Work summary report
of
“Information extraction of devices behind NAT using WebRTC”

Shivasankaran V P
19110104

The entire project could be found here
The work done in this project could be broadly divided into 5 categories.

1. Setting up Mininet environment with X11 forwarding
2. Running GUI browsers in hosts within Mininet network
3. Writing Server code
4. Private IP extraction, User-agent information and Public IP query
5. MAC address extraction.

Setting up Mininet environment with X11 forwarding

Installing Mininet was not a problem but setting it up to be able to run GUI browsers is a very difficult task at least for the first time.

The recommended way of setting up the environment

1. Enable host adapter in Mininet settings to allow communication between Mininet VM and your host machine.
2. Run “ifconfig -a”, you should ideally see 3 interfaces if you only see two interfaces run the following common “sudo dhclient eth1”
3. Set up X11 tunnelling and XAuth [resource].

If X11 is set up correctly then you should be able to open an xterm by typing “xterm $node” and running “sudo wireshark &” should open Wireshark instance in $node.[1]

GUI browser instances in nodes within Mininet[source]

Install any browser in Mininet VM. I would recommend starting with google-chrome.

If you have set up X11 forwarding correctly this should be straightforward by just following the following steps

1. Open a terminal in your host system and SSH into your Mininet VM with X forwarding enabled which can be accomplished by using the -X flag in the SSH command.
2. Start a virtual network with “sudo mn --nat”
3. Run the following command within the initiated virtual network “h1 /usr/sbin/sshd”

[1] "Mininet Walkthrough." http://mininet.org/walkthrough/. Accessed 27 Nov. 2021.

4. Open another terminal in your host system and SSH into your Mininet VM.
5. From the terminal opened in step 4 again SSH into mininet@10.0.0.1 with X forwarding enabled. If you could not SSH into 10.0.0.1 then run the following command “sudo ifconfig s1 10.12.12.12” and try again[2].
6. Now you should be in 10.0.0.1, you can verify by “ifconfig”. From here you can open any GUI browser for example tying “google-chrome” should open Chrome within 10.0.0.1.

While trying to open the browser you may come across some problems most of them could be solved by running the browser with no sandbox or headless tags. In case you have WebGL errors while trying to open the virtual graphics adapter that is not compatible you will have to manually change it in the Mininet settings.

If Chrome preferences are not getting saved and you are getting a profile error refer to the solution posted [here]

Server

The server is written in python’s Flask library. The server has three important URL’s

/admin: The administrator page which will display the extracted clients data
/MyIP : The page client will visit to know their public IP address and it also gets the private IP.
/localIP : This will handle the POST requests from javascript to the server

A SQL database is used to store the client’s information. sqlite3 library of python is used to interact with the database.[3]

Information extraction

On the client-side, a javascript is run in the background when the user tries to view his public IP.

Pseudocode of javascript[4]:

```
Create a peerConnection with iceServers
Create an offer by binding it with its local machine description
While new candidate:
    Close peerConnection
    private_IP = private IP information from candidate
```

[2] "[openflow-discuss] Is it possible to open a browser inside Virtual box ...." 20 Apr. 2013, https://mailman.stanford.edu/pipermail/openflow-discuss/2013-April/004491.html. Accessed 27 Nov. 2021.

[3] "sqlite3 — DB-API 2.0 interface for SQLite databases — Python 3.10 ...." https://docs.python.org/3/library/sqlite3.html. Accessed 27 Nov. 2021.

[4] "Can You Get A Users Local LAN IP Address Via JavaScript?." https://stackoverflow.com/questions/20194722/can-you-get-a-users-local-lan-ip-address-via-javascript. Accessed 27 Nov. 2021.

```
        If private_IP ends with .local
            Ask for microphone permission if given
            private_IP = private IP information from candidate
        Send private_IP to server via POST to /localIP
        break
```

```
In the server-side
    Get user agent Information from the incoming connection request
    If received a POST request on /localIP
        Query IPinfo.io. With the received public IP[5]
        Then store it as the private IP and query result in the database
```

MAC address extraction

I tried extracting the MAC address of the client but all my efforts failed. Forcing me to come to the conclusion that getting MAC from a browser session is not possible and we will need other methods to get the MAC address.

The experimented methods are:

1. Javascript: By default, browsers have no permission to access the MAC address of a system. Unless the user is manually running the browser with privileges its is not possible to get MAC information.
2. Java-applet: Java applet in theory should be able to get the MAC address of the machine but will require manual authentication from the client (signed applet). So it was not used/tried as we want to collect information secretly.[6]
3. Address Resolution Protocol (ARP): It is a protocol that binds changing IP addresses to fixed MAC addresses. Since we know the private IP address of the client the following were tried.
    a) Direct query resulted in failure because the client was behind NAT
    b) Tried manually pinging the user’s system which resulted in a failure because the client was behind a NAT and destination NAT was not enabled in the NAT device.
    c) Even if we were able to ping the user’s system after that running the arp command will only give the MAC addresses of all links between our system and the first router. So we would not be able to view the MAC address of machines behind the NAT.

Active-X objects should be able to retrieve MAC address but Active-X is only supported by Internet explorer[7]. So it was not used in this project.

[5] "IPinfo.io: Comprehensive IP address data, IP geolocation API and ...." https://ipinfo.io/. Accessed 27 Nov. 2021.

[6] "Getting MAC address on a web page using a Java applet." 9 Jan. 2013, https://stackoverflow.com/questions/4467905/getting-mac-address-on-a-web-page-using-a-java-applet. Accessed 27 Nov. 2021.

[7] "What Browsers Support ActiveX? | Techwalla." https://www.techwalla.com/articles/what-browsers-support-activex. Accessed 27 Nov. 2021.

Learnings

During this project, I got to learn and build the following skills

1. Mininet architecture and network simulation
2. Came to know about a lot of protocols related to real-time communication over web browsers.
3. Understanding how browsers function and the security aspect which comes with it.
4. Various flags are used in browsers to provide security features and how permissions indirectly affect these flags functionality.
5. HTML, Javascript
6. Python's Flask library
7. SQL database handling
8. Why network security is important and educating people on the same is important.
9. How NAT's route packets from outside to inside and destination NAT.
10. How P2P connection is set up on the Internet.
11. Many network tools and commands like SSH, port scanning, arp, etc.

References

1. http://mininet.org/
2. https://askubuntu.com/questions/32706/profile-error-when-launching-google-chrome
3. https://mailman.stanford.edu/pipermail/openflow-discuss/2013-April/004491.html
4. https://ipinfo.io/developers
5. https://developer.mozilla.org/en-US/docs/Glossary/ICE
6. https://docs.oracle.com/javase/7/docs/technotes/guides/jweb/security/rsa_signing.html
7. https://flask.palletsprojects.com/en/2.0.x/
8. https://docs.paloaltonetworks.com/pan-os/10-0/pan-os-admin/networking/nat/source-nat-and-destination-nat/destination-nat-dns-rewrite-use-cases/dest-nat-dns-rewrite-reverse-use.html