Colin Quinn

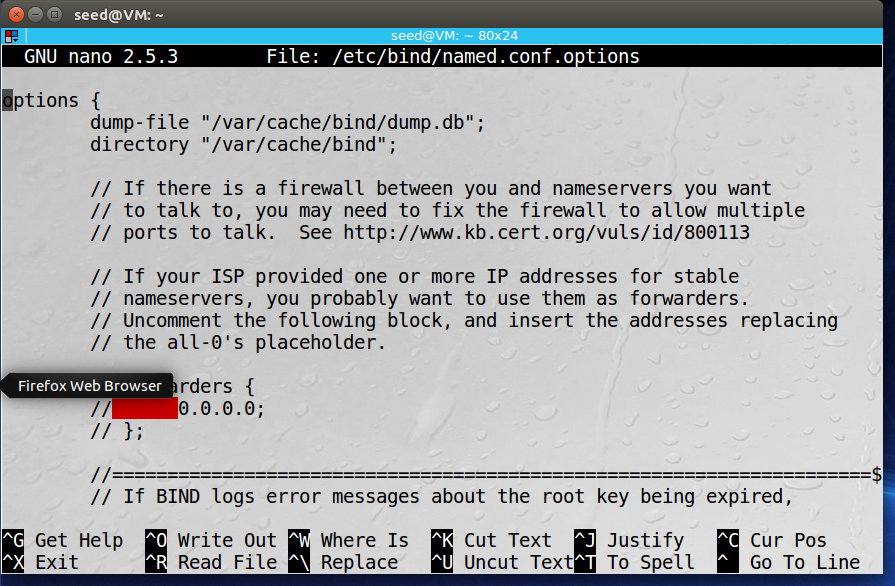
Wireless Security Lab 1

Lab 1: DNS Cache Poisoning

Graphical user interface, text

Description automatically generated

As task 1, the user machine must be set up so that it can call nameserver as the IP of our local DNS server. This is done by adding the line `nameserver <IP\_addr>` to our /etc/resolv.conf file.



Next is to add a dump file to our server. This lets us store the previous information of the server in a file that we can read. It has been done as the first line in options, as well as disabling dnssec below the screenshot.

Graphical user interface, table

Description automatically generated

Here is the wireshark capture of pinging a website through our DNS server. We can see that the first packet comes from our user (10.0.2.7) and goes to our server (10.0.2.9) before the server routes it to its destination, [www.google.com](http://www.google.com) (192.58.128.30).

Text, table

Description automatically generated

The users results from creating zones within the DNS server. We can see that there is now a barrier between the authorized and additional section. Any requests from users through the DNS server will now be additionally linked within the server and protected from outside IP addresses.

Text

Description automatically generated

After changing the host file of for the user to route any packets that are intended for [www.bank32.com](http://www.bank32.com) towards youtube.com (216.239.32.10). Youtube.com IP was found using `dig youtube.com`.

Graphical user interface, text

Description automatically generated

Once the user digs a DNS server that has someone sniffing it, this is what they will see as it looks like nothing is really out of the ordinary aside from the IP addresses that it goes through.

A picture containing text, receipt

Description automatically generated

And here is the attacker’s view of the request that the user sent through the server.

A picture containing graphical user interface

Description automatically generated

Here is the User’s packet data when running dig on the poisoned DNS server. We can see that they are unable to connect through the server as the ICMP packets are unable to connect through their intended ports. It should also be noted that when dumping the DNS server it returns out a bad cache. This is as a result of the failed GET request and the cache poisoning.

A piece of paper with writing

Description automatically generated with medium confidence

And this is what the attacker sees when the user tries to connect through the poisoned DNS server. Note that the filter is applied so that only packets from 10.0.2.7 and 192.168.0.101 are displayed.

Graphical user interface, text, application

Description automatically generated

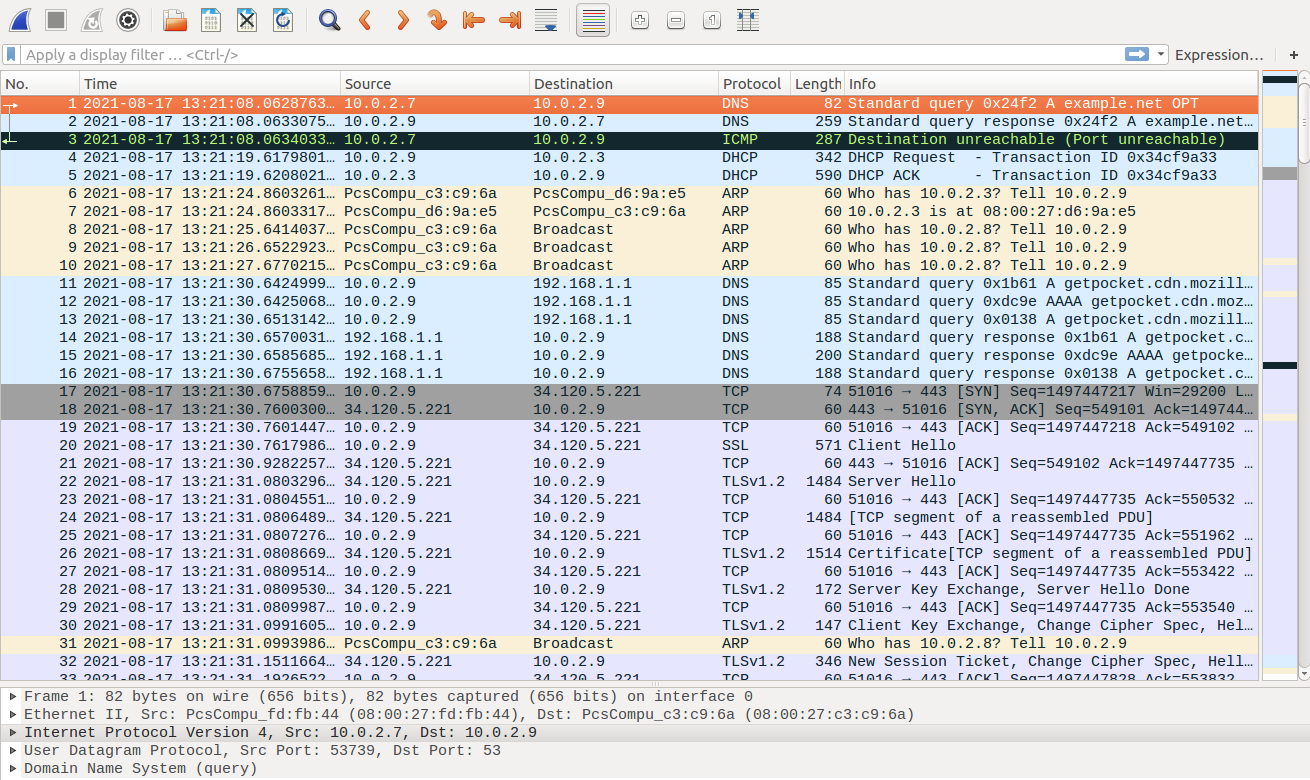
Here is the scapy script used for task 7 that targets the authority section of the DNS server. This will alter the request from the user’s intended target to our [www.attacker32.com](http://www.attacker32.com) address, where we do not get a response, but we can see that it gets rerouted to that address below: A picture containing calendar

Description automatically generated

In order to target multiple IP addresses, we can simply add new lines in our scapy tool. Such as the one added to target google.com here:

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

We should also update our DNSpkt field to represent multiple nameservers that we are targeting.



Here is the user’s wireshark data when sending a request with a targeted additional section. We can see that this attack did not seem successful as the initial request does not go through the server. A picture containing text, receipt

Description automatically generatedAs also shown here, the user simply receives information that they have requested example.net that has gone through the additional servers at a.iana-servers.NET. This shows the failed poisoning because we should see that rather than routing through these servers, it should go through one of the servers added into our additional section in our scapy script.