Project Report: Network Monitoring and Firewall Setup

Bidhan Adhikari

Haaga-Helia University of Applied Sciences
Degree Name: BIT
Report type: Linux Final Project
Report Completion Year:2024

Abstract

Author(s) Bidhan Adhikari
Degree Bachelor of Business Administration
Report/Thesis Title **Network monitoring and firewall setup using apache2, php and nmap**
Number of pages and appendix pages x + y
Key words The abstract ends with a list of keywords, 3–6 words that best describe the contents of your thesis, in order of importance. Make use of glossaries available at http://finto.fi/fi/ and https://annif.org/

Table of Contents

1. Introduction

The main object of the project is to create a server environment that scans the network to which we are connected and provides a full-detail view by displaying it through a PHP web interface and ensuring the network is protected by the firewall. To create this project we must have some knowledge about the Linux environment, commands and the Linux interface. Before, Starting here are the following objectives, Tools and Technologies, and software requirement that we are using while creating a Project.

1.1 Objective:

- Host a web interface with a real-time network scan.
- Automate periodic network scans using cron jobs.
- UFW firewall for secure connections.
- Provide detailed network, device details, and potential risks.

1.2 Tools and Technologies:

1. Apache2: To build a secure, efficient and extensible HTTP server as standards-compliant open source software,
2. PHP: Server-side scripting language for dynamic content generation and displaying network scan results.
3. Nmap: Network scanning tool to identify devices and open ports on the network.
4. Cron Jobs: Scheduling Nmap scans every 10 minutes.
5. UFW (Uncomplicated Firewall): A simple firewall for securing the server and network

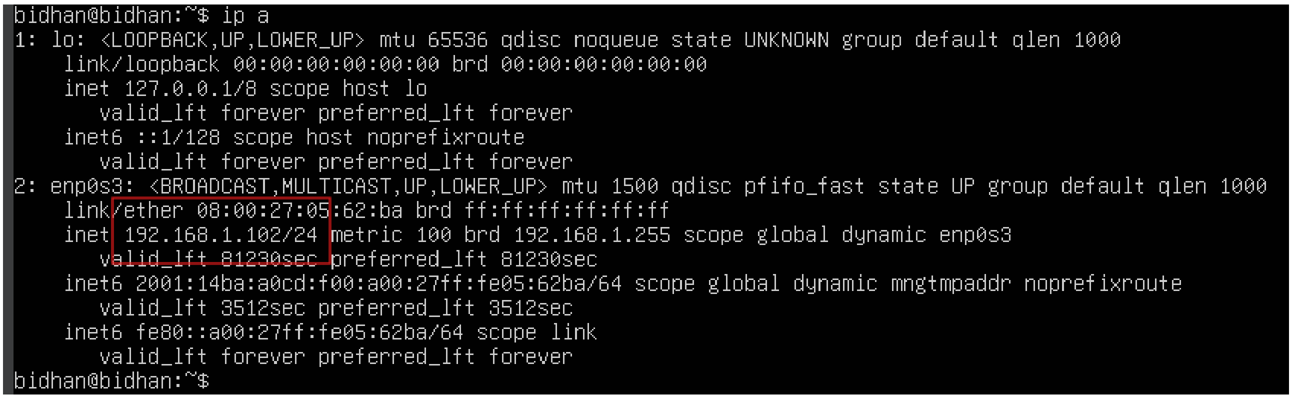
1.3 System Requirements

1. Operating System: Ubuntu 20.04 LTS or higher (Linux).
2. Hardware: VirtualBox running Ubuntu or a Linux server.
3. Softwares:
    - Apache2 Web Server
    - PHP 7.4 or higher
    - Nmap tool
    - Cron for scheduling tasks
    - UFW for firewall management.

2. Installation:

2.1 Checking IP Address:

Before starting with the Project Check your IP Address using following command:



```
bidhan@bidhan:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:05:62:ba brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.102/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
       valid_lft 81230sec preferred_lft 81230sec
    inet6 2001:14ba:a0cd:f00:a00:27ff:fe05:62ba/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 3512sec preferred_lft 3512sec
    inet6 fe80::a00:27ff:fe05:62ba/64 scope link
       valid_lft forever preferred_lft forever
bidhan@bidhan:~$
```

Figure 1: IP Address

2.2 Installing Apache2 Web Server:

Step1:

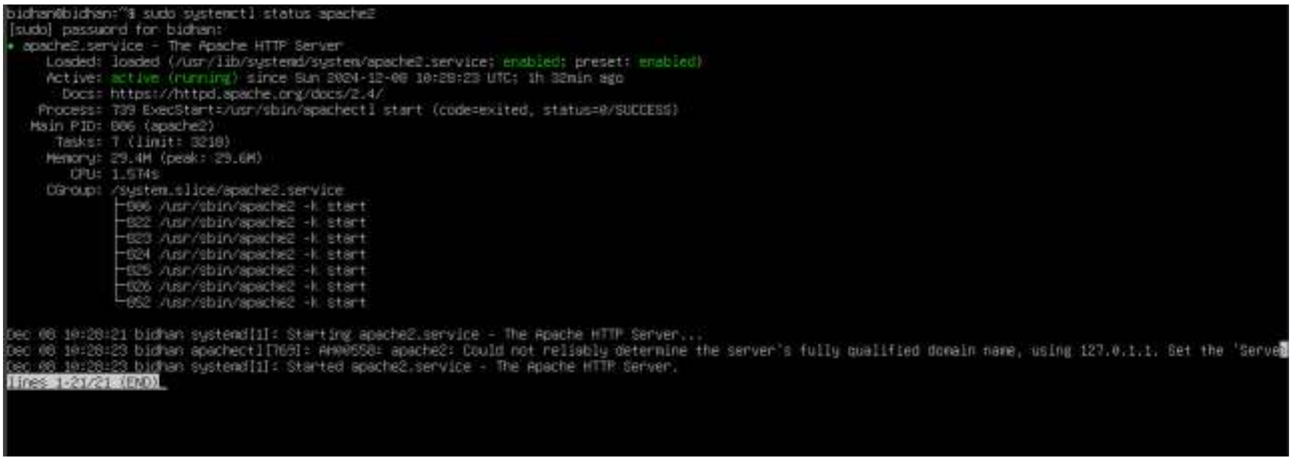
```
sudo apt-get update

sudo apt-get upgrade
```

Step2:

```
sudo apt-get install apache2

sudo systemctl status apache2
```

Step3:

After installing the apche2 verify weather apache2 is running or not



```
bidhan@bidhan:~$ sudo systemctl status apache2
[sudo] password for bidhan:
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-12-08 10:28:23 UTC; 1h 32min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 739 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 806 (apache2)
      Tasks: 7 (limit: 3218)
     Memory: 29.4M (peak: 29.6M)
        CPU: 1.574s
     CGroup: /system.slice/apache2.service
             ├─806 /usr/sbin/apache2 -k start
             ├─822 /usr/sbin/apache2 -k start
             ├─823 /usr/sbin/apache2 -k start
             ├─824 /usr/sbin/apache2 -k start
             ├─825 /usr/sbin/apache2 -k start
             ├─826 /usr/sbin/apache2 -k start
             └─852 /usr/sbin/apache2 -k start

Dec 08 10:28:21 bidhan systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 08 10:28:23 bidhan apachectl[769]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'Serve
Dec 08 10:28:23 bidhan systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-21/21 (END)
```

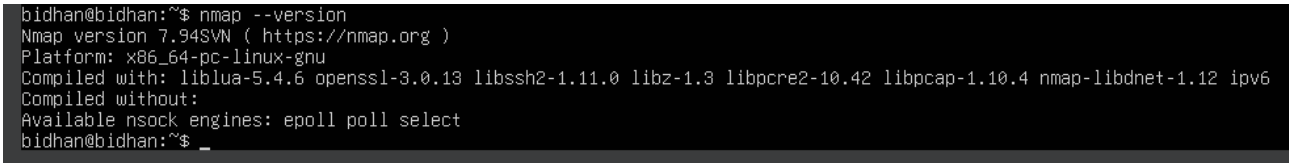
2.3 Installing PHP

```
sudo apt-get install PHP

php --version
```

2.4 Installing nmap

```
sudo apt-get install nmap

nmap --version
```

After installing Nmap we can check nmpa by using nmap –version command

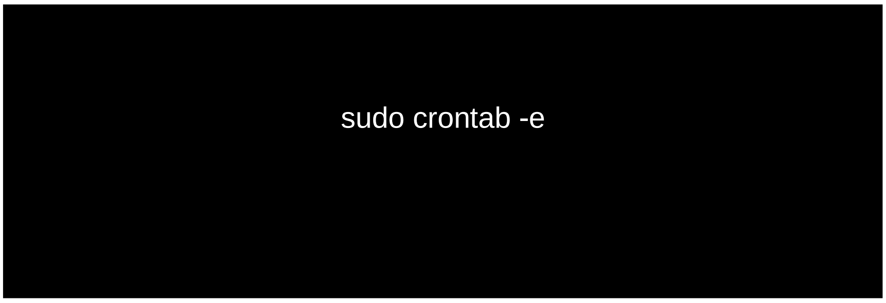


```
bidhan@bidhan:~$ nmap --version
Nmap version 7.94SVN ( https://nmap.org )
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.4.6 openssl-3.0.13 libssh2-1.11.0 libz-1.3 libpcre2-10.42 libpcap-1.10.4 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
bidhan@bidhan:~$ _
```

3. Cron Jobs

A Cron Job is a Linux program that allows users to schedule the execution of a piece of software, often in the form of a shell script or a compiled executable. Cron is typically used when you have a task that needs to be run on a fixed schedule, and/or to automate repetitive tasks like downloading files or sending emails. The cron job is used in this project to automate the Nmap scan at regular intervals, so you don't have to manually trigger the scan every time you want to get updated results.

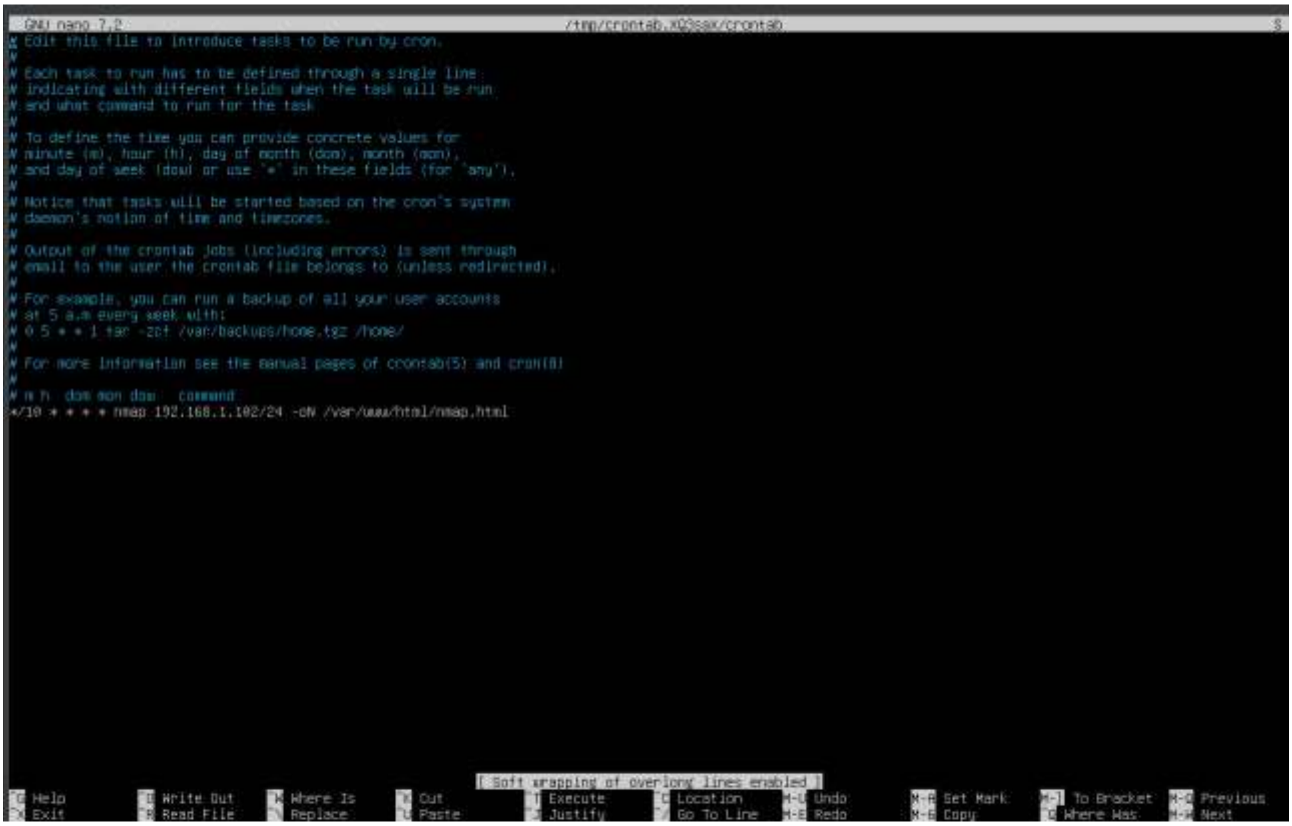
3.1 Configure Cron jobs



```
sudo crontab -e
```

After this step we should have to configure the content inside the crontab and add following line inside the crontab file so that nmap will run in every 10 Minutes

***/10 * * * * nmap 192.168.1.0/24 -oN /var/www/html/nmap.html**



```
  GNU nano 7.2                                   /tmp/crontab.xQ3sax/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/10 * * * * nmap 192.168.1.102/24 -oN /var/www/html/nmap.html

                                        [ Soft wrapping of overlong lines enabled ]
```

Figure 2: Cron Tab configure

4. Creating a Web interface

4.1 Step 1:

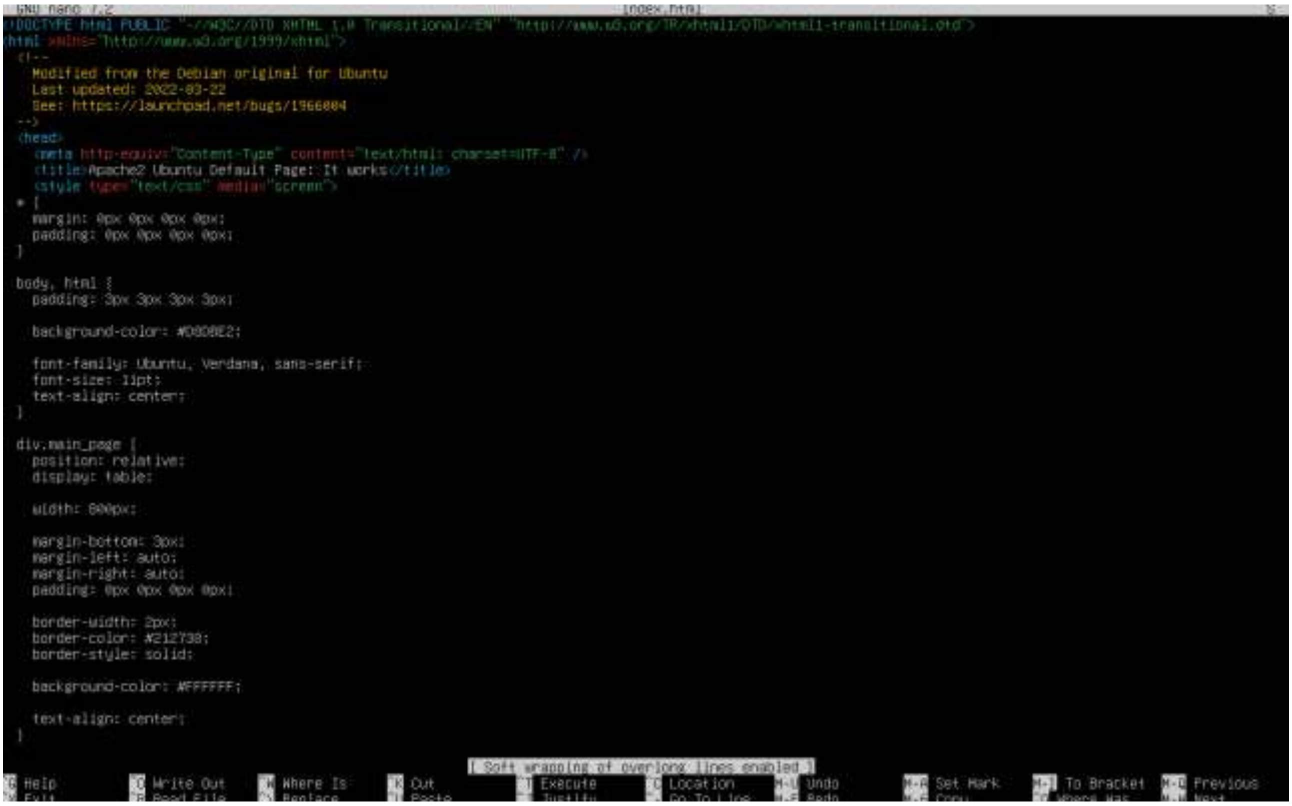
We have to create a file inside this directory : /var/www/html/

4.1.1 Steps to Follow:

- Cd ..
- Cd /var/www/html
- Sudo nano network.php

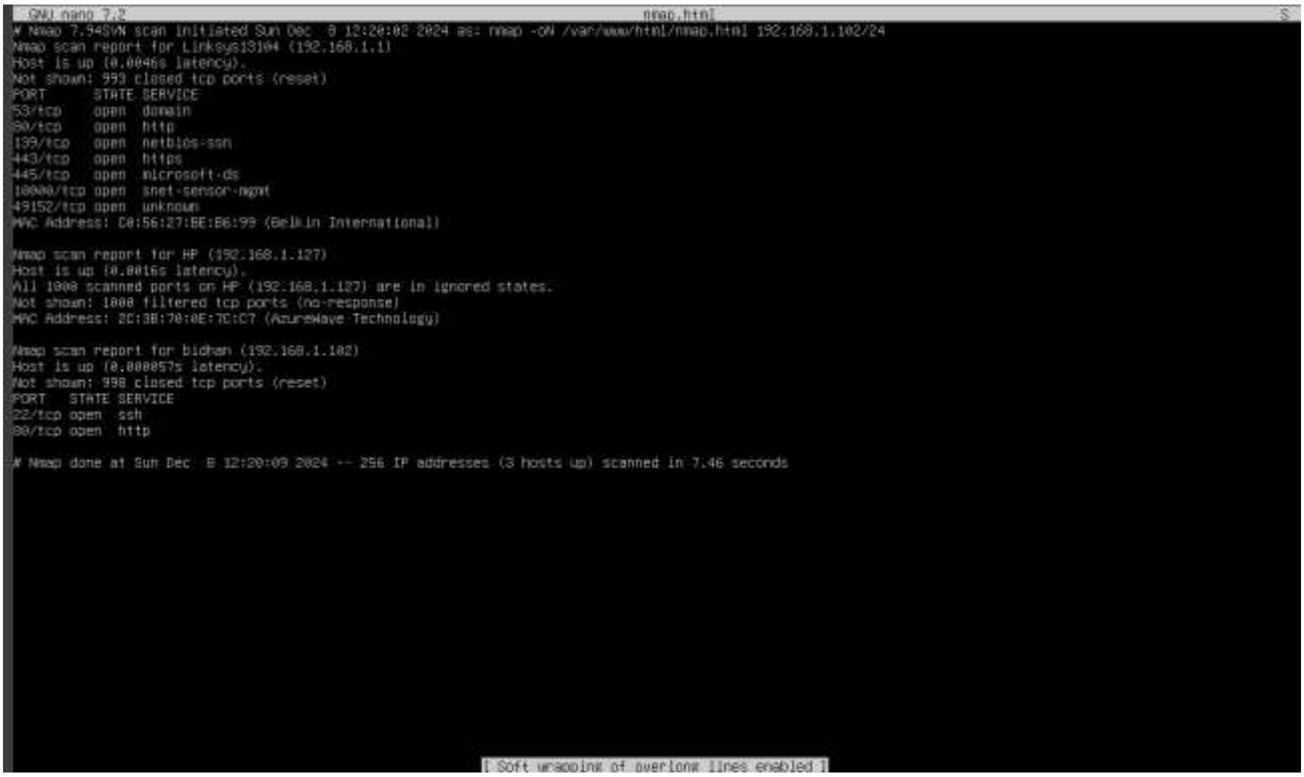
4.2 Step 2:

Before creating a Script lets naviagte to index.html and nmap.html



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }

  body, html {
    padding: 3px 3px 3px 3px;

    background-color: #D8DBE2;

    font-family: Ubuntu, Verdana, sans-serif;
    font-size: 11pt;
    text-align: center;
  }

  div.main_page {
    position: relative;
    display: table;

    width: 800px;

    margin-bottom: 3px;
    margin-left: auto;
    margin-right: auto;
    padding: 0px 0px 0px 0px;

    border-width: 2px;
    border-color: #212738;
    border-style: solid;

    background-color: #FFFFFF;

    text-align: center;
  }
```

Figure 3: Index.html

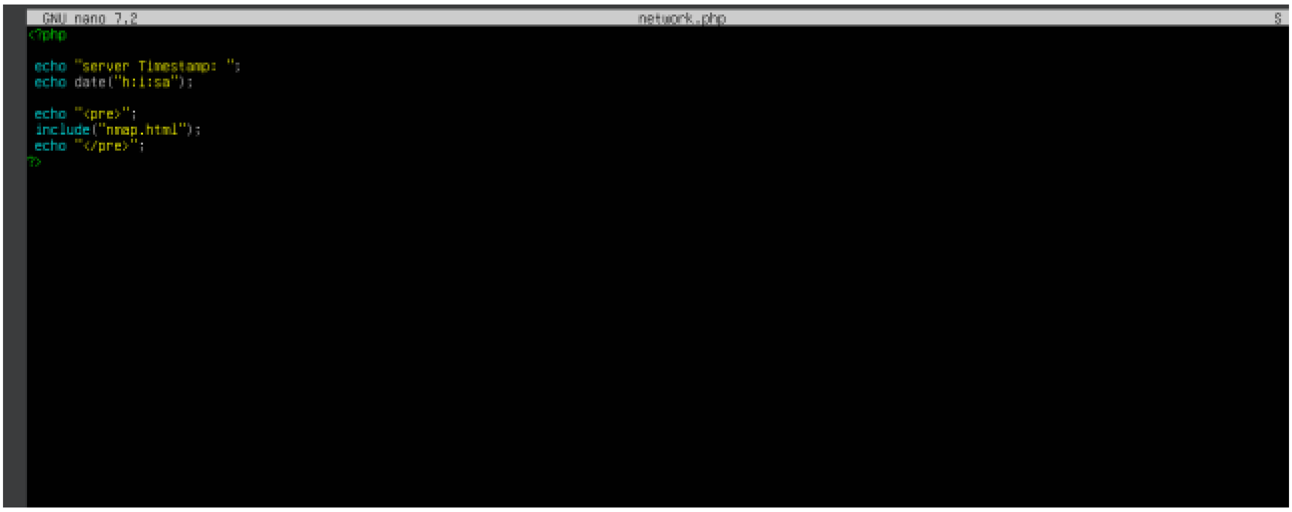


```
  GNU nano 7.2                                    nmap.html                                         S
# Nmap 7.94SVN scan initiated Sun Dec  8 12:20:02 2024 as: nmap -oN /var/www/html/nmap.html 192.168.1.102/24
Nmap scan report for Linksys13104 (192.168.1.1)
Host is up (0.0046s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
10000/tcp open  snet-sensor-mgmt
49152/tcp open  unknown
MAC Address: C0:56:27:BE:B6:99 (Belkin International)

Nmap scan report for HP (192.168.1.127)
Host is up (0.0016s latency).
All 1000 scanned ports on HP (192.168.1.127) are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 2C:3B:70:0E:7C:C7 (AzureWave Technology)

Nmap scan report for bidhan (192.168.1.102)
Host is up (0.000857s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

# Nmap done at Sun Dec  8 12:20:09 2024 -- 256 IP addresses (3 hosts up) scanned in 7.46 seconds

                              [ Soft wrapping of overlong lines enabled ]
```

Figure 4: namp.html

4.3 Step3

After creating a Php file we have to add some script inside the php file which will display the server timestamp and the result of the Nmap network scan.



```
  GNU nano 7.2                                    network.php                                       S
<?php

 echo "server Timestamp: ";
 echo date("h:i:sa");

 echo "<pre>";
 include("nmap.html");
 echo "</pre>";
?>
```

Figure 5: Network.php

The above script shows the output to the webpage. In the above script, we are using a function date("h:i:sa"): This function generates the current time on the server in a specific format:

- h: Hour (12-hour format)
- i: Minute
- s: Second
- a: AM/PM indicator

include("nmap.html");: This command includes and displays the content of the nmap.html file located in the /var/www/html/ directory. This file contains the output of the Nmap scan (generated by the cron job)

5. Firewall:

5.1 Introduction

UFW (Uncomplicated Firewall) is a front-end for iptables and is particularly well-suited for host-based firewalls. UFW was developed specifically for Ubuntu (but is available in other distributions), and is also configured from the terminal. This firewall is used to control incoming and outgoing traffic based on security rules. Moreover, this is used to restrict certain IP address networks and ranges. In this project, we used UFW to allow only essential services like HTTP (port 80) and SSH (port 22) while blocking all other incoming traffic.

5.2 Installing UFW

To download the UFW we should have to use the following command to download the Firewall

```
sudo apt-get install ufw
```

5.3 Configure ufw

After downloading the firewall the next step is to configure the UFW so that we can allow necessary services. This command is used to allow traffic only for the specific services by their port numbers and protocol.

```
sudo ufw allow 22/tcp

sudo ufw allow 80/tcp

sudo ufw allow 443/tcp
```

- SSH (22): This allows us to manage the server remotely
- HTTP (80): HTTP allows us to access web interface without encryption.
- HTTPS (443): HTTPS allows us to make a secure access to web interface.

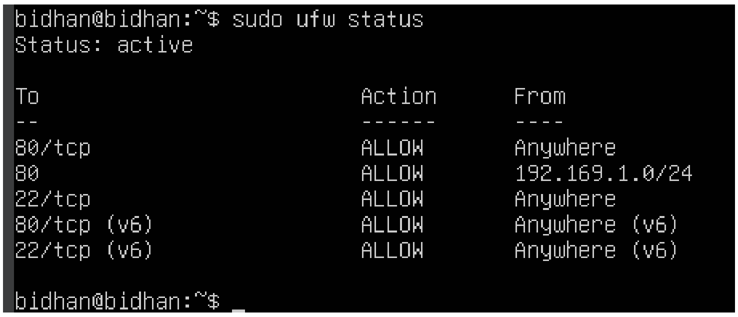
5.4 Default Firewall Configuration

These steps include configuring the incoming and outgoing traffic from the server by default and unless expelicitly allowed by the rule. The main reason for using this command is to reject the potentially harmful traffic and make outgoing connections less risky.

```
sudo ufw default deny incoming

sudo ufw default allow outgoing
```

5.5 Enable firewall and Checking status of firewall

```
sudo ufw enable

sudo ufw status
```



```
bidhan@bidhan:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW       Anywhere
80                         ALLOW       192.169.1.0/24
22/tcp                     ALLOW       Anywhere
80/tcp (v6)                ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)

bidhan@bidhan:~$ _
```

Figure 6: Result

The output shows the UFW firewall is active with the following rules:

- HTTP (port 80): Allowed globally (IPv4/IPv6) and locally from 192.168.1.0/24.
- SSH (port 22): Allowed globally (IPv4/IPv6).

5.6 Steps to view Blocked Traffic

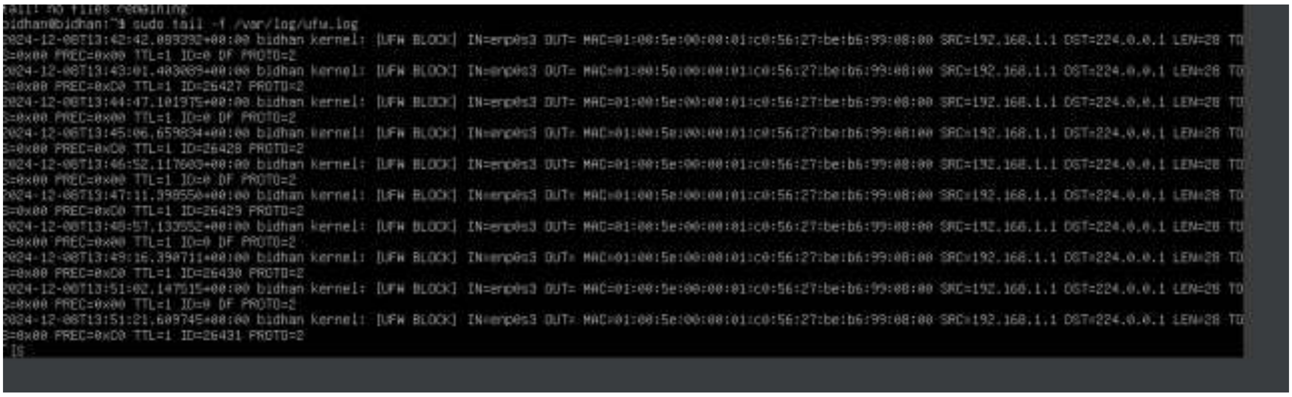
To view or checked the blocked incoming traffic we have to follow this steps:

Step 1:

sudo ufw logging on

Step 2:

sudo tail -f /var/log/ufw.log



```
tail: no files remaining
bidhan@bidhan:~$ sudo tail -f /var/log/ufw.log
2024-12-08T13:42:42.093392+00:00 bidhan kernel: [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:c0:56:27:be:b6:99:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TO
S=0x00 PREC=0x00 TTL=1 ID=0 DF PROTO=2
2024-12-08T13:43:01.403089+00:00 bidhan kernel: [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:c0:56:27:be:b6:99:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TO
S=0x00 PREC=0xC0 TTL=1 ID=26427 PROTO=2
2024-12-08T13:44:47.101975+00:00 bidhan kernel: [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:c0:56:27:be:b6:99:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TO
S=0x00 PREC=0x00 TTL=1 ID=0 DF PROTO=2
2024-12-08T13:45:06.659834+00:00 bidhan kernel: [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:c0:56:27:be:b6:99:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TO
S=0x00 PREC=0xC0 TTL=1 ID=26428 PROTO=2
2024-12-08T13:46:52.117603+00:00 bidhan kernel: [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:c0:56:27:be:b6:99:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TO
S=0x00 PREC=0x00 TTL=1 ID=0 DF PROTO=2
2024-12-08T13:47:11.398550+00:00 bidhan kernel: [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:c0:56:27:be:b6:99:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TO
S=0x00 PREC=0xC0 TTL=1 ID=26429 PROTO=2
2024-12-08T13:48:57.133552+00:00 bidhan kernel: [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:c0:56:27:be:b6:99:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TO
S=0x00 PREC=0x00 TTL=1 ID=0 DF PROTO=2
2024-12-08T13:49:16.390711+00:00 bidhan kernel: [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:c0:56:27:be:b6:99:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TO
S=0x00 PREC=0xC0 TTL=1 ID=26430 PROTO=2
2024-12-08T13:51:02.147515+00:00 bidhan kernel: [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:c0:56:27:be:b6:99:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TO
S=0x00 PREC=0x00 TTL=1 ID=0 DF PROTO=2
2024-12-08T13:51:21.609745+00:00 bidhan kernel: [UFW BLOCK] IN=enp0s3 OUT= MAC=01:00:5e:00:00:01:c0:56:27:be:b6:99:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TO
S=0x00 PREC=0xC0 TTL=1 ID=26431 PROTO=2
```

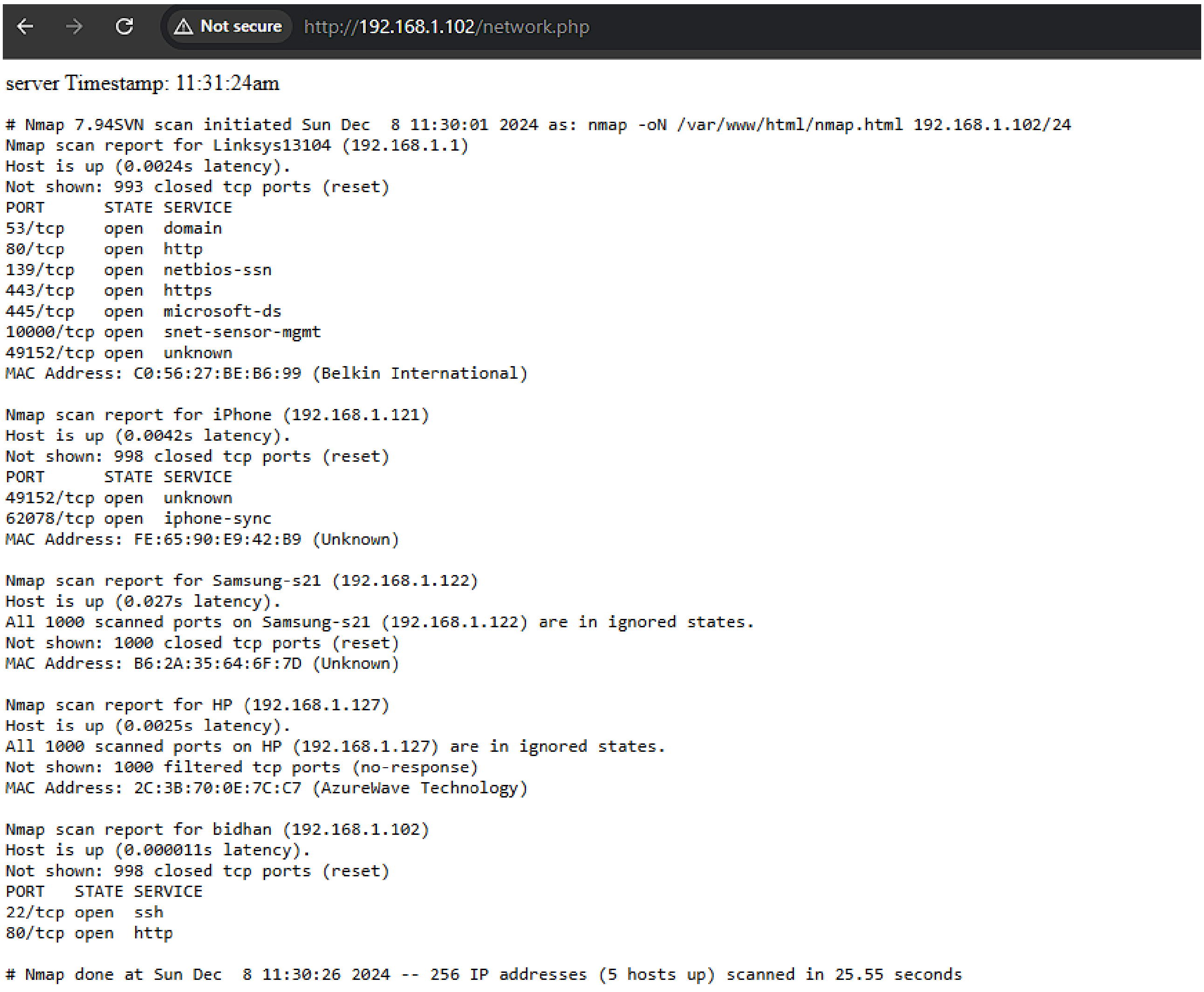
The above figure shows the firewall is working and blocking the incoming traffic to the assigned port.

6. Result

Before serving the page, ensure that the correct IP address has been used in all the steps to guarantee accurate results on the webpage. To access the server page, identify the IP address by running the following command: ip a. Note down the IP address and insert it into the URL as follows:

http://<server-ip>/network.php

http://192.168.1.102/network.php



server Timestamp: 11:31:24am

```
# Nmap 7.94SVN scan initiated Sun Dec  8 11:30:01 2024 as: nmap -oN /var/www/html/nmap.html 192.168.1.102/24
Nmap scan report for Linksys13104 (192.168.1.1)
Host is up (0.0024s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
10000/tcp open  snet-sensor-mgmt
49152/tcp open  unknown
MAC Address: C0:56:27:BE:B6:99 (Belkin International)

Nmap scan report for iPhone (192.168.1.121)
Host is up (0.0042s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
49152/tcp open  unknown
62078/tcp open  iphone-sync
MAC Address: FE:65:90:E9:42:B9 (Unknown)

Nmap scan report for Samsung-s21 (192.168.1.122)
Host is up (0.027s latency).
All 1000 scanned ports on Samsung-s21 (192.168.1.122) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: B6:2A:35:64:6F:7D (Unknown)

Nmap scan report for HP (192.168.1.127)
Host is up (0.0025s latency).
All 1000 scanned ports on HP (192.168.1.127) are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 2C:3B:70:0E:7C:C7 (AzureWave Technology)

Nmap scan report for bidhan (192.168.1.102)
Host is up (0.000011s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

# Nmap done at Sun Dec  8 11:30:26 2024 -- 256 IP addresses (5 hosts up) scanned in 25.55 seconds
```

Figure 7: Final result

6.1 Result Description

The above screenshots provide a brief result of an nmap scan conducted on 192.168.1.102/24. This scan result shows the number of devices connected over the network and displays their ip address, ports and Mac address. Moreover, at the top of the display, we can see the time we access the webpage with timestamps. This page has detected 5 hosts as up on the network and some devices like Samsungs don’t have visible ports ikely due to strict firewall settings or inactive services. This detailed output is valuable for identifying connected devices, analyzing open ports, and assessing potential vulnerabilities within the network.

In conclusion:

- The Nmap scan was performed on the 192.168.1.102/24 network
- A total of 5 devices are detected with their IP address and Mac address.
- Proper use of UFW firewall to block incoming traffic and ensure a high level of security.
- Nmap scans the network in every 10 minutes and provide details in nmap.html
- Network.php displays the result of the Nmap scan providing real-time data.
- Integration of a firewall to protect the server from unauthorized access

Sources

https://www.kali.org/tools/apache2/#:~:text=The%20Apache%20HTTP%20Server%20Project's,web%20server%20on%20the%20Internet.

https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/installing_and_using_dynamic_programming_languages/assembly_using-the-php-scripting-language_installing-and-using-dynamic-programming-languages

https://cronitor.io/guides/cron-jobs

https://help.ubuntu.com/community/Firewall

https://www.digitalocean.com/community/tutorials/ufw-essentials-common-firewall-rules-and-commands

https://www.tecmint.com/run-php-codes-from-linux-commandline/

Appendices

Appendix 1. xxx