Name: Pelumi Johnson
Date: 01/06/2026
Course / Platform: Self-Directed Cybersecurity Lab
Tool: Bettercap
Operating System: Ubuntu (Virtual Machine – Bridged Networking)

Objective

The objective of this lab was to understand how devices communicate on a local network by **passively observing network traffic** using Bettercap. This lab focused on learning how devices announce themselves, how network connections begin, and how encryption affects what information is visible on the network.

No active attacks, impersonation, or interference were performed during this lab.

Tools & Environment Used

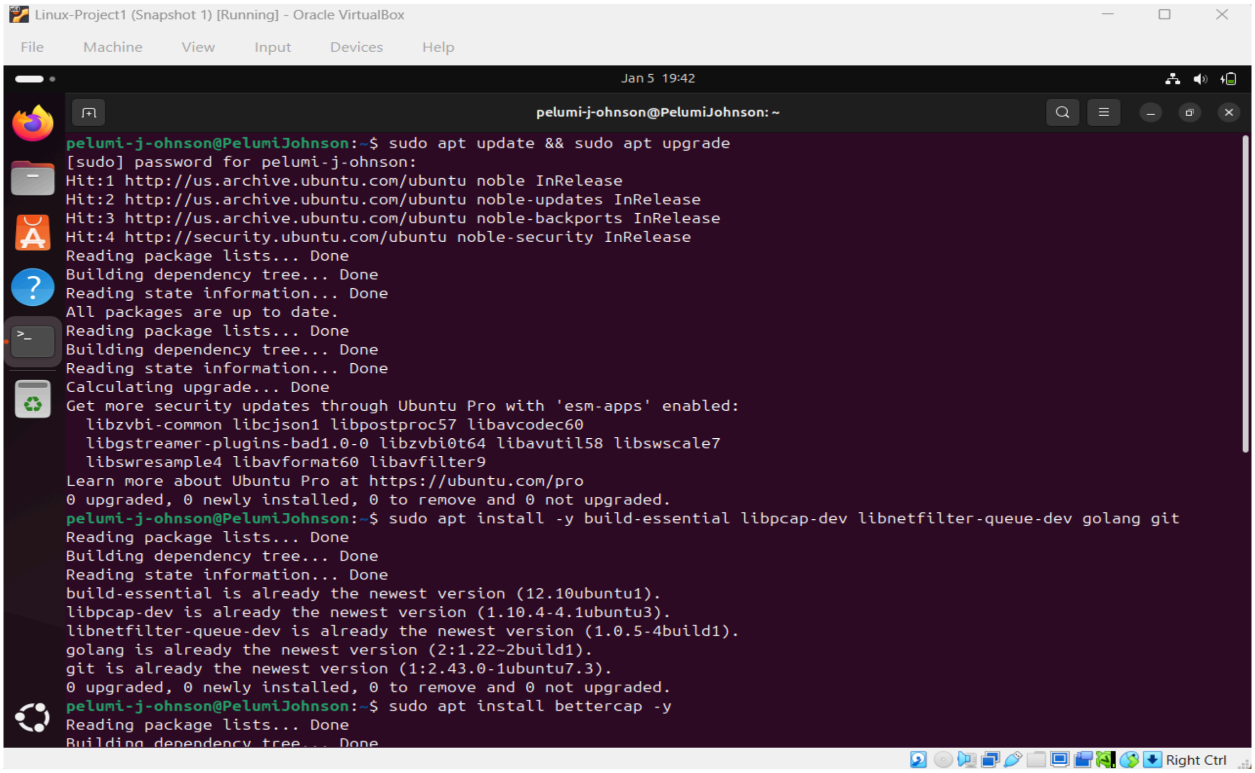
- Ubuntu Linux (Virtual Machine)
- Oracle VirtualBox (Bridged Network Mode)
- Bettercap
- Local Home Network (Observation Only)

Step 1: Network Setup Verification

Before starting the lab, I verified that the Ubuntu virtual machine was using **bridged networking**.
This allowed the VM to receive its own IP address on the same network as the host machine.

I confirmed the IP address using the `ip a` command and verified that the VM had a unique IP in the `192.168.1.0/24` range.

This step ensured the virtual machine behaved as a separate device on the network.

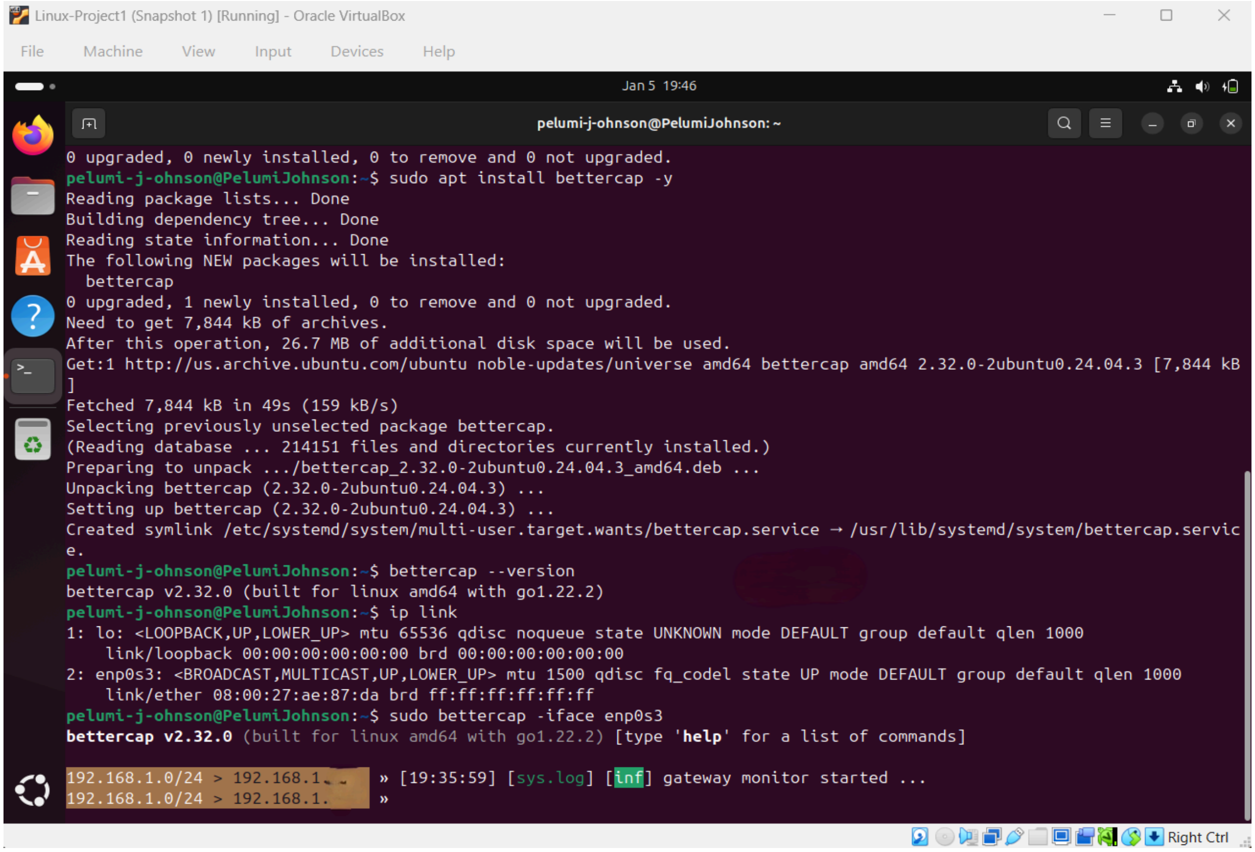

Linux-Project1 (Snapshot 1) [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
Jan 5 19:42
pelumi-j-ohnson@PelumiJohnson: ~
Right Ctrl

```
pelumi-j-ohnson@PelumiJohnson:~$ sudo apt update && sudo apt upgrade
[sudo] password for pelumi-j-ohnson:
Hit:1 http://us.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
  libzvbi-common libcjson1 libpostproc57 libavcodec60
  libgstreamer-plugins-bad1.0-0 libzvbi0t64 libavutil58 libswscale7
  libswresample4 libavformat60 libavfilter9
Learn more about Ubuntu Pro at https://ubuntu.com/pro
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pelumi-j-ohnson@PelumiJohnson:~$ sudo apt install -y build-essential libpcap-dev libnetfilter-queue-dev golang git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.10ubuntu1).
libpcap-dev is already the newest version (1.10.4-4.1ubuntu3).
libnetfilter-queue-dev is already the newest version (1.0.5-4build1).
golang is already the newest version (2:1.22~2build1).
git is already the newest version (1:2.43.0-1ubuntu7.3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pelumi-j-ohnson@PelumiJohnson:~$ sudo apt install bettercap -y
Reading package lists... Done
Building dependency tree... Done
```

Step 2: Starting Bettercap

I launched Bettercap using the correct network interface associated with the virtual machine.
Bettercap was started in **interactive mode**, allowing commands to be issued manually.

Once Bettercap was running, no network activity occurred until specific modules were enabled.


Linux-Project1 (Snapshot 1) [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
Jan 5 19:46
pelumi-j-ohnson@PelumiJohnson: ~
Right Ctrl

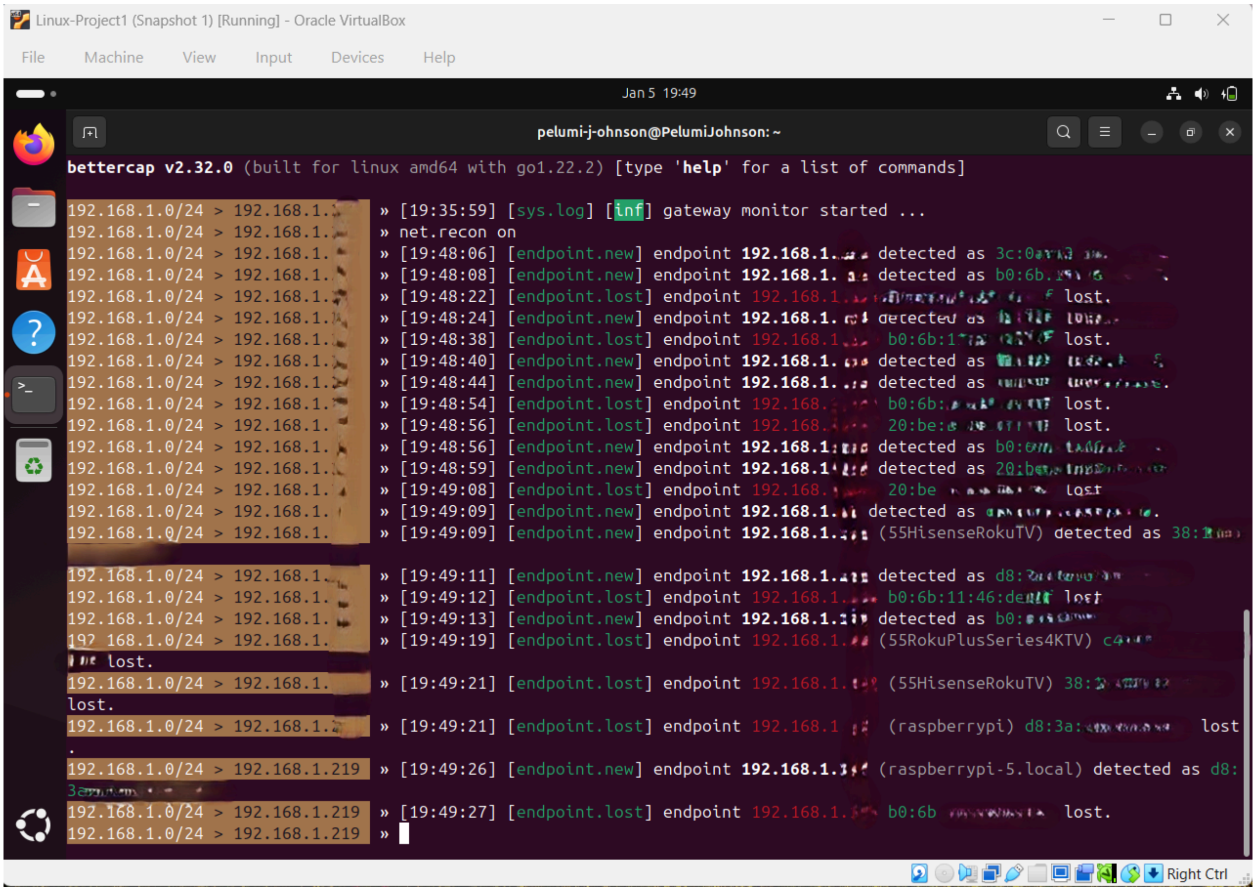
```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pelumi-j-ohnson@PelumiJohnson:~$ sudo apt install bettercap -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  bettercap
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,844 kB of archives.
After this operation, 26.7 MB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu noble-updates/universe amd64 bettercap amd64 2.32.0-2ubuntu0.24.04.3 [7,844 kB
]
Fetched 7,844 kB in 49s (159 kB/s)
Selecting previously unselected package bettercap.
(Reading database ... 214151 files and directories currently installed.)
Preparing to unpack .../bettercap_2.32.0-2ubuntu0.24.04.3_amd64.deb ...
Unpacking bettercap (2.32.0-2ubuntu0.24.04.3) ...
Setting up bettercap (2.32.0-2ubuntu0.24.04.3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/bettercap.service → /usr/lib/systemd/system/bettercap.servic
e.
pelumi-j-ohnson@PelumiJohnson:~$ bettercap --version
bettercap v2.32.0 (built for linux amd64 with go1.22.2)
pelumi-j-ohnson@PelumiJohnson:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:ae:87:da brd ff:ff:ff:ff:ff:ff
pelumi-j-ohnson@PelumiJohnson:~$ sudo bettercap -iface enp0s3
bettercap v2.32.0 (built for linux amd64 with go1.22.2) [type 'help' for a list of commands]

192.168.1.0/24 > 192.168.1.[redacted]  » [19:35:59] [sys.log] [inf] gateway monitor started ...
192.168.1.0/24 > 192.168.1.[redacted]  »
```

Step 3: Passive Network Discovery (net.recon)

I enabled passive network reconnaissance using the net.recon module.

This allowed Bettercap to **listen** for existing network traffic without sending or modifying packets. As devices communicated normally on the network, Bettercap detected new endpoints and displayed their IP and MAC addresses.

Messages such as endpoint.new indicated when a device appeared on the network, while endpoint.lost showed when a device went offline or became inactive.



Linux-Project1 (Snapshot 1) [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
Jan 5 19:49
pelumi-j-ohnson@PelumiJohnson: ~
Right Ctrl

```
bettercap v2.32.0 (built for linux amd64 with go1.22.2) [type 'help' for a list of commands]

192.168.1.0/24 > 192.168.1.[redacted]  » [19:35:59] [sys.log] [inf] gateway monitor started ...
192.168.1.0/24 > 192.168.1.[redacted]  » net.recon on
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:06] [endpoint.new] endpoint 192.168.1.[redacted] detected as 3c:0[redacted]
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:08] [endpoint.new] endpoint 192.168.1.[redacted] detected as b0:6b.[redacted]
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:22] [endpoint.lost] endpoint 192.168.1.[redacted] lost.
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:24] [endpoint.new] endpoint 192.168.1.[redacted] detected as [redacted]
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:38] [endpoint.lost] endpoint 192.168.1.[redacted] b0:6b:1[redacted] lost.
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:40] [endpoint.new] endpoint 192.168.1.[redacted] detected as [redacted]
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:44] [endpoint.new] endpoint 192.168.1.[redacted] detected as [redacted].
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:54] [endpoint.lost] endpoint 192.168.[redacted] b0:6b:[redacted] lost.
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:56] [endpoint.lost] endpoint 192.168.[redacted] 20:be:[redacted] lost.
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:56] [endpoint.new] endpoint 192.168.1[redacted] detected as b0:[redacted]
192.168.1.0/24 > 192.168.1.[redacted]  » [19:48:59] [endpoint.new] endpoint 192.168.1[redacted] detected as 20:be[redacted]
192.168.1.0/24 > 192.168.1.[redacted]  » [19:49:08] [endpoint.lost] endpoint 192.168.[redacted] 20:be [redacted] lost
192.168.1.0/24 > 192.168.1.[redacted]  » [19:49:09] [endpoint.new] endpoint 192.168.1.[redacted] detected as [redacted].
192.168.1.0/24 > 192.168.1.[redacted]  » [19:49:09] [endpoint.new] endpoint 192.168.1.[redacted] (55HisenseRokuTV) detected as 38:[redacted]

192.168.1.0/24 > 192.168.1.[redacted]  » [19:49:11] [endpoint.new] endpoint 192.168.1.[redacted] detected as d8:[redacted]
192.168.1.0/24 > 192.168.1.[redacted]  » [19:49:12] [endpoint.lost] endpoint 192.168.1.[redacted] b0:6b:11:46:de[redacted] lost
192.168.1.0/24 > 192.168.1.[redacted]  » [19:49:13] [endpoint.new] endpoint 192.168.1.[redacted] detected as b0:[redacted]
192 168.1.0/24 > 192.168.1.[redacted]  » [19:49:19] [endpoint.lost] endpoint 192.168.1.[redacted] (55RokuPlusSeries4KTV) c4[redacted]
[redacted] lost.
192.168.1.0/24 > 192.168.1.[redacted]  » [19:49:21] [endpoint.lost] endpoint 192.168.1.[redacted] (55HisenseRokuTV) 38:[redacted]
lost.
192.168.1.0/24 > 192.168.1.[redacted]  » [19:49:21] [endpoint.lost] endpoint 192.168.1 [redacted] (raspberrypi) d8:3a:[redacted] lost
.
192.168.1.0/24 > 192.168.1.219  » [19:49:26] [endpoint.new] endpoint 192.168.1.[redacted] (raspberrypi-5.local) detected as d8:
3[redacted]
192.168.1.0/24 > 192.168.1.219  » [19:49:27] [endpoint.lost] endpoint 192.168.1.[redacted] b0:6b [redacted] lost.
192.168.1.0/24 > 192.168.1.219  »
```

Step 4: Observing Local Name Discovery (mDNS)

While listening to traffic, I observed frequent mdns messages.

These messages showed devices performing **local name discovery**, such as resolving names ending in .local (for example, printers, phones, and a Raspberry Pi).

This demonstrated how devices find each other on a local network without using a central DNS server.

Unknown query means:
The TV asked *“Who owns this name?”* and **no device responded**.

So it simply means:

- That device is **not connected**
- Or it’s **not on the network**

- Or it’s **not responding anymore**

PTR Query (Reverse Lookup)

A **PTR query** is a reverse lookup request used on local networks.

Simple meaning:
Instead of asking *“Who owns this name?”*, a device asks:
“What is the name of this device or service I see?”

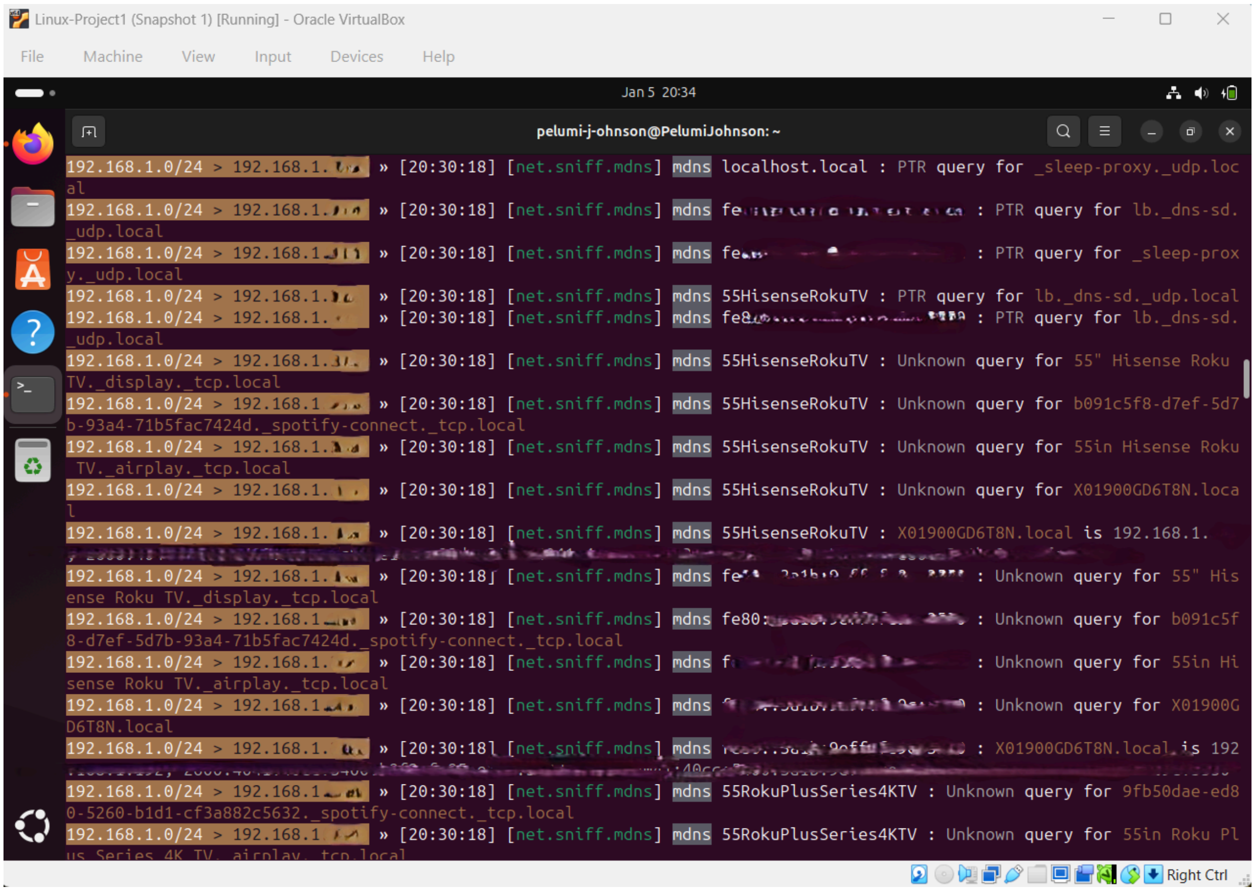
On local networks using mDNS, PTR queries are used to:

- Discover **what services exist** (such as printers, casting devices, or media services)
- Learn **friendly names** for devices that have already been detected

Response behavior:
If a device or service exists, it responds with its name and service type.
If no response is received, it means the service is not available or not responding.

In short:

- **Query** = asking a question
- **PTR query** = asking *“What are you called and what do you do?”*


Linux-Project1 (Snapshot 1) [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
Jan 5 20:34
pelumi-j-ohnson@PelumiJohnson: ~
Right Ctrl

```
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns localhost.local : PTR query for _sleep-proxy._udp.loc
al
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns fe[redacted] : PTR query for lb._dns-sd.
_udp.local
192.168.1.0/24 > 192.168.1[redacted] » [20:30:18] [net.sniff.mdns] mdns fe[redacted] : PTR query for _sleep-prox
y._udp.local
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns 55HisenseRokuTV : PTR query for lb._dns-sd._udp.local
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns fe8[redacted] : PTR query for lb._dns-sd.
_udp.local
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns 55HisenseRokuTV : Unknown query for 55" Hisense Roku
TV._display._tcp.local
192.168.1.0/24 > 192.168.1[redacted] » [20:30:18] [net.sniff.mdns] mdns 55HisenseRokuTV : Unknown query for b091c5f8-d7ef-5d7
b-93a4-71b5fac7424d._spotify-connect._tcp.local
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns 55HisenseRokuTV : Unknown query for 55in Hisense Roku
 TV._airplay._tcp.local
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns 55HisenseRokuTV : Unknown query for X01900GD6T8N.loca
l
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns 55HisenseRokuTV : X01900GD6T8N.local is 192.168.1.
[redacted]
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns fe[redacted] : Unknown query for 55" His
ense Roku TV._display._tcp.local
192.168.1.0/24 > 192.168.1[redacted] » [20:30:18] [net.sniff.mdns] mdns fe80:[redacted] : Unknown query for b091c5f
8-d7ef-5d7b-93a4-71b5fac7424d._spotify-connect._tcp.local
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns f[redacted] : Unknown query for 55in Hi
sense Roku TV._airplay._tcp.local
192.168.1.0/24 > 192.168.1[redacted] » [20:30:18] [net.sniff.mdns] mdns [redacted] : Unknown query for X01900G
D6T8N.local
192.168.1.0/24 > 192.168.1.[redacted] » [20:30:18] [net.sniff.mdns] mdns [redacted] : X01900GD6T8N.local is 192
[redacted]
192.168.1.0/24 > 192.168.1[redacted] » [20:30:18] [net.sniff.mdns] mdns 55RokuPlusSeries4KTV : Unknown query for 9fb50dae-ed8
0-5260-b1d1-cf3a882c5632._spotify-connect._tcp.local
192.168.1.0/24 > 192.168.1[redacted] » [20:30:18] [net.sniff.mdns] mdns 55RokuPlusSeries4KTV : Unknown query for 55in Roku Pl
us Series 4K TV._airplay._tcp.local
```

Step 5: Observing HTTP Traffic

When network sniffing was enabled, I observed **HTTP requests** made by devices on the network.

HTTP traffic appeared in clear text, allowing the destination website and request type (such as GET) to be visible.

This demonstrated how unencrypted traffic can be easily observed by any system listening on the network.

A query

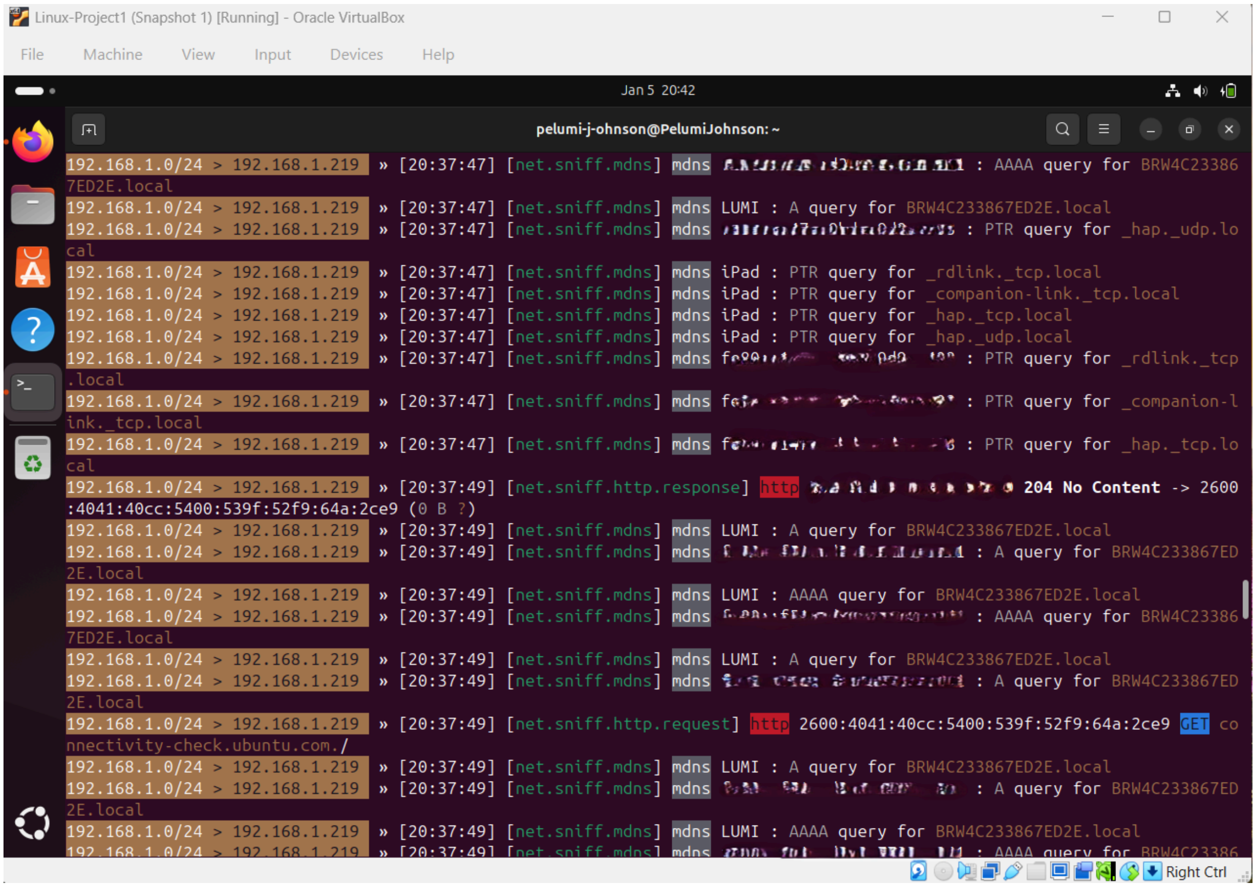
A device is asking: **“What is the IPv4 address for this name?”**

- IPv4 looks like: 192.168.1.192
- Older, shorter internet address format

AAAA query

A device is asking: **“What is the IPv6 address for this name?”**

- IPv6 looks like: 2600:4041:40cc:…
- Newer, longer internet address format


Linux-Project1 (Snapshot 1) [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
Jan 5 20:42
pelumi-j-ohnson@PelumiJohnson: ~
Right Ctrl

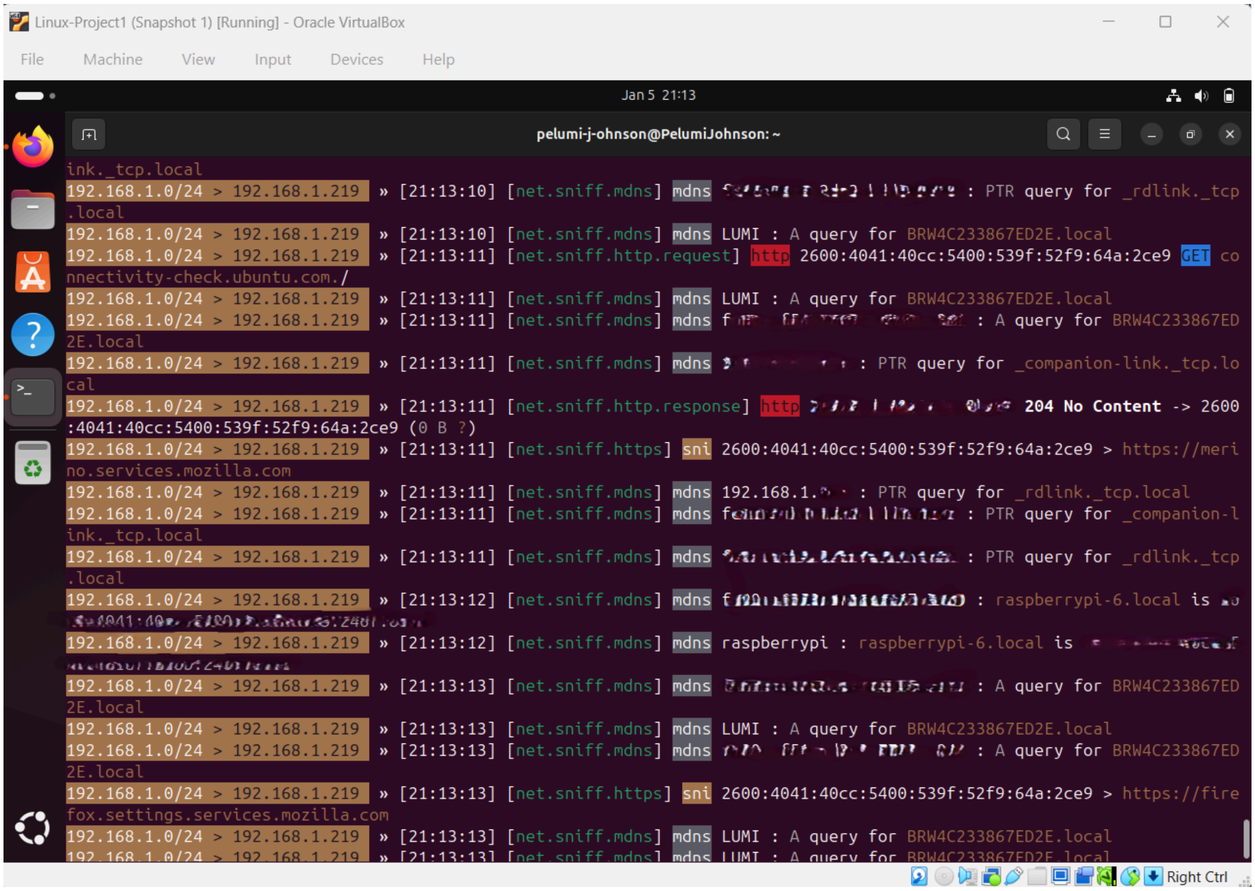
```
192.168.1.0/24 > 192.168.1.219  » [20:37:47] [net.sniff.mdns] mdns [redacted] : AAAA query for BRW4C23386
7ED2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:47] [net.sniff.mdns] mdns LUMI : A query for BRW4C233867ED2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:47] [net.sniff.mdns] mdns [redacted] : PTR query for _hap._udp.lo
cal
192.168.1.0/24 > 192.168.1.219  » [20:37:47] [net.sniff.mdns] mdns iPad : PTR query for _rdlink._tcp.local
192.168.1.0/24 > 192.168.1.219  » [20:37:47] [net.sniff.mdns] mdns iPad : PTR query for _companion-link._tcp.local
192.168.1.0/24 > 192.168.1.219  » [20:37:47] [net.sniff.mdns] mdns iPad : PTR query for _hap._tcp.local
192.168.1.0/24 > 192.168.1.219  » [20:37:47] [net.sniff.mdns] mdns iPad : PTR query for _hap._udp.local
192.168.1.0/24 > 192.168.1.219  » [20:37:47] [net.sniff.mdns] mdns [redacted] : PTR query for _rdlink._tcp
.local
192.168.1.0/24 > 192.168.1.219  » [20:37:47] [net.sniff.mdns] mdns [redacted] : PTR query for _companion-l
ink._tcp.local
192.168.1.0/24 > 192.168.1.219  » [20:37:47] [net.sniff.mdns] mdns [redacted] : PTR query for _hap._tcp.lo
cal
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.http.response] http [redacted] 204 No Content -> 2600
:4041:40cc:5400:539f:52f9:64a:2ce9 (0 B ?)
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.mdns] mdns LUMI : A query for BRW4C233867ED2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.mdns] mdns [redacted] : A query for BRW4C233867ED
2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.mdns] mdns LUMI : AAAA query for BRW4C233867ED2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.mdns] mdns [redacted] : AAAA query for BRW4C23386
7ED2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.mdns] mdns LUMI : A query for BRW4C233867ED2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.mdns] mdns [redacted] : A query for BRW4C233867ED
2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.http.request] http 2600:4041:40cc:5400:539f:52f9:64a:2ce9 GET co
nnectivity-check.ubuntu.com./
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.mdns] mdns LUMI : A query for BRW4C233867ED2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.mdns] mdns [redacted] : A query for BRW4C233867ED
2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.mdns] mdns LUMI : AAAA query for BRW4C233867ED2E.local
192.168.1.0/24 > 192.168.1.219  » [20:37:49] [net.sniff.mdns] mdns [redacted] : AAAA query for BRW4C23386
```

Step 6: Observing HTTPS Traffic and Encryption

I also observed **HTTPS connections**, which appeared differently from HTTP traffic.

While the destination domain name was visible using SNI (Server Name Indication), the actual content of the communication remained encrypted.

This demonstrated how encryption protects the data itself, even though some connection metadata remains visible.



Linux-Project1 (Snapshot 1) [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
Jan 5 21:13
pelumi-j-ohnson@PelumiJohnson: ~
Right Ctrl

```
ink._tcp.local
192.168.1.0/24 > 192.168.1.219  » [21:13:10] [net.sniff.mdns] mdns [redacted] : PTR query for _rdlink._tcp
.local
192.168.1.0/24 > 192.168.1.219  » [21:13:10] [net.sniff.mdns] mdns LUMI : A query for BRW4C233867ED2E.local
192.168.1.0/24 > 192.168.1.219  » [21:13:11] [net.sniff.http.request] http 2600:4041:40cc:5400:539f:52f9:64a:2ce9 GET co
nnectivity-check.ubuntu.com./
192.168.1.0/24 > 192.168.1.219  » [21:13:11] [net.sniff.mdns] mdns LUMI : A query for BRW4C233867ED2E.local
192.168.1.0/24 > 192.168.1.219  » [21:13:11] [net.sniff.mdns] mdns [redacted] : A query for BRW4C233867ED
2E.local
192.168.1.0/24 > 192.168.1.219  » [21:13:11] [net.sniff.mdns] mdns [redacted] : PTR query for _companion-link._tcp.lo
cal
192.168.1.0/24 > 192.168.1.219  » [21:13:11] [net.sniff.http.response] http [redacted] 204 No Content -> 2600
:4041:40cc:5400:539f:52f9:64a:2ce9 (0 B ?)
192.168.1.0/24 > 192.168.1.219  » [21:13:11] [net.sniff.https] sni 2600:4041:40cc:5400:539f:52f9:64a:2ce9 > https://meri
no.services.mozilla.com
192.168.1.0/24 > 192.168.1.219  » [21:13:11] [net.sniff.mdns] mdns 192.168.1.[redacted] : PTR query for _rdlink._tcp.local
192.168.1.0/24 > 192.168.1.219  » [21:13:11] [net.sniff.mdns] mdns [redacted] : PTR query for _companion-l
ink._tcp.local
192.168.1.0/24 > 192.168.1.219  » [21:13:11] [net.sniff.mdns] mdns [redacted] : PTR query for _rdlink._tcp
.local
192.168.1.0/24 > 192.168.1.219  » [21:13:12] [net.sniff.mdns] mdns [redacted] : raspberrypi-6.local is [redacted]
192.168.1.0/24 > 192.168.1.219  » [21:13:12] [net.sniff.mdns] mdns raspberrypi : raspberrypi-6.local is [redacted]
192.168.1.0/24 > 192.168.1.219  » [21:13:13] [net.sniff.mdns] mdns [redacted] : A query for BRW4C233867ED
2E.local
192.168.1.0/24 > 192.168.1.219  » [21:13:13] [net.sniff.mdns] mdns LUMI : A query for BRW4C233867ED2E.local
192.168.1.0/24 > 192.168.1.219  » [21:13:13] [net.sniff.mdns] mdns [redacted] : A query for BRW4C233867ED
2E.local
192.168.1.0/24 > 192.168.1.219  » [21:13:13] [net.sniff.https] sni 2600:4041:40cc:5400:539f:52f9:64a:2ce9 > https://fire
fox.settings.services.mozilla.com
192.168.1.0/24 > 192.168.1.219  » [21:13:13] [net.sniff.mdns] mdns LUMI : A query for BRW4C233867ED2E.local
192.168.1.0/24 > 192.168.1.219  » [21:13:13] [net.sniff.mdns] mdns LUMI : A query for BRW4C233867ED2E.local
```

Step 7: Understanding Device Visibility

Throughout the lab, multiple devices such as phones, tablets, and a Raspberry Pi appeared and disappeared in the logs.

This occurred naturally as devices connected, disconnected, or entered low-power states.

This reinforced the concept that devices continuously announce their presence on a network as part of normal operation.

Completion

This lab successfully demonstrated how network traffic can be observed **passively** without performing any active attacks.

All observations were made using standard network behavior generated by devices on the local network.

Key Takeaways

- Network devices announce themselves automatically through protocols like ARP and mDNS
- Passive listening reveals significant information without interacting with devices
- HTTP traffic is readable, while HTTPS traffic protects content through encryption
- Visibility on a network depends on trust and connection, not hacking
- Understanding normal traffic behavior is essential before studying advanced attacks