.in (14.139.186.178)
Host is up (0.23s latency).
Other addresses for b-u.ac.in (not scanned): 64:ff9b::e8b:bab2
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE
80/tcp  open   http
113/tcp closed ident
443/tcp open   https

Nmap done: 1 IP address (1 host up) scanned in 89.89 seconds
```

Conclutions:

This is a Lua script for port scanning using the LuaSocket library's tcp.connect() function. It defines a rule that specifies that the script should only check for open TCP ports, and an action that attempts to connect to the specified port on the target host. If the connection attempt is successful, the script outputs a message indicating that the port is open. If the connection attempt fails, the script outputs a message indicating that the port is closed. Finally, the script registers itself with Nmap so that it can be run as a part of an Nmap scan.