Local DNS Attack Lab Report

Mudit Vats
mpvats@syr.edu
2/16/2019

Table of Contents

Overview

This lab report presents observations and explanations for the tasks described in the Local DNS Attack Lab.

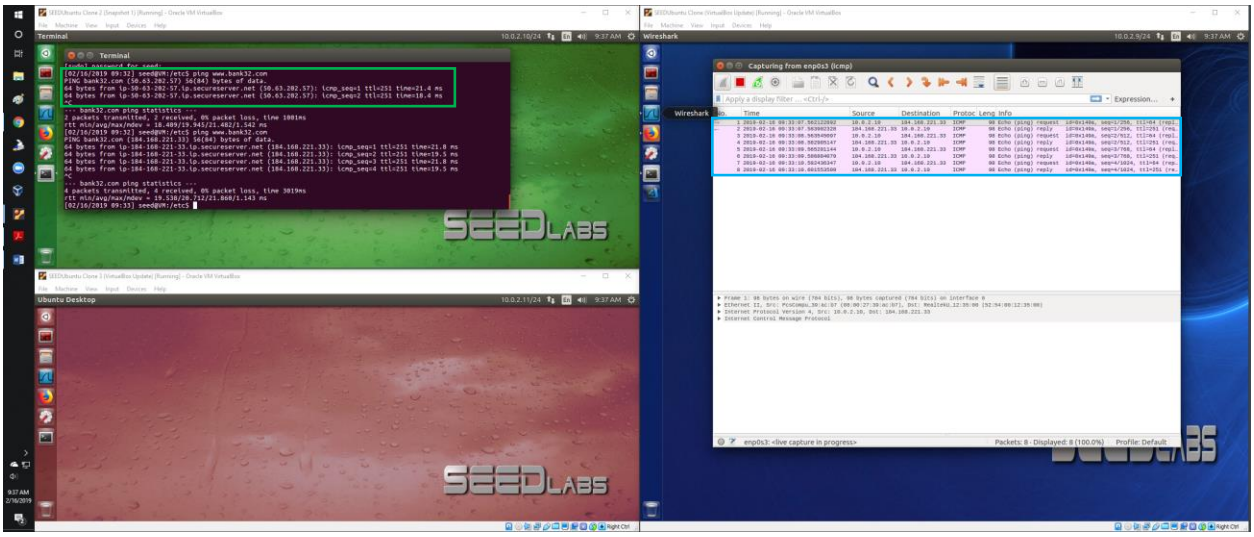
In the screen shot below, you can see three VMs.

VM Background	Role	IP Address
Blue	Local DNS Server	10.0.2.9
Green	User Machine	10.0.2.10
Red	Attacker	10.0.2.11

Task 4: Modifying the Host File

Modify the /etc/hosts, file for www.bank32.com. Observe results.

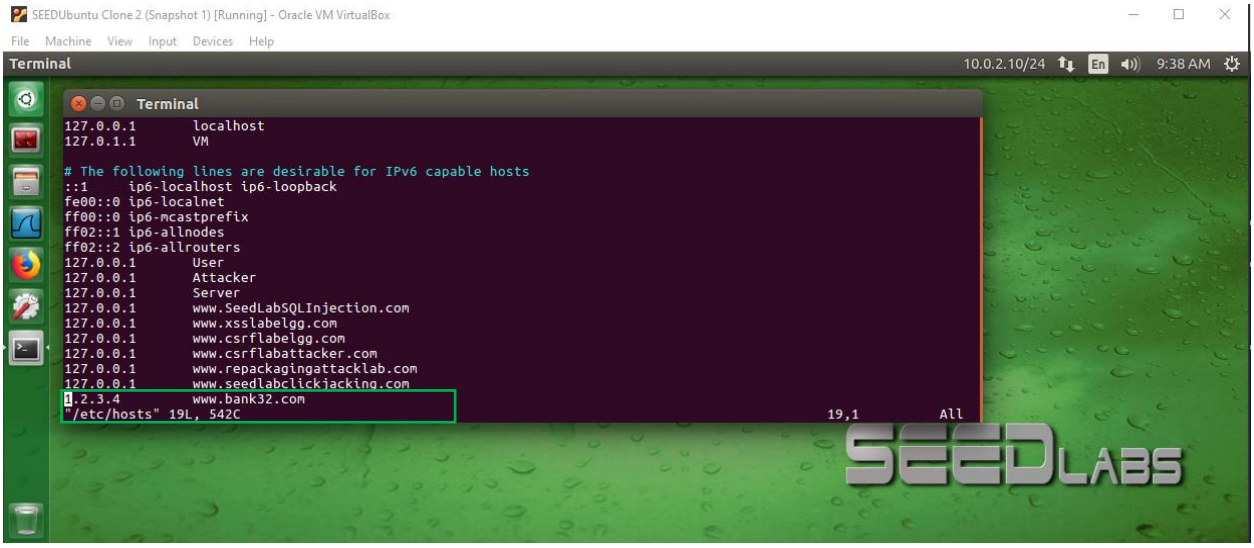
In the figure below, we see the ping of www.bank32.com. This is shown in the Green VM. We can also see the WireShark output in the Blue VM which shows the ICMP ping requests from Green VM to the IP address of www.bank32.com.



In the figure below, we update the /etc/hosts file in the Green VM. We add the following entry:

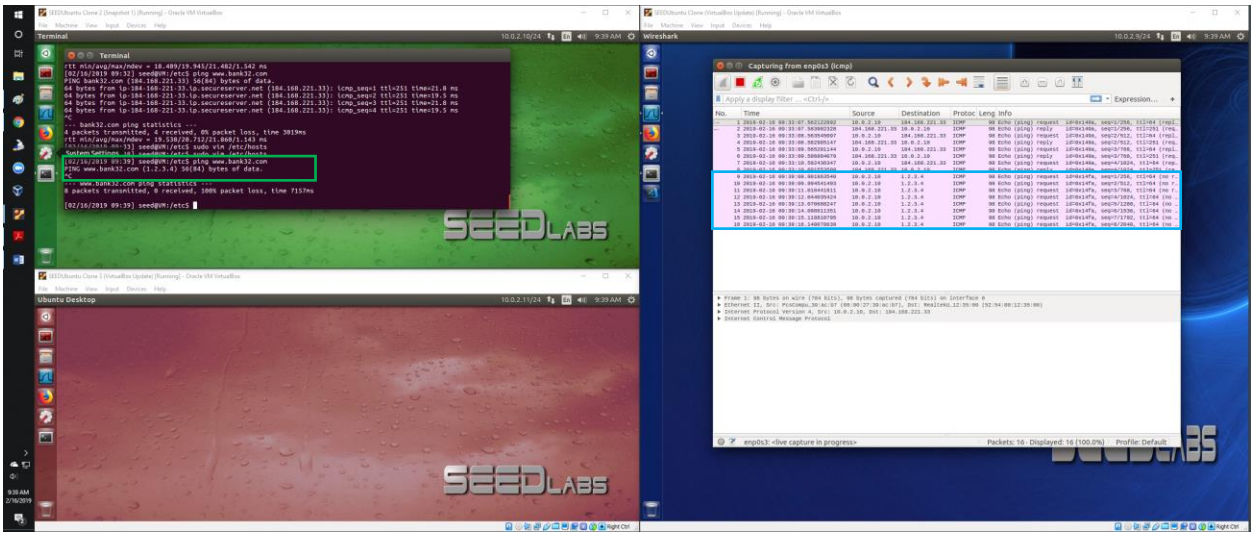
- "1.2.3.4 www.bank32.com"

By adding this entry in the Green VM's hosts file, we now have a host to IP mapping for www.bank32.com. This entry will be used first as part of the host to IP resolution.



In the figure below, we see a ping on Green VM to www.bank32.com. On the right, we see the Blue VM which shows the WireShark capture.

The difference this time is that when we try to ping www.bank32.com, it resolves to the IP address we placed into the /etc/hosts file which is 1.2.3.4. This can be seen on the Green VM and the Blue VM WireShark capture.



Observations / Explanations

In this task we added an IP and host entry for www.bank32.com in Green VM's /etc/hosts file. This is the file which is checked first for hostname-to-address resolution before going out to the Local DNS.

Once we make the modification and attempt to ping www.bank32.com, it pings the IP address we placed in the /etc/hosts file that is associated with www.bank32.com; i.e. 1.2.3.4. We can see the 1.2.3.4 IP address via the ping command and the WireShark ouput.

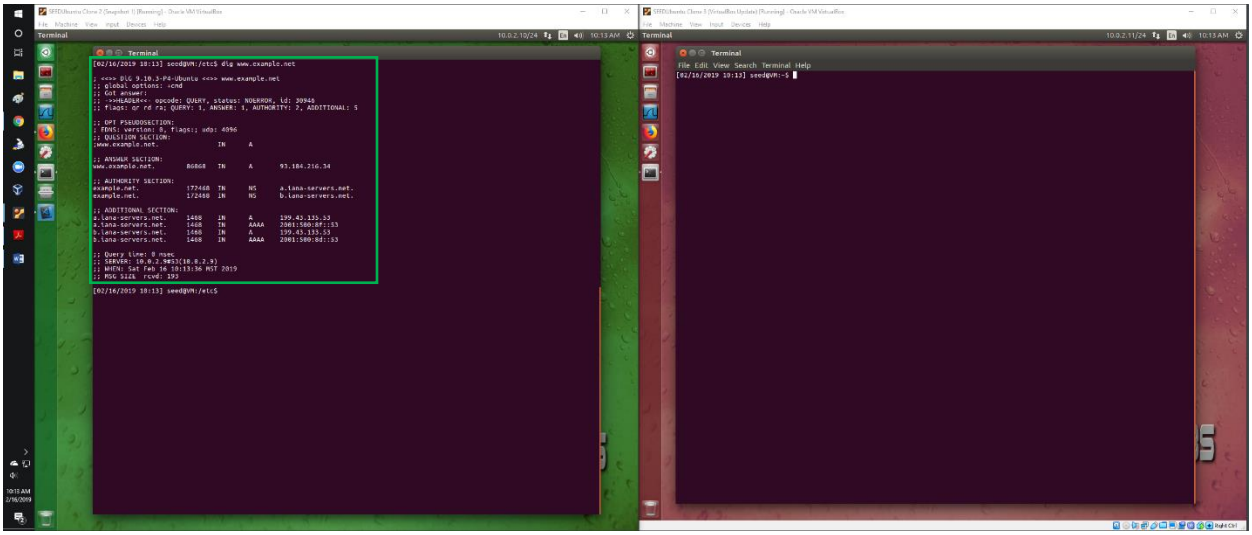
We can see in this task that the /etc/hosts is indeed checked and since the IP to hostname mapping is there, the user system does use that entry to resolve the hostname instead of doing a Local DNS query.

Please note, after this task, the entry in /etc/hosts was removed.

Task 5: Directly Spoofing Response to User

Spoof a DNS response to the user machine.

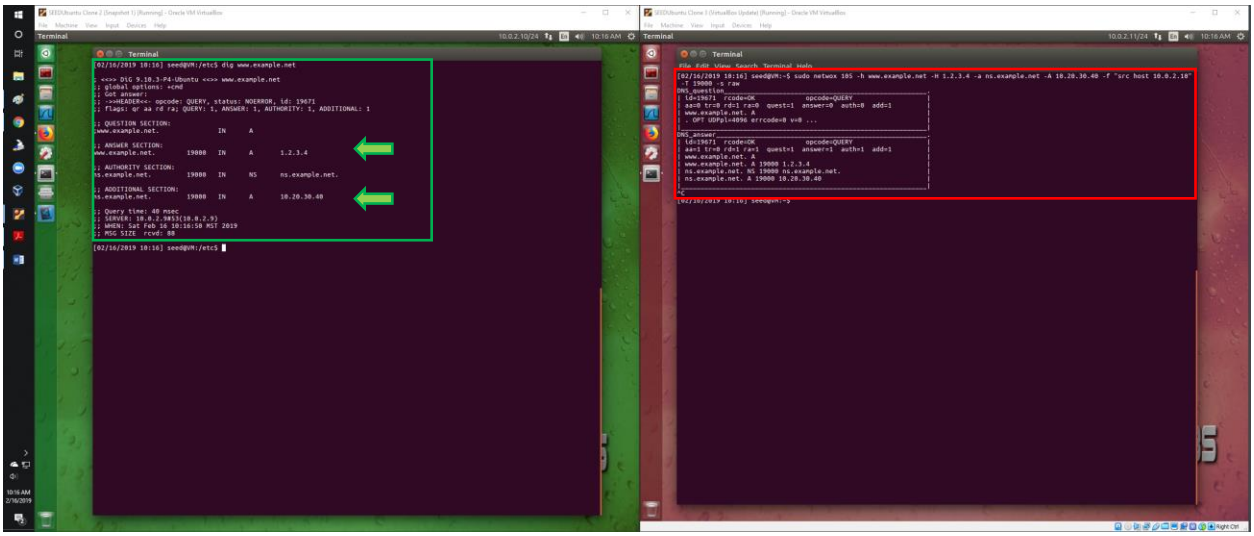
In the figure below, on the Green VM we see the dig command for www.example.net. Here we can see the name resolves to www.example.net's actual IP address as resolved by the Local DNS.



In the figure below, we attack the Green VM from the Red VM. We issue the command below which sniffs DNS requests and spoofs DNS replies. We target the IP address of the user system and specify a different address for the www.example.net and a different address for the name server. The command is as follows:

sudo netwox 105 -h www.example.net -H 1.2.3.4 -a ns.example.net -A 10.20.30.40 -f "src host 10.0.2.10" -T 19000 -s raw

- -h www.example.net – hostname to spoof
- -H 1.2.3.4 – IP address of hostname
- -a ns.example.net – spoofed name server which resolves example.net hosts
- -A 10.20.30.40 – nameserver IP address
- -f "src host 10.0.2.10" – the filter we use to specify the Green VM's IP address as the target.
- -T 19000 – time to live (in seconds)
- -s raw – send raw packet



Observations / Explanations

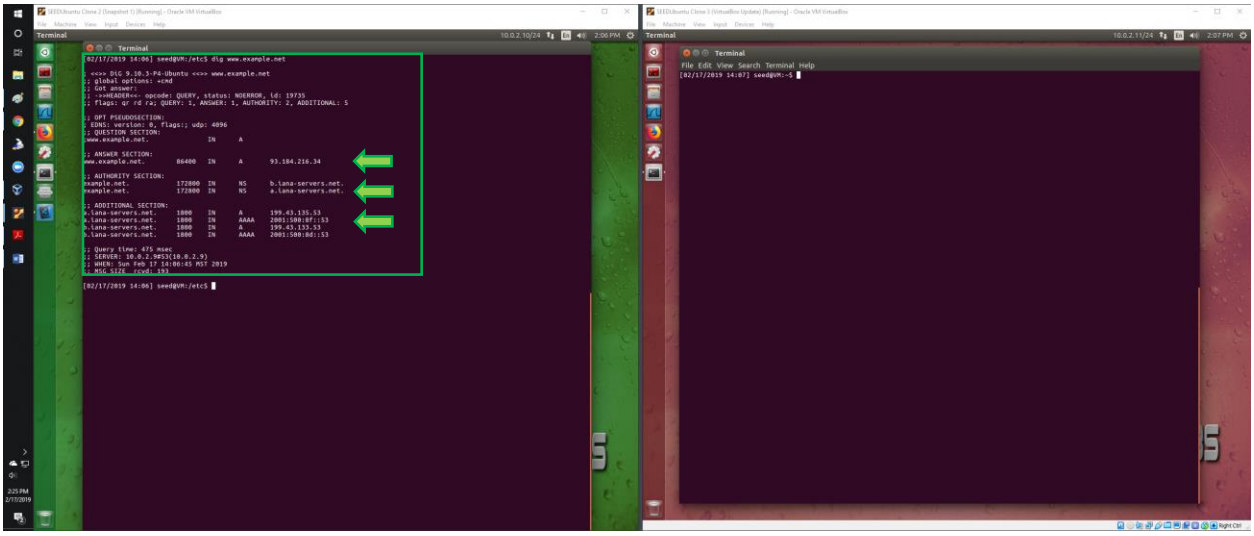
In this task we launched and attack on the user machine. Before the attack, we see the resolution of www.example.net as it should be – pointing to the correct DNS looked-up IP address. After the attack, the dig command reveals that our spoofed replies did indeed work and the dig results point to our spoofed IP address for www.example.net.

Compared to the previous attack, this one did not require the user machine to be compromised. It simply attacked the DNS request by sniffing and spoofing replies.

Task 6: DNS Cache Poisoning Attack

Attack the Local DNS and perform a cache poisoning attack.

In the figure below, we see the dig of www.example.net on the Green VM. This host is not a host which is part of our domain, so the Local DNS will attempt to 1) look at its cache to see if the hostname is cached and, if not, 2) go out to the Root DNS for resolution.

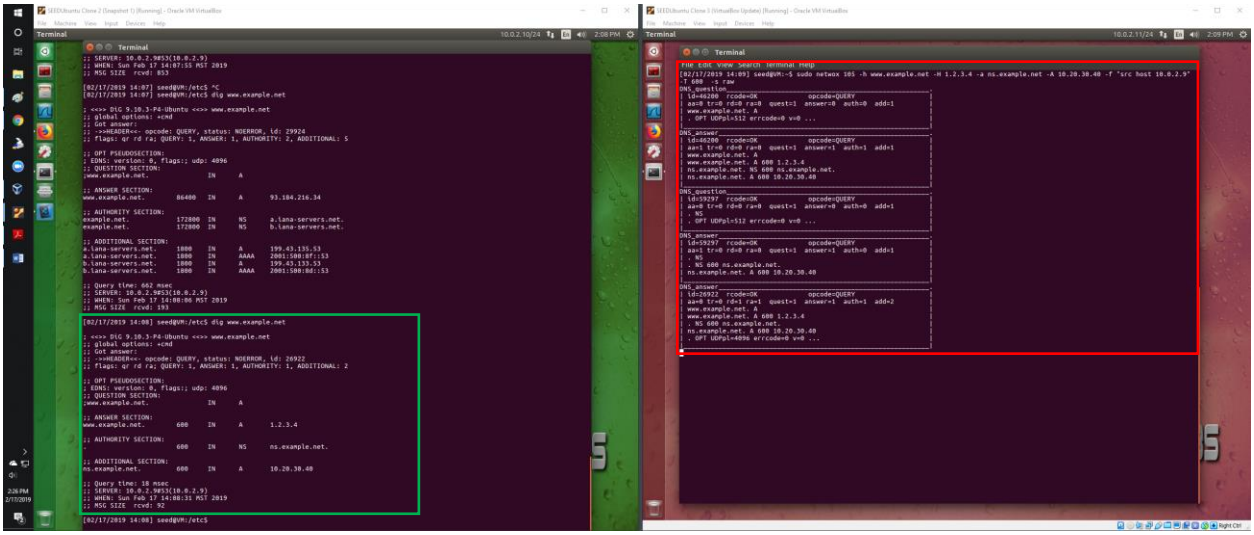


In the figure below, we execute the attack on the Red VM. The command is below. ***Please note that prior to executing the attack, we clear the DNS cache by "sudo rndnc flush" on the Blue VM.***

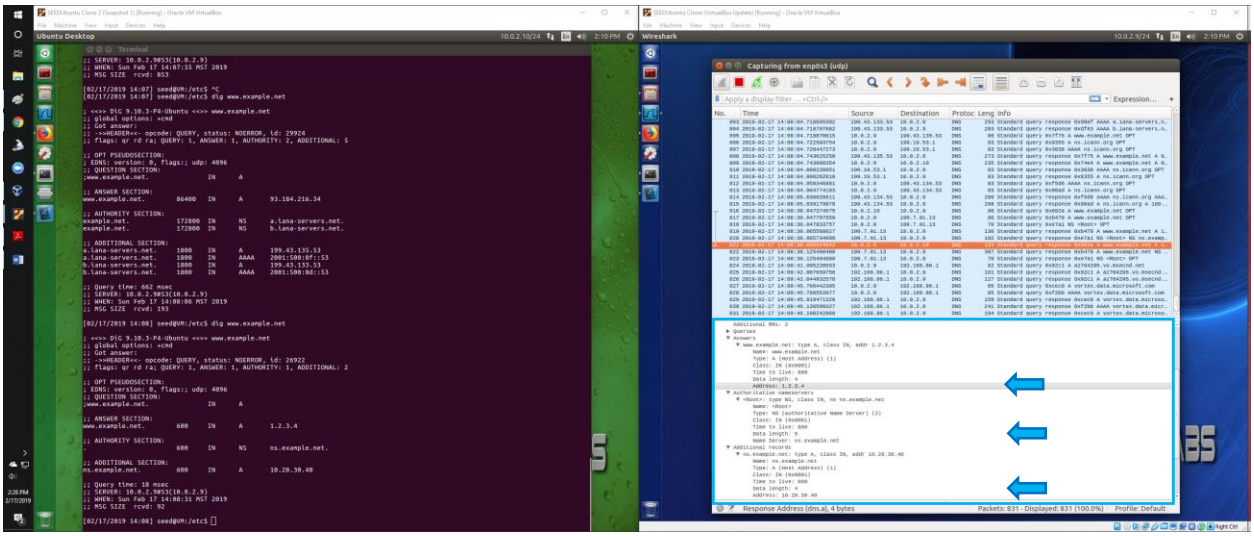
sudo netwox 105 -h www.example.net -H 1.2.3.4 -a ns.example.net -A 10.20.30.40 -f "src host 10.0.2.9" -T 600 -s raw

- -h www.example.net – hostname to spoof
- -H 1.2.3.4 – IP address of hostname
- -a ns.example.net – spoofed name server which resolves example.net hosts
- -A 10.20.30.40 – nameserver IP address
- -f "src host 10.0.2.9" – the filter we use to specify the Blue VM's IP address as the target. This is the DNS Server.
- -T 600 – time to live (in seconds)
- -s raw – send raw packet

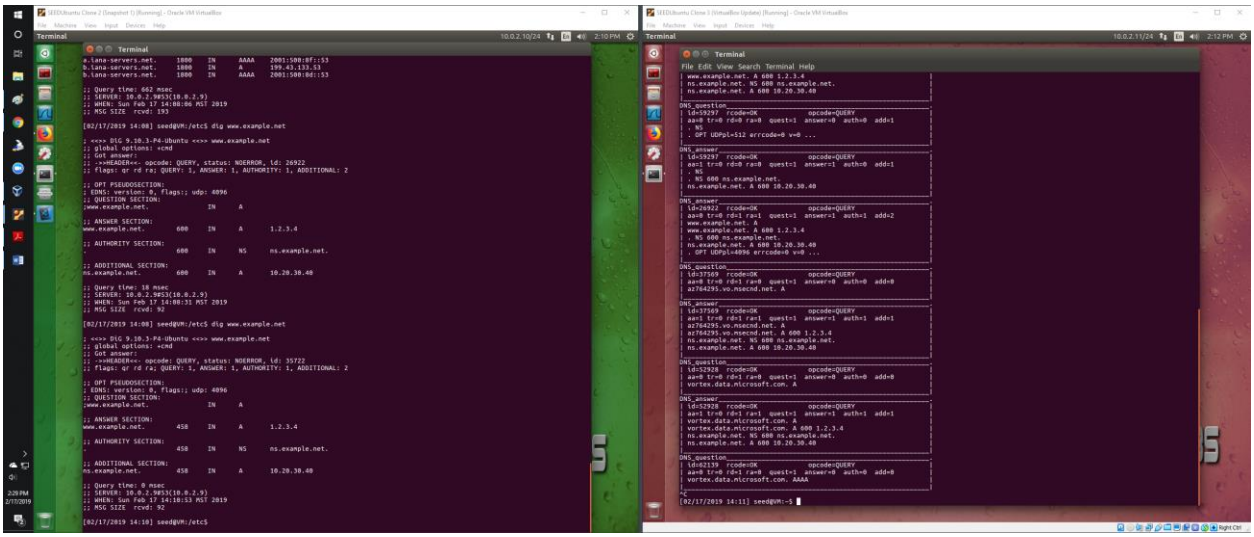
We can see in the Green VM, that after the attack, the IP address of the www.example.net is our spoofed 1.2.3.4 value. Also, the nameserver / IP are also spoofed.



The figure below shows the Blue VM WireShark messages. The highlighted message is the spoofed DNS response.



After killing the attack (CTRL+C in Red VM), when we dig www.example.net, we can still see the DNS Cache Poisoned values for the www.example.net hostname and the nameserver -- even though the attack is no longer running. That is because the DNS cache has been successfully poisoned and is retaining the DNS values which we spoofed!



Observations / Explanations

In this lab, we took the previous task a step further. Instead of simply spoofing the user system. We spoofed the DNS request and poised the DNS cache. We did this by using netwox, like before, but instead targeted the DNS server.

We sent the attack to the DNS server which looks for DNS requests for www.example.net. When it sees it, from the Green VM, it responds with a spoof reply that correctly answers the request and gets cached on the DNS server.

The next time the Green VM tries to resolve the www.example.net hostname, it uses the Local DNS cached value, which will stay in the cache up to our TLL (10 minutes).

By executing the attack on the Local DNS server, we have now affected all systems which use the Local DNS for hostname resolution, which can be quite impactful.