Yoshiaki Ryuzaki
Sukh Atwal

# Assignment 3 - DNS Spoofing

## Contents

Yoshiaki Ryuzaki
Sukh Atwal

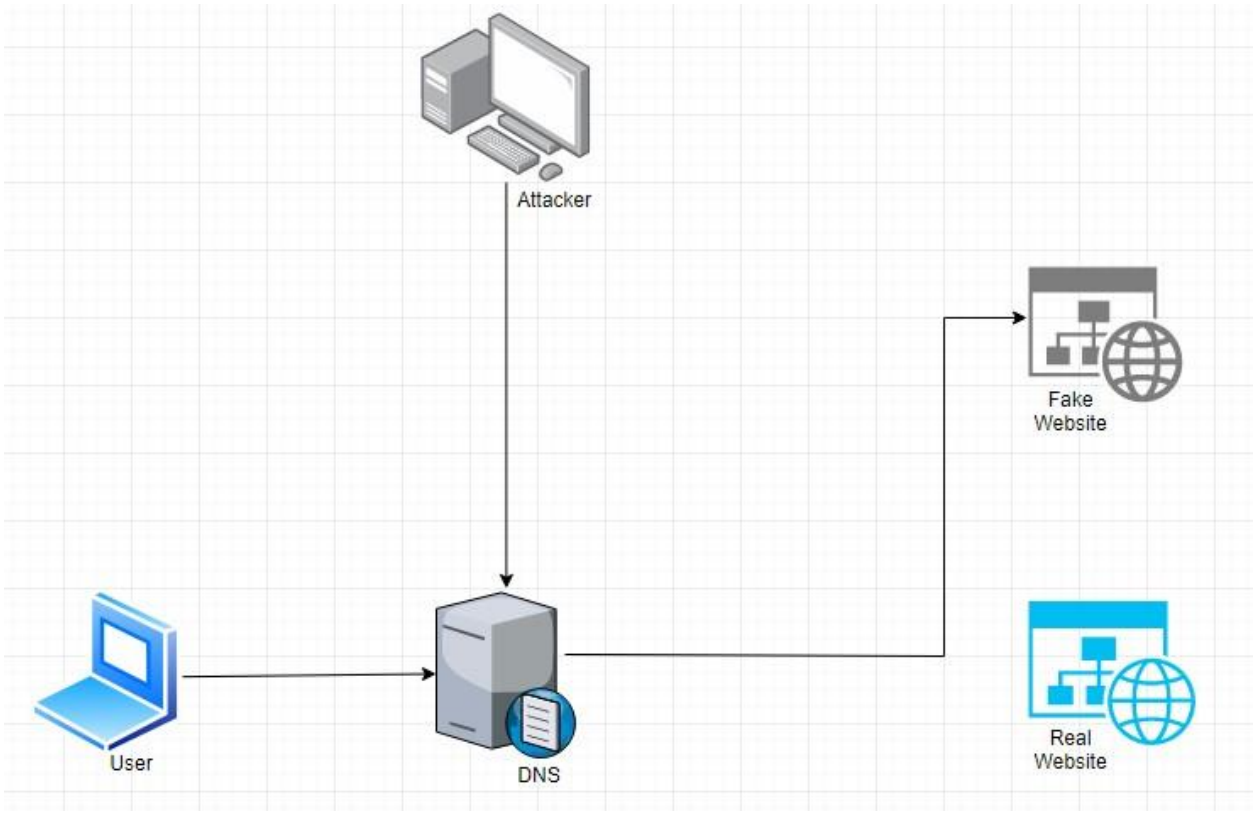
# Introduction

The objective is to learn how DNS spoofing works directly by writing the code. We created a simple DNS and ARP spoof application in Python2 with the goal to redirect the victims web traffic to an IP address that we've specified. All code as well as a how-to quick reference guide are located in the same zip file.

The application starts with obtaining the IP address of the victim, your router and the spoofed website. Due to the nature of the victim PC often receiving a legitimate DNS response before our spoofing application can, we have to add iptables rules. These rules would make it so that any legitimate DNS responses that are being forwarded would be blocked. It would then move forward to crafting ARP packets, sending out spoofed ARP packets and sniffing DNS requests where we then send back our spoofed response.

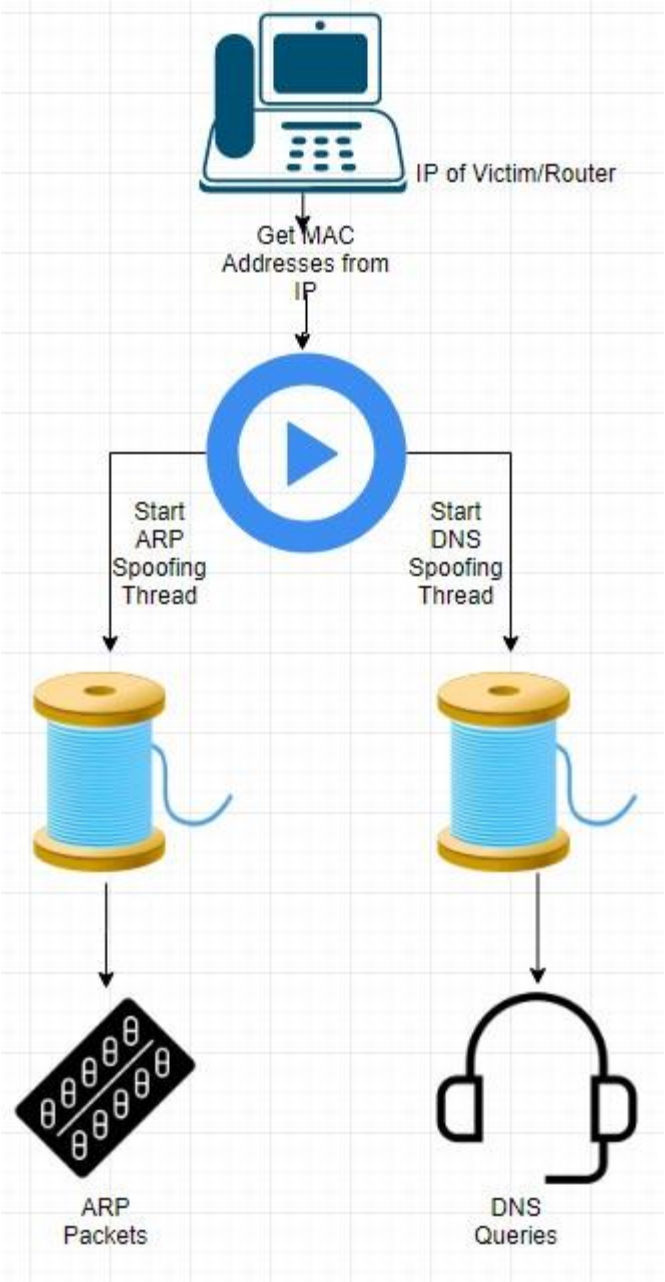
Yoshiaki Ryuzaki
Sukh Atwal

## Design




This is a general overview of how spoofing happens. Here, the user sends a request to a website they want to visit. However, the Attack injects a fake DNS entry. This causes the User to reach the Fake Website instead of the Real Website that they intended on visiting.

Yoshiaki Ryuzaki
Sukh Atwal




This is a brief overview of how the Python application will work. You start by inputting the Victim and Router IP in addition to changing the web server IP if needed in the main.py file. It'll search for the MAC Addresses using those IP addresses. Afterwards two threads will start, one for ARP and the other for DNS. The ARP would be responsible for sending packets to the victim and router. Meanwhile, the DNS would listen for queries and send spoofed responses back. During all this, the attacking machine will sniff all the traffic on port 53.

Yoshiaki Ryuzaki
Sukh Atwal

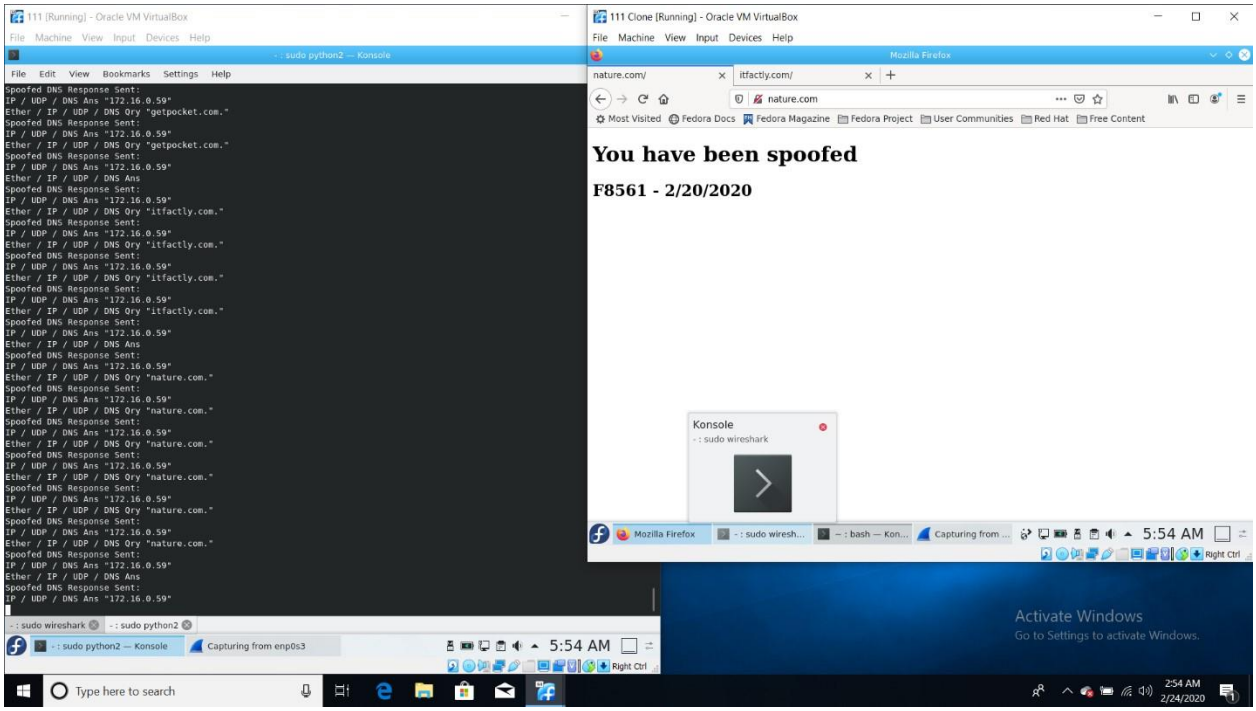
## Testing

The victim IP is 172.16.0.60, our router IP is 172.16.0.1 and the website is hosted on our IP 172.16.0.59. This was all done on Fedora KDE Plasma using two Oracle Virtual Machines.

| # | Description | Tools Used | Expectations | Actuality | Pass/Fail |
|---|---|---|---|---|---|
| 1 | Check if Victim is redirected to our locally hosted website | Chrome, Python, Wireshark | The Victim is redirected to our spoofed website when trying to access any website | The Victim is redirected to our spoofed website | Pass |
| 2 | Check if spoofing works for arbitrary domains | Wireshark | Spoofed DNS response is sent to any random string that the Victim requests | If the website doesn't exist, it'll be spoofed to our locally hosted website | Pass |

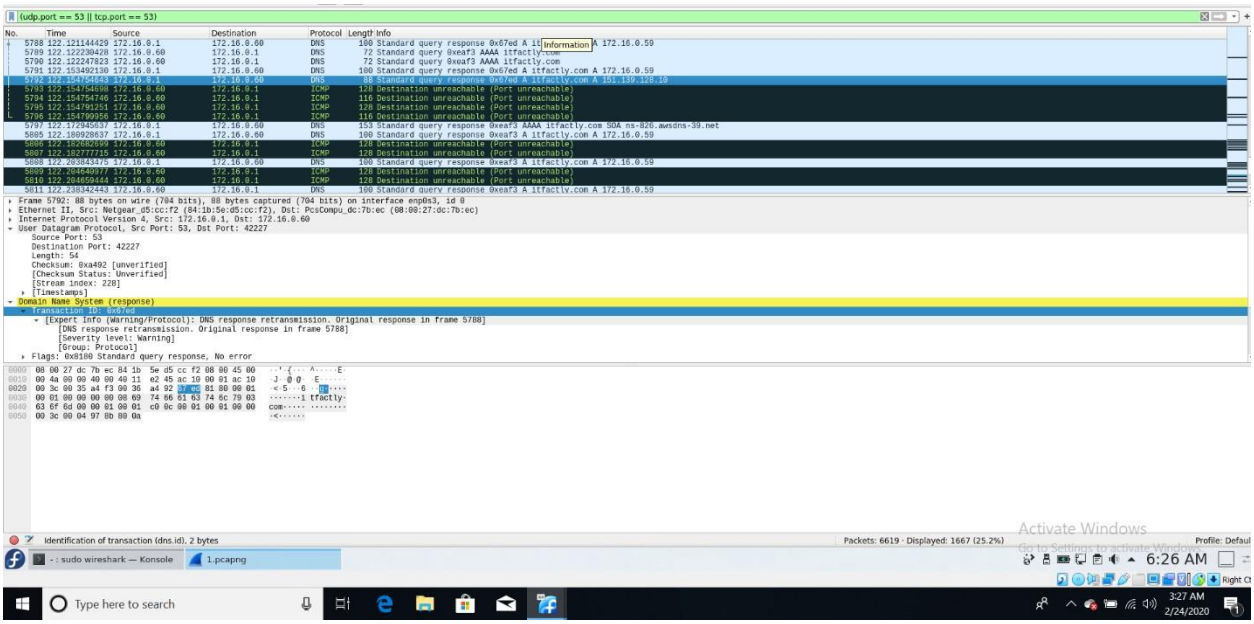
Yoshiaki Ryuzaki
Sukh Atwal

## Test 1



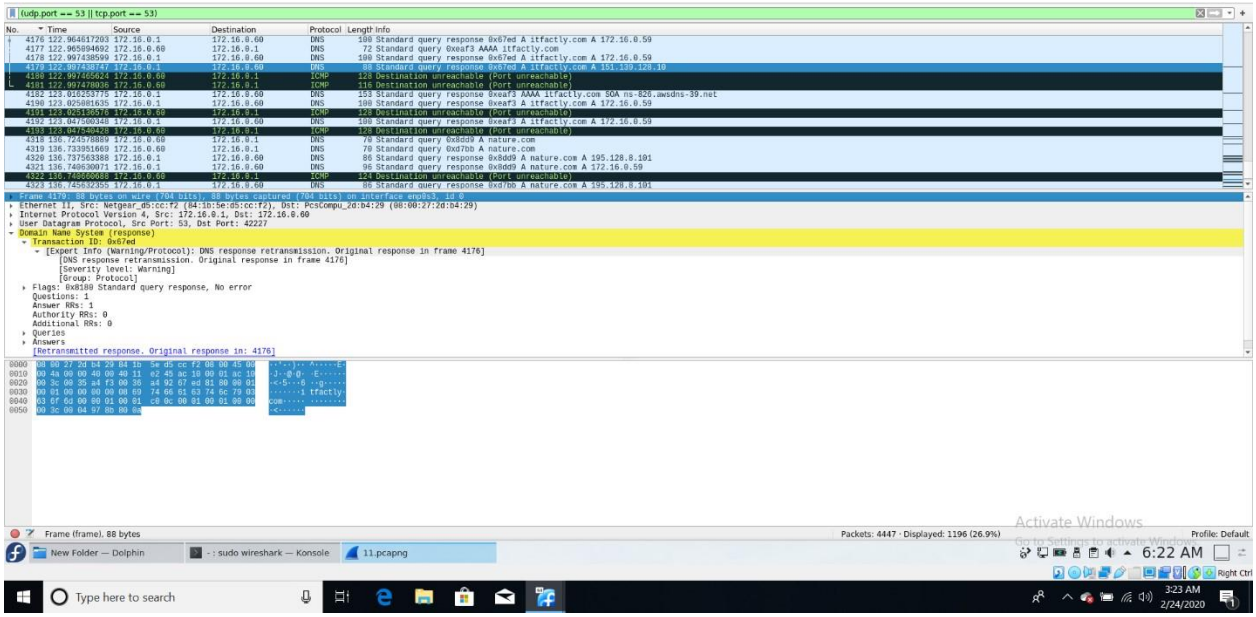

Both the websites itfactly.com and nature.com both returned a spoofed DNS response. You can also see the terminal showcasing the website along with the IP it was going to be spoofed to 172.16.0.59. I've also attached two Wireshark screenshots from the Victim and Attacker's side.

Attacker:



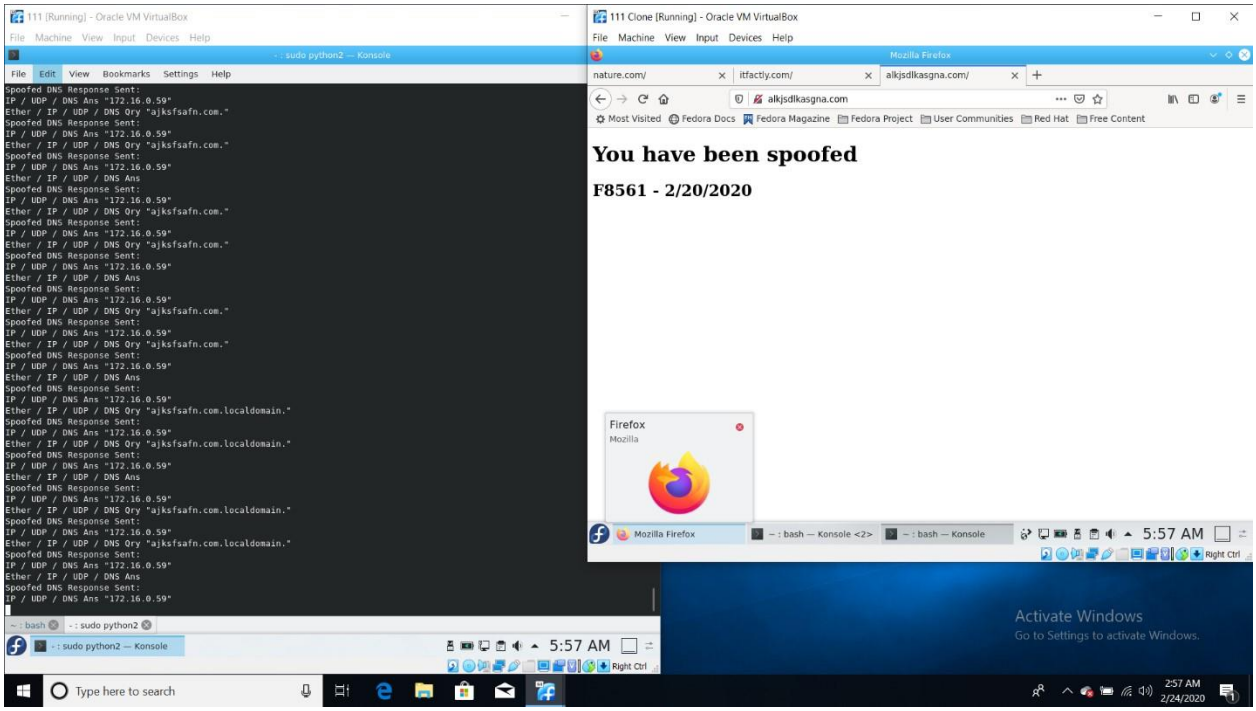
Victim:

Yoshiaki Ryuzaki
Sukh Atwal



Yoshiaki Ryuzaki
Sukh Atwal

## Test 2




This showcases it's able to handle any arbitrary(random) domain strings.