NSE Script for Network Information Gathering

Metbrains Internship Project - Automated Host Reconnaissance

Team - **Jannatul Nayeem** & **Jash Vaidya**

Mentor **- Professor Dimple Chauhan**

Warning & Disclaimer

This project is strictly for educational and defensive purposes. We performed all scans on authorized testing targets like scanme.nmap.org and local labs. No unauthorized networks were accessed. The goal is to understand Vulnerability Assessment, not exploitation.

Overview

Project automates security scanning to save analyst time

Enhanced analysis

Security professionals can focus on analyzing results rather than collecting data. This improves overall assessment quality and thoroughness.

Nmap integration

We leverage the powerful Nmap Scripting Engine to execute all security analysis tasks simultaneously without manual intervention.

Automated workflow

The project combines port scanning, software version checking, and website identification into a single automated process.

Time efficiency

Security analysts typically perform multiple manual steps during assessment. Our solution consolidates these tasks to save valuable time.

Project requirements

Hardware requirements

- PC or Laptop
- Minimum 4 GB RAM

Software requirements

- OS: Linux / Windows
- Nmap (v7.x or above)
- Text Editor
- Admin / Root Privileges

Knowledge requirements

- Basic networking concepts
- Basic Lua scripting
- Understanding of Nmap

Tool Insight: Nmap Scripting Engine (NSE)

What is it?
Actually, NSE is the most powerful and flexible part of Nmap. It allows users to write simple scripts to automate a wide variety of networking tasks. Instead of just seeing that a port is open, NSE lets us interact with that port to gather deep intelligence.

The Core Technology

Language

It uses a tiny, fast programming language called Lua.

Libraries

It comes with pre-written "books" of code for things like HTTP, SSL, and SMB, which is why our script can talk to web servers so easily.

Automation

It runs these scripts in parallel, meaning it can scan hundreds of targets at once without slowing down.

Introduction - The Concept: Smart reconnaissance for effective hacking

Information Gathering is the most critical phase of hacking or auditing. If you miss an open door here, you miss it forever. Our Solution: We built a script called masterrecon.nse that adapts to the target.

This "smart" scanner unifies the reconnaissance process by intelligently responding to different services. It acts like a web browser for web servers and identifies versions for databases, creating a comprehensive information gathering tool.

NSE

Custom code for advanced Nmap tasks

Efficient operation
Runs directly inside the Nmap scan process

Lua scripting
Write custom code in Lua programming language

Custom solutions
Create personalized tools instead of using defaults

Advanced functionality
Extends Nmap beyond basic port scanning capabilities

Flexible implementation
Adapt scanning logic to specific security requirements

The Script Logic

Our masterrecon.nse script includes metadata and library imports, with key components.

File Edit Search View Document Help

```lua
-- Metadata and Library Imports
local http = require "http"
local stdnse = require "stdnse"
local shortport = require "shortport"

description = [[
Design a NSE script for collecting host or network based information
Metbrains Internship project for deep host reconnaissance
]]

author = "Jannatul Nayeem and Jash Vaidya"
license = "Same as Nmap"
categories = {"discovery", "safe"}

-- The Rule: Trigger on any open port
portrule = function(host, port)
  return port.state == "open"
end

-- The Action: Collecting the data
action = function(host, port)
  local gather = {}
  table.insert(gather, "Target IP " .. host.ip)

  if port.version and port.version.product then
    table.insert(gather, "Service " .. port.version.product)
    table.insert(gather, "Version " .. (port.version.version or "Unknown"))
  end

  if shortport.http(host, port) then
    local res = http.get(host, port, "/")
    if res and res.body then
      local title = string.match(res.body, "<title>(.-)</title>")
      if title then
```

*~/masterrecon.nse - Mousepad

File Edit Search View Document Help

```lua
license = "Same as Nmap"
categories = {"discovery", "safe"}

-- The Rule: Trigger on any open port
portrule = function(host, port)
  return port.state == "open"
end

-- The Action: Collecting the data
action = function(host, port)
  local gather = {}
  table.insert(gather, "Target IP " .. host.ip)

  if port.version and port.version.product then
    table.insert(gather, "Service " .. port.version.product)
    table.insert(gather, "Version " .. (port.version.version or "Unknown"))
  end

  if shortport.http(host, port) then
    local res = http.get(host, port, "/")
    if res and res.body then
      local title = string.match(res.body, "<title>(.-)</title>")
      if title then
        table.insert(gather, "Web Title " .. title)
      end
    end

    local rob = http.get(host, port, "/robots.txt")
    if rob and rob.status == 200 then
      table.insert(gather, "Found robots.txt file")
    end
  end

  return stdnse.format_output(true, gather)
end
```

Action function collects comprehensive target data

Portrule triggers on any open port

</>

Requires http, stdnse, and shortport libraries

Design Logic

Three Key Approaches to identifying vulnerabilities in servers.

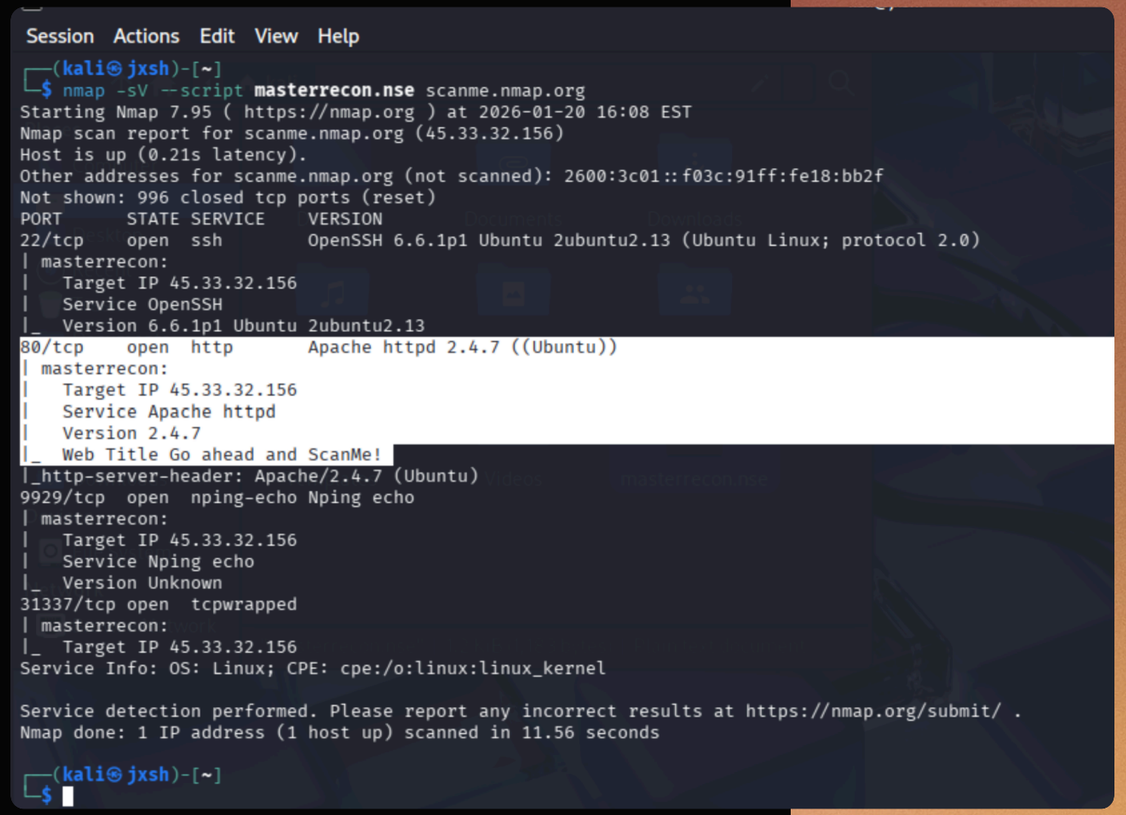
Version Check: Utilizing port.version.product functionality to extract software names, which is essential for identifying vulnerable older servers that may require security updates or patching.

Smart Web Check: Utilizing shortport.http to identify websites, capture titles, and find hidden folders via robots.txt.

The Broad Net: Not limiting to just Port 80. Our script activates when any port is open.

Results

Output analysis and results from testing



```
Session  Actions  Edit  View  Help
┌──(kali㉿jxsh)-[~]
└─$ nmap -sV --script masterrecon.nse scanme.nmap.org
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-20 16:08 EST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.21s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| masterrecon:
|   Target IP 45.33.32.156
|   Service OpenSSH
|_  Version 6.6.1p1 Ubuntu 2ubuntu2.13
80/tcp    open  http       Apache httpd 2.4.7 ((Ubuntu))
| masterrecon:
|   Target IP 45.33.32.156
|   Service Apache httpd
|   Version 2.4.7
|_  Web Title Go ahead and ScanMe!
|_http-server-header: Apache/2.4.7 (Ubuntu)
9929/tcp  open  nping-echo Nping echo
| masterrecon:
|   Target IP 45.33.32.156
|   Service Nping echo
|_  Version Unknown
31337/tcp open  tcpwrapped
| masterrecon:
|_  Target IP 45.33.32.156
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.56 seconds

┌──(kali㉿jxsh)-[~]
└─$
```

Hidden content

The script flagged the existence of robots.txt file, proving its ability to discover potentially sensitive directories and files.

Target verification

We confirmed the IP address of scanme.nmap.org, ensuring we were scanning the correct target for our testing process.

Service detection

The script successfully identified Apache 2.4.7 running on the target, providing crucial service version information for assessment.

Web reconnaissance

Our tool extracted the page title "Go ahead and ScanMe!" demonstrating effective content extraction capabilities.

Conclusion & Final Verdict

This project demonstrates that custom NSE scripts are superior to manual scanning. We successfully:
1. Automated the collection of host data.
2. Implemented conditional logic (checking for web vs non-web services).
3. Created a tool that is ready for real-world security auditing.

Thank You & Acknowledgments

A special thanks to Professor Dimple Chauhan for guiding us through the complexities of network scripting.