NASAHW6

B07902117 陳漱宇

Network Administration

- **Short Answers**

  1. We will set several IPs to the same domain name because we want to achieve load balancing. We can use serveral servers to provide DNS service, which mostly be used in some domain names that are highly inquired. In this way, we can balance the network traffic to those servers to release the loading of the single server. Additionally, load balancing increases fault tolerance of DNS service, because when there is one server broken, the other servers are still operate normally so the circumstance may not be so bad at all.

     Reference:

     http://phorum.study-area.org/index.php?topic=64894.0

     https://zh.wikipedia.org/wiki/%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1

  2. If we want to get the IP of the domain name, if the record of the mapping does not exist in the cache in our operating system and web browser, we will send a query to the DNS server. And DNS will lookup the IP of the domain name and send the response to us, and then add the record to its cache so that it can respond to us faster next time. And for recursive server and authoritative server, when we ask DNS server, it will first go to recursive DNS server, recursive DNS server will check if there is a cached record from the authoritative DNS server. If not, then it will start going through the authoritative DNS hierarchy and then respond the result to the client and store the record to the recursive DNS server.

     Reference:

     https://umbrella.cisco.com/blog/2014/07/16/difference-authoritative-recursive-dns-nameservers/

     https://www.arthurtoday.com/2011/06/recursive-dns-server.html

  3. When making a DNS change, it takes time for the changes to take effect, and that is called DNS propagation. It is the time it takes for the domain DNS to refresh the cache on the network. DNS will refresh according to the TTL (Time To Live). When the DNS refreshes according to its TTL, the propagation is complete and your site will load. And we will prefer using lower TTL when we make any record changes, bacause any change we make will not propagate until the TTL expires. In contrast, when we have nothing to do with the record, we will prefer using longer TTL, because the site will be queried too many times when using lower TTL and we will need to pay additional cost for excessive queries. Also, longer TTL's can cut resolution times. Every time a query has to ask an authoritative name server, it adds an additional lookup, which could add precious milliseconds.

     Reference:

     https://www.siteground.com/kb/what_is_dns_propagation_and_why_it_takes_so_long/

     https://www.inmotionhosting.com/support/domain-names/dns-nameserver-changes/domain-names-dns-changes

     http://social.dnsmadeeasy.com/blog/long-short-ttls/

4. If the processes of DNS query and response are not encrypted, the attackers may counterfeit the DNS packet, responding the uncorrect IP instead of the correct IP, and then make the site of the uncorrect IP almost same to the correct one, in this case, the client may keep some information in it unwittingly and then be at risk. But using DNSSEC can avoid it. DNSSEC can provide three secure service, data integrity, origin authentication of DNS data, and authenticated denial of existence. First, for data integrity, it can generate RRSIG (Resource Record Signature) by using resource record and digital signature, and the receiver can confirm whether the data is correct or not by using RRSIG and DNSKEY. Second, for origin authentication of DNS data, we need to sure that the DNS server of the domain is authentic. DNS server will keep the digital signature of its DNSKEY in its parent zone server, adn that is called DS (Delegation Signer), so the DS in parent zone can vertify the DNSKEY of child zone is right. So after recursion, we only need to sure that the root zone is correct so that we can trust the other zones. And the root zone is now be maintained by the professional staff. Last, for authenticated denial of existence, when a domain name does not exist, we need to sure that it indeed does not exist rather than attackers counterfeit the response which makes us misunderstand the truth. Using DNSSEC, it will add NSEC record in each alphabetically DNS record. And we can check out the NSEC record and compare with the others to check if it really does not exist.

   Reference:

   http://www.cc.ntu.edu.tw/chinese/epaper/0022/20120920_2206.html

- **DnsDoOmSday**

1. When I looked into the file called named.conf.options, I found that there is something which may be dangerous potentially. That's in below.

   ```
   recursion yes;
   allow-recursion { any; };
   ```

   And for why it is dangerous I will explain in problem 2.

   Reference:

   https://ssorc.tw/420

2. Continue to problem 1, in this case, your server will become an open DNS server, that is, you allow anyone (allow-recursion { any; };) to query other domains (recursion yes;), and it may cause DNS amplification attack, that is, attackers might spoof thier IPs and use your server as medium and implement DDoS attack, so when your server will send a large number of response packets back to those true IPs which be counterfeited before by attackers and make the network traffic be full.

   Reference:

   http://www.cc.ntu.edu.tw/chinese/epaper/0028/20140320_2808.html

3. Seeing into the query log file. I first check out all the clients (the fourth column) by the following command.

   ```
   cat query.log | awk '{print $4}' | sed 's/#.*//g' | sort | uniq -c
   ```

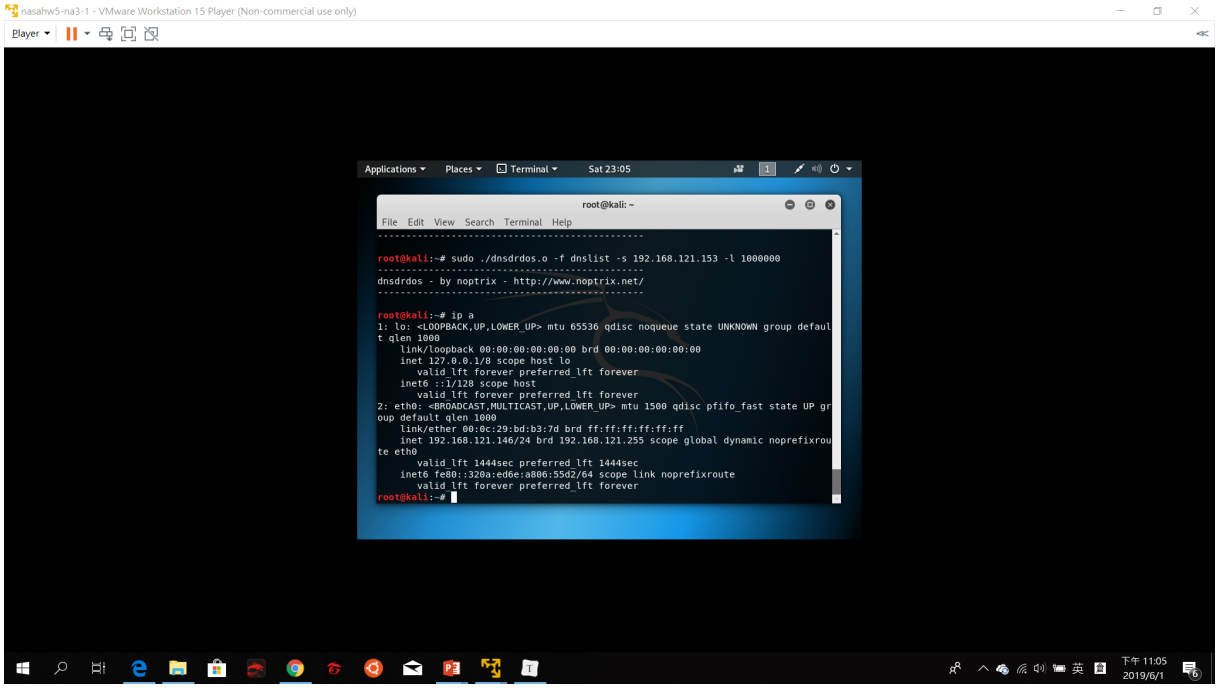
   And then the result is in the below. The first column is the times they appear, and the second column is their IP (victim).

```
10988 140.112.30.32
11101 140.112.30.33
11087 140.112.30.34
11151 140.112.30.35
11143 140.112.30.36
11169 168.95.98.250
11278 168.95.98.252
22083 168.95.98.254
```

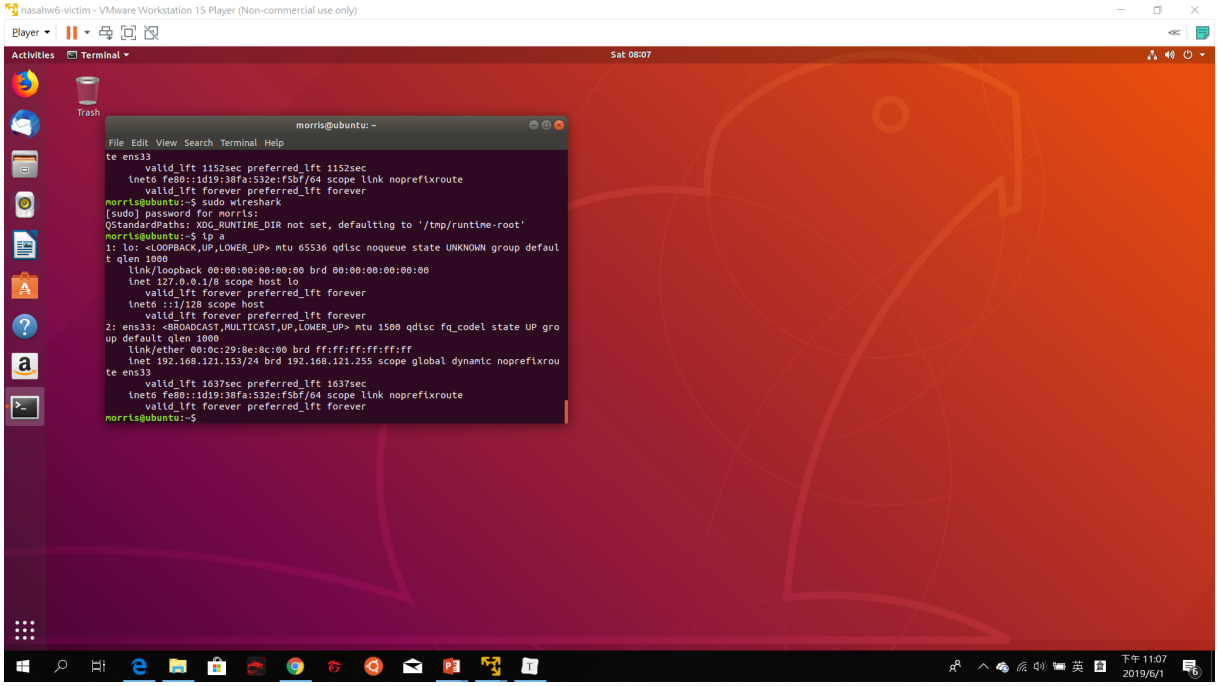
And I find out the IP of DNS server is 168.95.98.200. The attack conduct IP spoofing to counterfeit the source IP to the IPs of the aboved result, and send a lot of query to the DNS server (DDoS), and then DNS will conduct recursive query and send response packet back to original IPs so it will lead to the explosion of network traffic and waste their resources.

4. To prevent it, we can configure the named.conf.options, and set recursion option to "no" or modify allow-recursion option, just allow the people we trust instead of allow anyone.
5. I build three VMs, one kali linux for attacker, two ubuntus for victim and DNS server. And then I first check up their IPs.

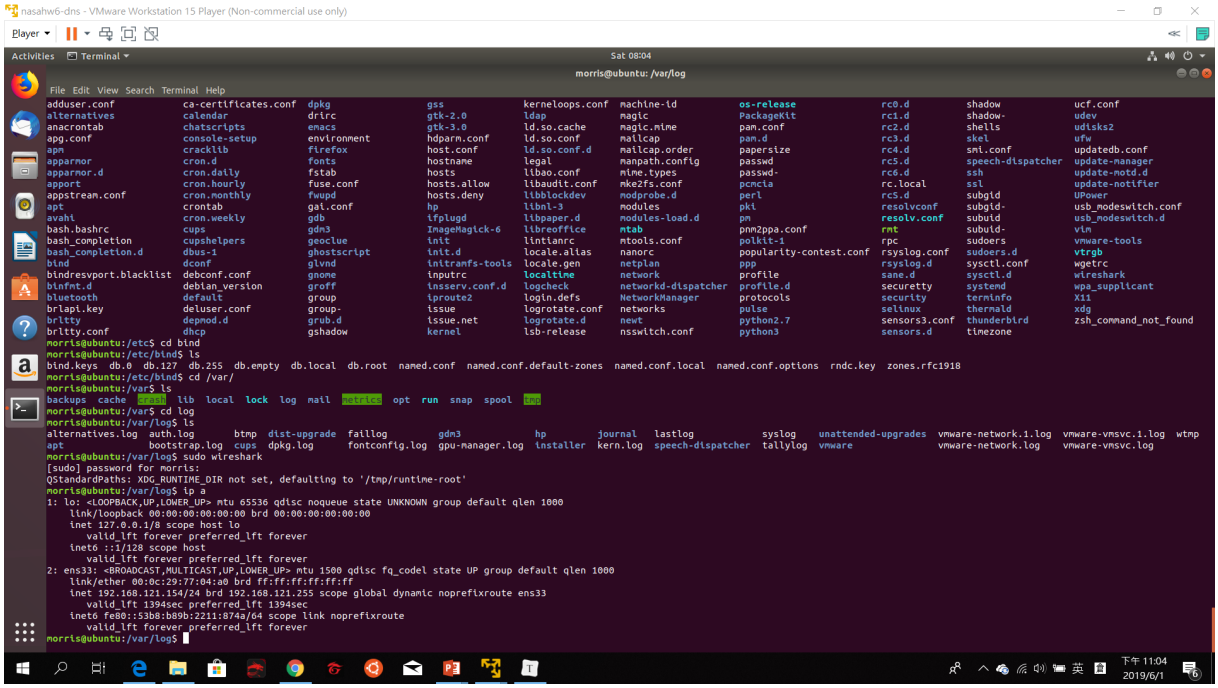
   Attacker (192.168.121.146)



   Victim (192.168.121.153)



DNS server (192.168.121.154)



In DNS server, I do the following operations.

```
sudo apt-get install bind9
sudo vi /etc/bind/named.conf.options
```

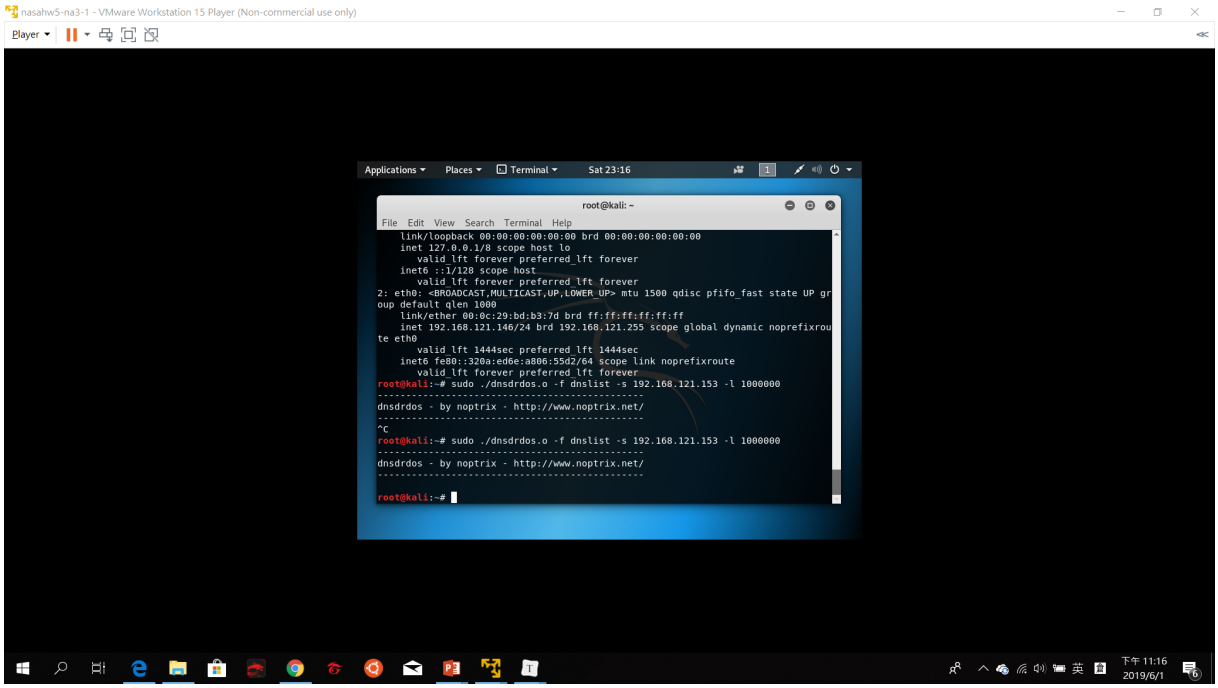
Then modify named.conf.options to become as same as the provided option file.

For attacker, I use the code from https://raw.githubusercontent.com/rodarima/lsi/master/p2/dnsdrdos.c and compile it.

```
gcc dnsdrdos.c -o dnsdrdos.o -Wall -ansi
```

According to the code's option, I create a file named dnslist, containing the IP of my DNS server (open DNS server), and then I conduct the following command.

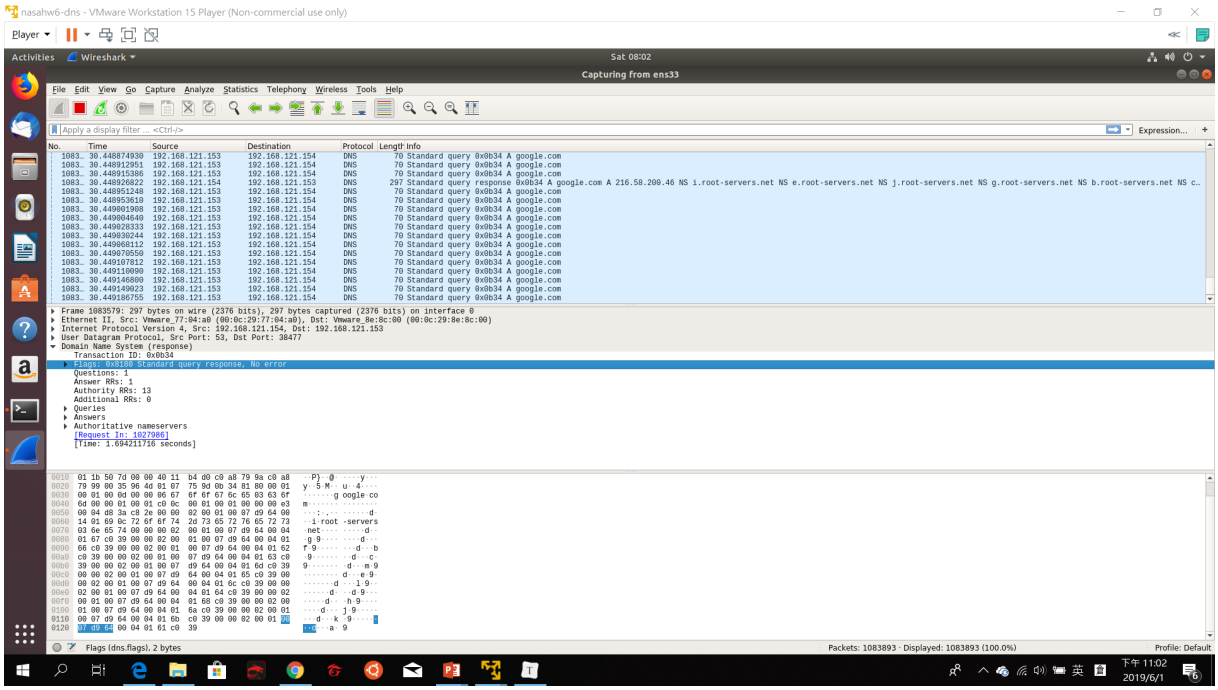
```
sudo ./dnsdrdos.o -f dnslist -s 192.168.121.153 -l 1000000
```



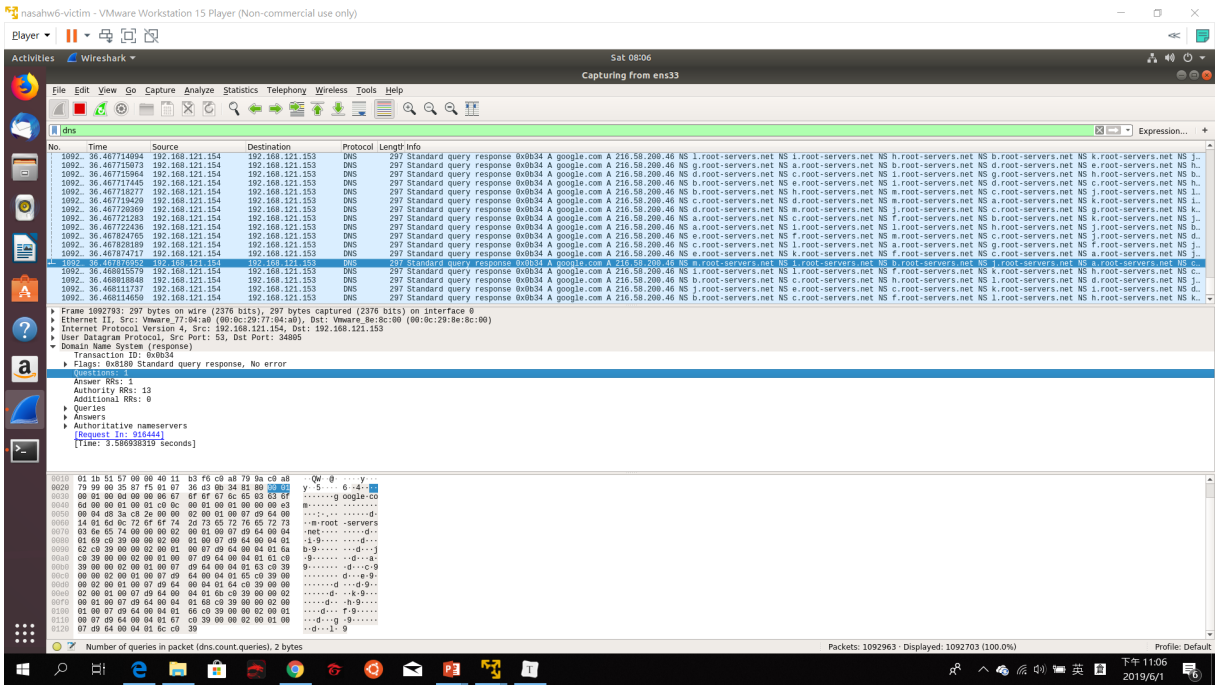
Argument -f gives the file which contains the IP of DNS server, argument -s gives the IP (victim IP) which the attacker want to spoof it, and argument -l gives the number of times sending queries to DNS server.

Then I see into the wireshark in DNS server, I see it catch DNS query and response packets. And I see into the wireshark in victim, I also see DNS query and response packets, because DNS server will send response packet to the victim (true IP).

Wireshark in DNS



Wireshark in victim



Reference:

https://www.youtube.com/watch?v=1mw7GE1hQa4&fbclid=IwAR227855H7FJAYdk_t9zO8fLWDtEaaMCzFKoWA00EBh7MR7ij1oZKasAwCg

https://raw.githubusercontent.com/rodarima/lsi/master/p2/dnsdrdos.c