

Computer Security HW1 Report

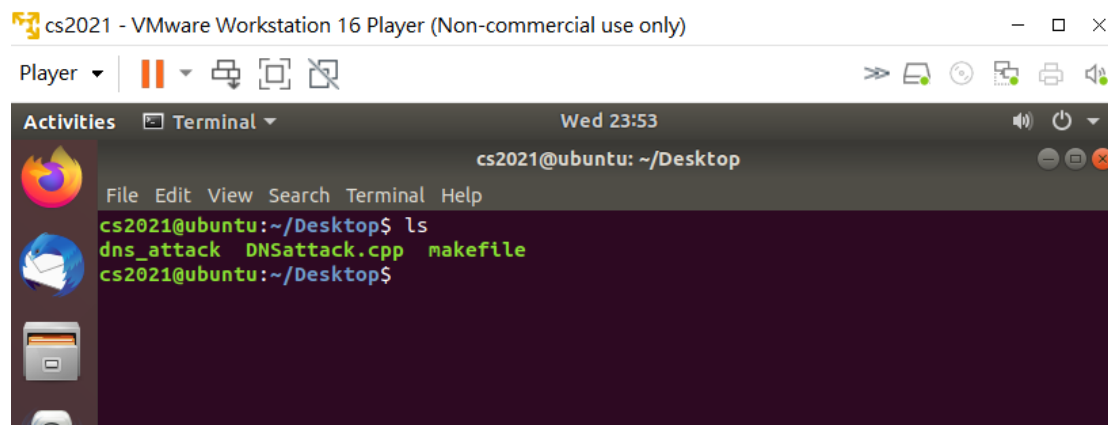
0716234 蕭彧

Task1 & Task2 demonstration:

Here is the main function of my code. Obviously, I'll send three packets of different website queries by calling the function, DNSattack.

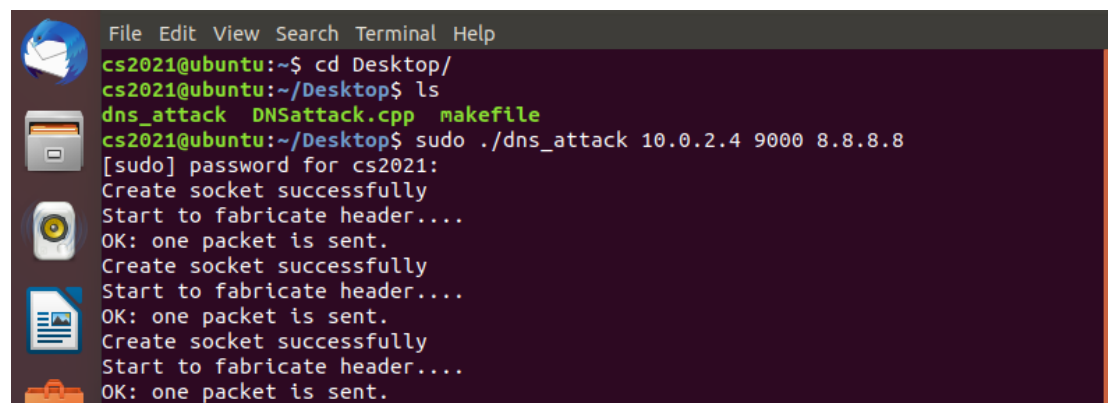
```
int main(int argc, char *argv[]){
    DNSattack(argv[1], argv[2], argv[3], (unsigned char *)("nctu.edu.tw"));
    DNSattack(argv[1], argv[2], argv[3], (unsigned char *)("google.com"));
    DNSattack(argv[1], argv[2], argv[3], (unsigned char *)("ieee.org"));
    return 0;
}
```

I've already type make to compile my .cpp file to an executable file. Next, I'll directly execute the dns_attack file.



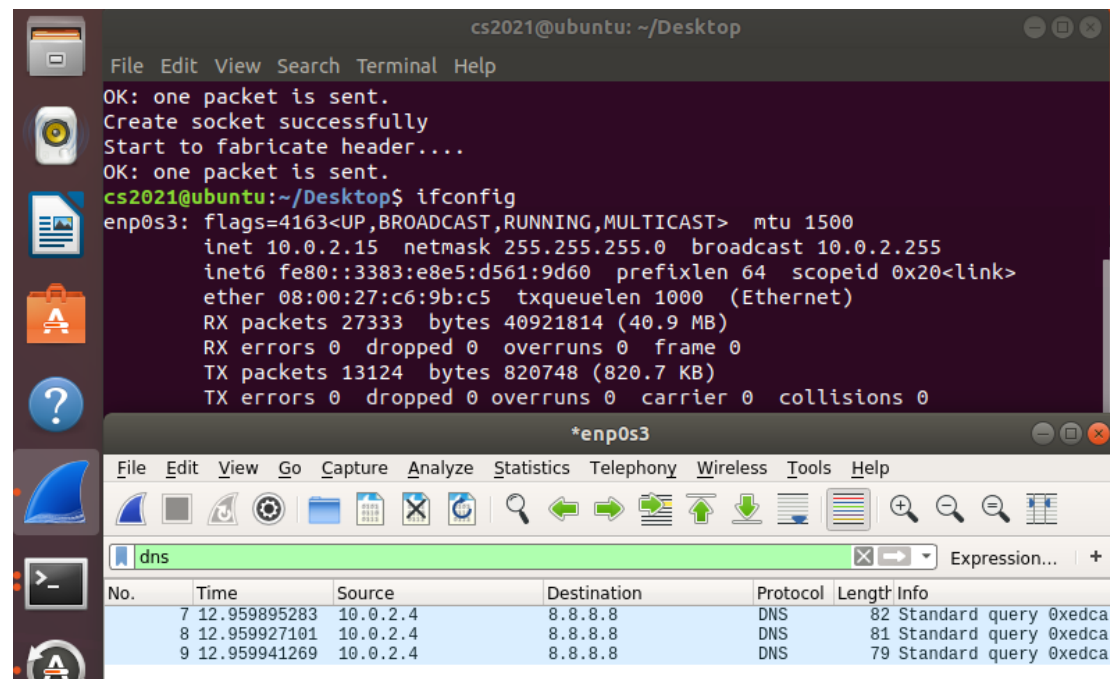
```
cs2021@ubuntu: ~/Desktop
ls
dns_attack  DNSattack.cpp  makefile
cs2021@ubuntu:~/Desktop$ make
```

The victim IP is another vm machine, and I send the spoofed packet to google server.

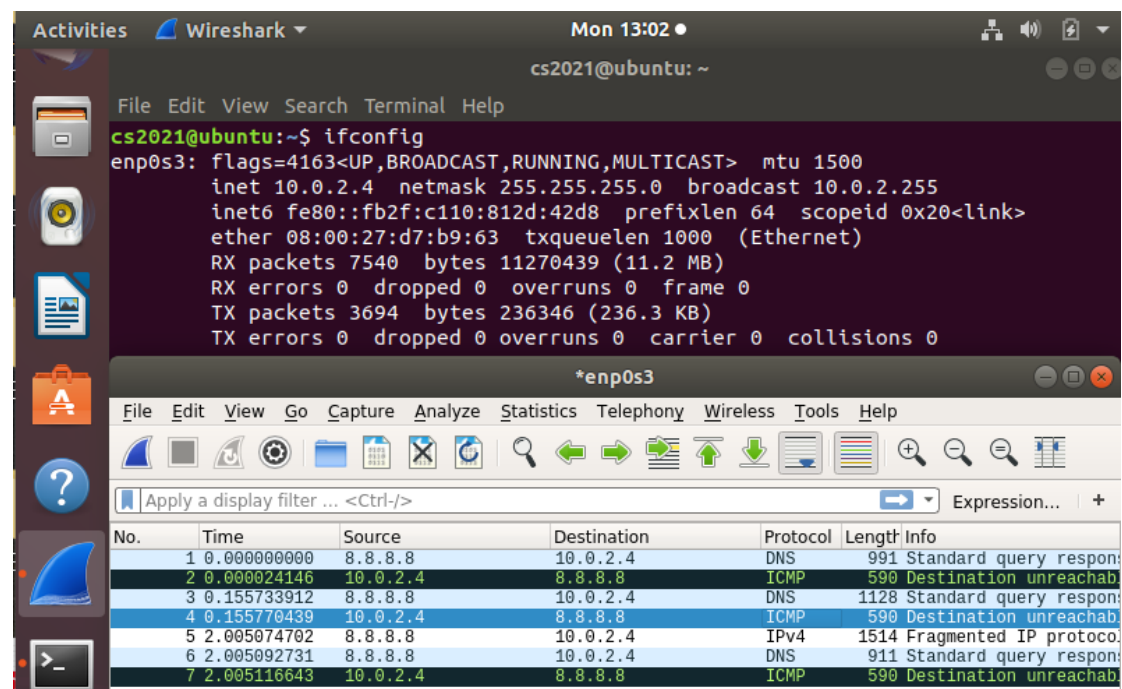


```
cs2021@ubuntu:~$ cd Desktop/
cs2021@ubuntu:~/Desktop$ ls
dns_attack  DNSattack.cpp  makefile
cs2021@ubuntu:~/Desktop$ sudo ./dns_attack 10.0.2.4 9000 8.8.8.8
[sudo] password for cs2021:
Create socket successfully
Start to fabricate header....
OK: one packet is sent.
Create socket successfully
Start to fabricate header....
OK: one packet is sent.
Create socket successfully
Start to fabricate header....
OK: one packet is sent.
```

Wireshark shows that there are three packets sent out. Next, we'll check out the victim Wireshark.



Victim's Wireshark receives three DNS packets. All of them have an amplification ratio more than 10.



Amplification technique:

First thing I do is to break the limitation of size of a DNS packet. I use EDNS protocol to let the response packet size can up to 4096 bytes. Then I set the type of

DNS query ANY, which can list all of the information about this website such as other related server or TLS information and so on. By setting to ANY, we can effectively amplify the response packet size with a small size of request packet. Therefore, the amplification rate of my packet can higher than 10.

Defend against Dos attack:

We can use a regular expression filter to filter those malicious packets. Applying traffic signature filters can be an effective defense against reflected amplification attacks. These attacks have identifiable repetitive structures from which a regular expression can be derived. We can just remove those packets. But there are some drawbacks, inspecting every single packet may lower the performance, which can even worsen the situation. Therefore, this method relies on a fast and stable filtering technique.