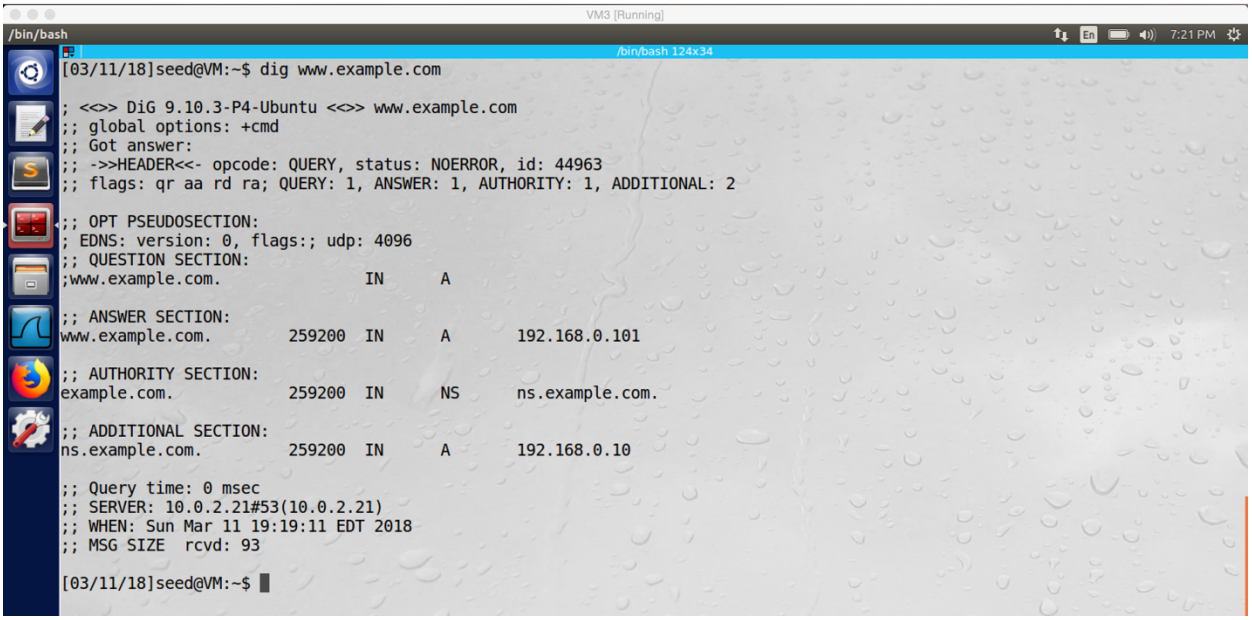
CSE644 Lab5
Yishi Lu
3/23/2018

In this task, I use three VMs, they are VM1, VM2, and VM3. They are all in the same NatNetwork.
VM1, it has IP address 10.0.2.20, it is the attacker machine.
VM2, it has IP address 10.0.2.21, it is the DNS server machine.
VM3, it has IP address 10.0.2.25, it is the user machine.

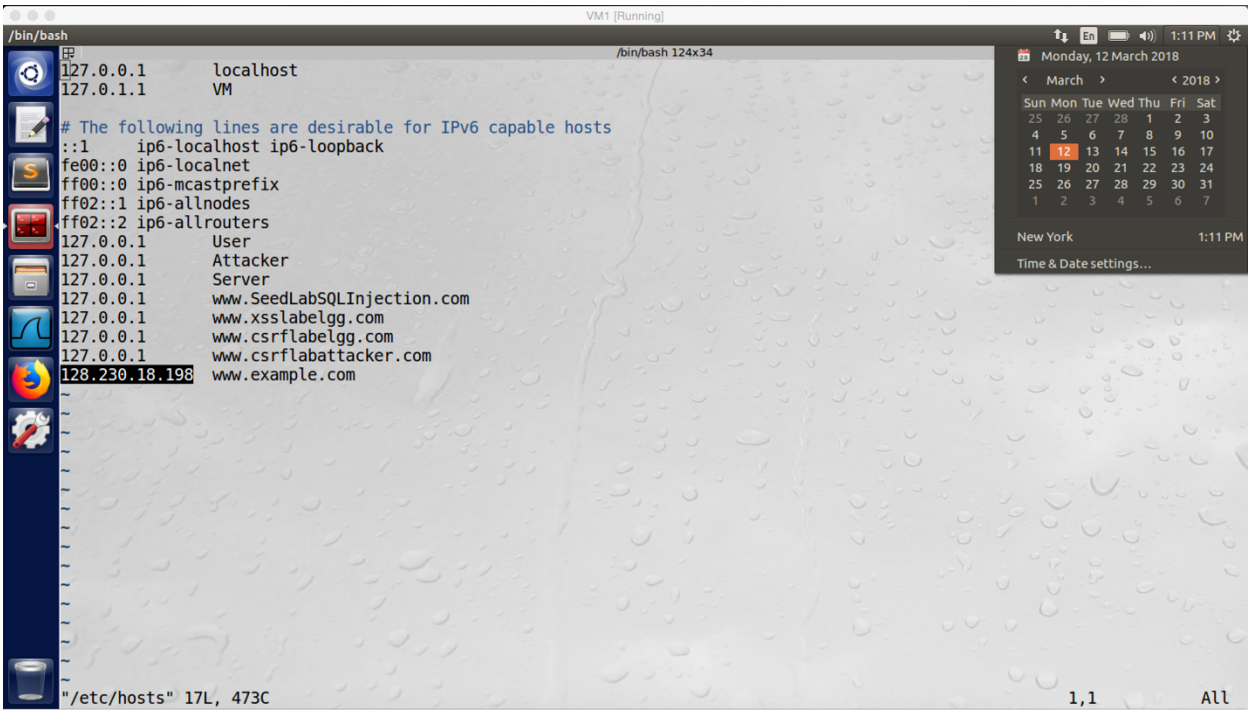
Firstly, I follow the steps on the lab description to setup the DNS server machine and user machine.



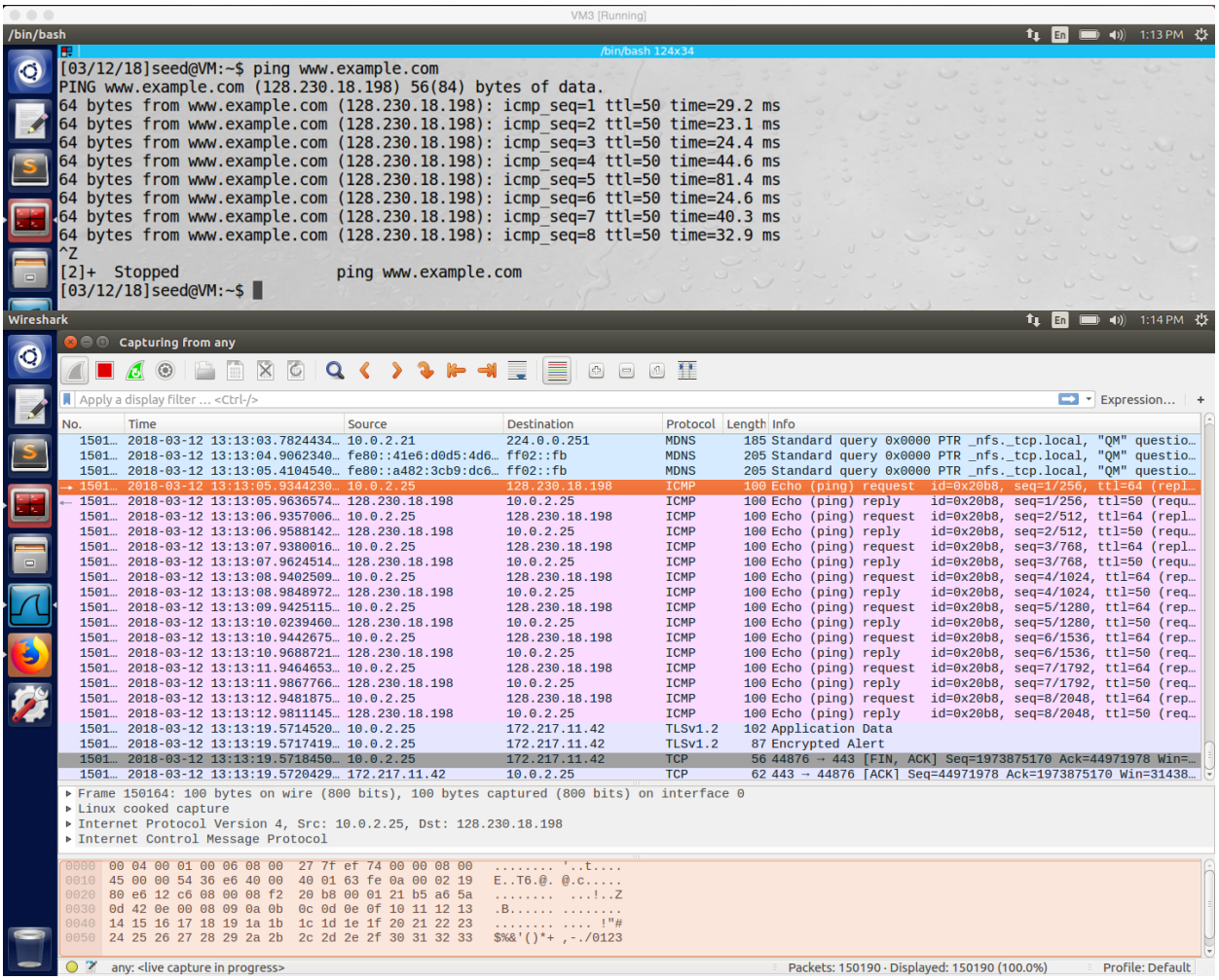
After setup, we can make query to the local DNS server (VM2), and the local DNS server returns the answer to the user (VM3) successfully.

Task1: Attackers have already compromised the victim's machine

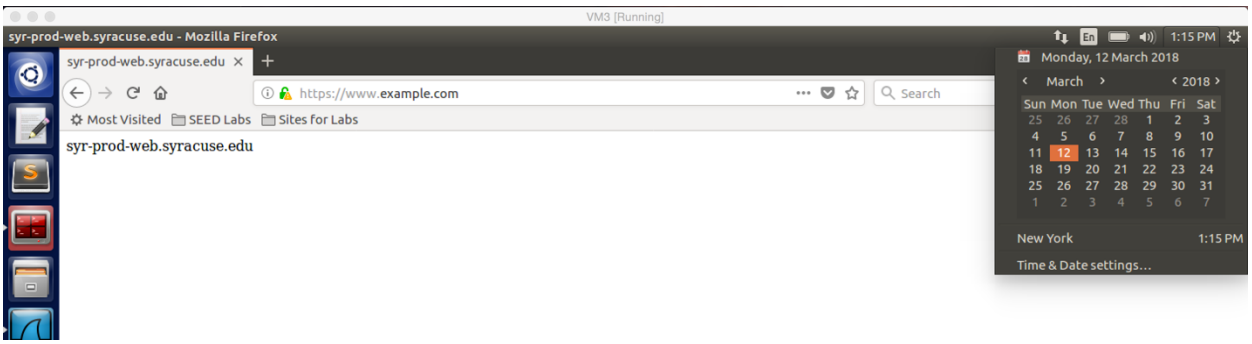
screenshot1 on VM1. Assume the VM3 is compromised to the attacker machine, so we connect VM3 from VM1 by using SSH.



screenshot2 on VM1. We open the hosts file of VM3, and we added new entry to this file.



screenshot3 on VM3. After we added the entry, we go back to VM3 and ping www.example.com, and then we get reply from 128.230.18.198.



screenshot4 on VM3. After we added the entry, we enter www.example.com on the browser, and then we got the above page

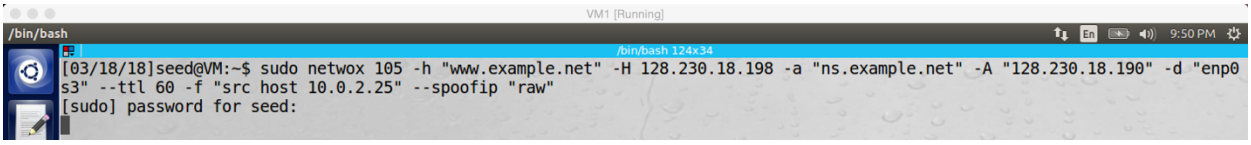
Observation and Explanation:

In this task, we assume the attacker machine (VM1) has granted the root privilege of the user machine (VM3). Firstly, we connect to VM3 from VM1 by using SSH (screenshot1). And then we open the /etc/hosts file of VM3 and modify its content. For the modification, we added

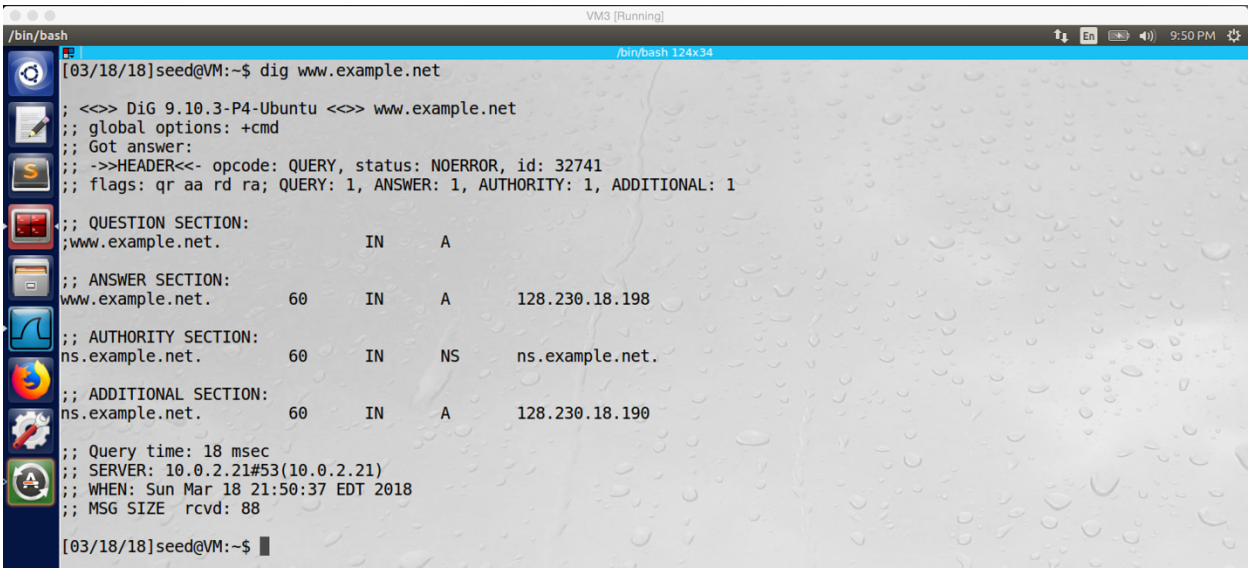
a new entry, in my case, it is “128.230.18.198 www.example.com” (screenshot2). This entry provides IP address to the domain www.example.com. Therefore, when the user enters example.com in the browser or ping example.com on VM3; the user machine will not send query to DNS server, instead it will look at the hosts file and get the IP address directly. Afterwards, I go back to VM3, when I ping example.com, it actually pings 128.230.18.198, and the ICMP packet is transferred between 128.230.18.198 and VM3 (screenshot3); moreover, when I enter www.example.com in the browser, the page is redirected to 128.230.18.198 (screenshot4).

Task2: Directly Spoof Response to User

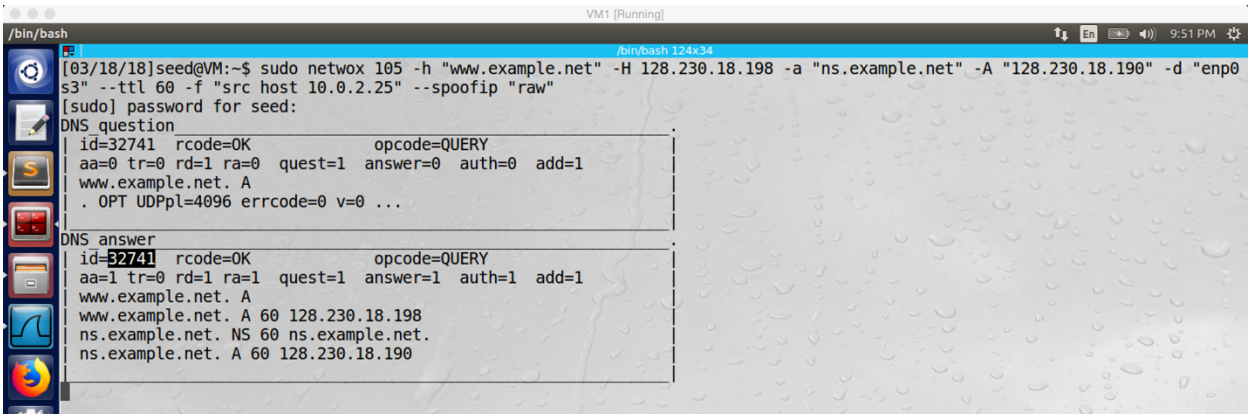
In this task, because the DNS server and attacker machine are in same the local network, I use www.example.com to try a lot of time, but, every time the user machine receives the DNS server response first. Therefore, as TA Kailiang suggestion, I use www.example.net (the local DNS server VM2 has no answer for this domain) to do this task.



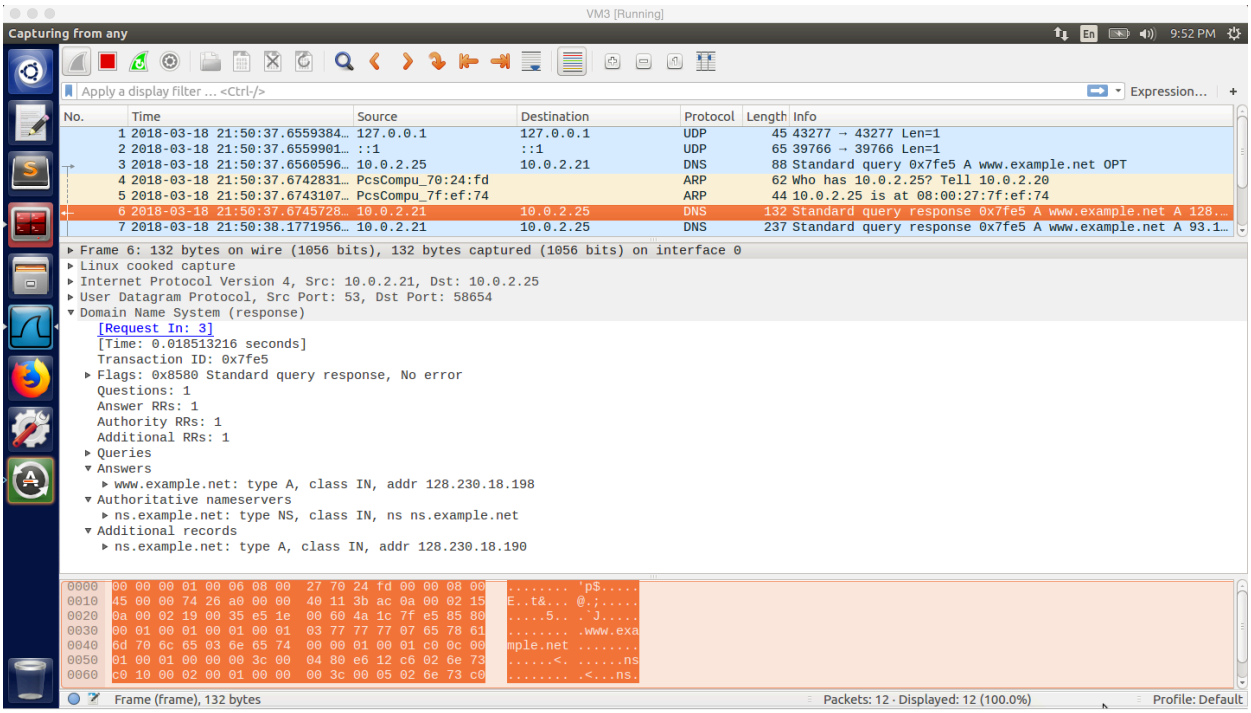
screenshot1 on VM1. Run netwox 105, this command first sniffs the traffic of VM3. If there is a DNS query from VM3, then it will send spoofed DNS response packet to VM3.



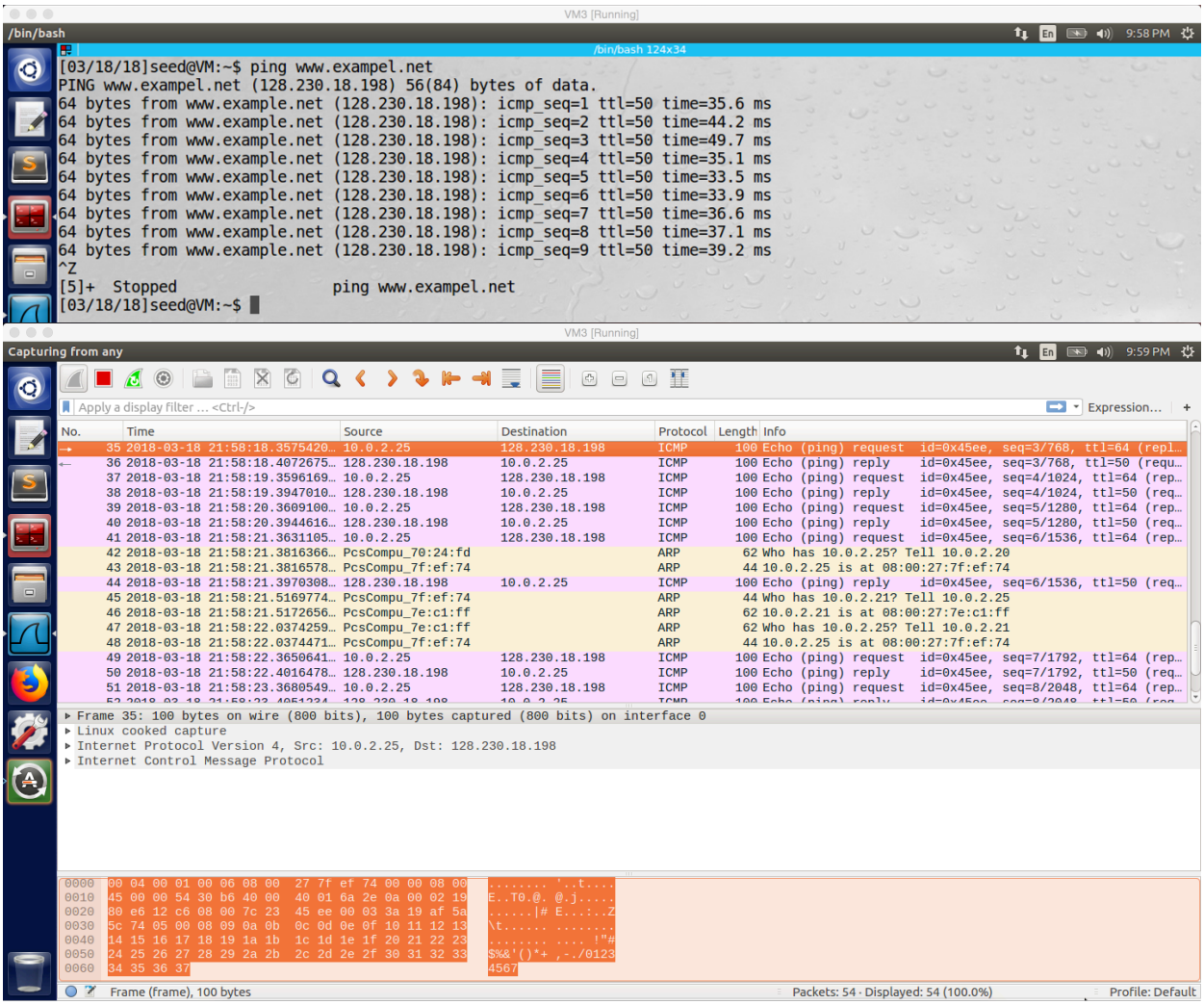
screenshot2 on VM3. Running “dig www.example.net” on VM3, because the spoofed DNS response reaches VM3 earlier than the real DNS response, we see that the IP addresses in answer section and additional section are “128.230.18.198” and “128.230.18.190”, these values are set by netwox 105 on VM1



screenshot3 on VM1. The attacker machine prints the DNS query which it sniffed from the traffic of VM3, and it also prints the spoofed DNS response



screenshot4. The entry No.6 is the spoofed DNS response which is sent by the attacker machine VM1. The entry No.7 is the real DNS response which is sent by the DNS server VM2. Because the spoofed response arrives first, VM3 accepts the spoofed response and discards the real DNS response



screenshot5 on VM3. We successfully ping www.example.net, and the IP address is 128.230.18.198

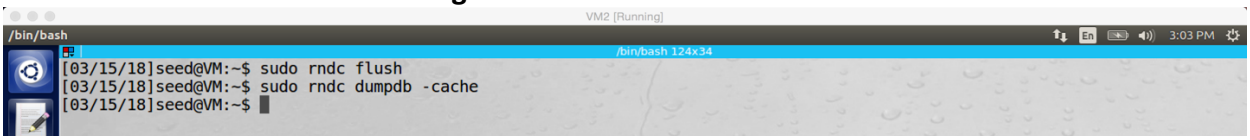
Observation and Explanation:

If the attacker machine does not have the root privilege of the user machine, then attacker can attack the user machine by sending spoofed DNS response packet. When the hosts file does not contain the IP address of the desired domain, the user machine will send DNS query to the local DNS server for asking desired domain IP address. Therefore, the attacker can send spoofed DNS reply to the user machine when the user is waiting for the answer; if the user machine received the spoofed DNS reply earlier than the real DSN reply, it will accept the spoofed DNS reply, and it will use the IP address provided by the attacker as a trustable answer, then the attack is successful.

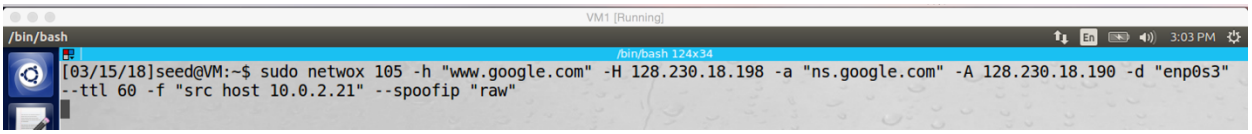
In this the task, we want to perform such attack, and we need to use tool netwox 105, this tool will do sniffing and spoofing. Basically, it monitors the user machine's traffic, if there is a DNS query from target machine, then the tool will forge a DNS response packet and send it to the target machine. We run netwox 105 on VM1 (screenshot1), -h is hostname, -H is hostname IP, -a is authoritative name server, -A is authoritative name server IP, -d is device name, -ttl is

the value of time to live, and the -filter is pcap filter (here we set “src host 10.0.2.25”, so every DNS query packet of the user machine VM3 will be sniffed and spoofed). After running this command on VM1, we go back to the VM3 and run “dig www.example.net”, this command will send a DNS query to local DNS server (VM2). Meanwhile, VM1 sniffed this DNS query and send a spoofed DNS response to VM3. Because VM2 does not have the answer of the query, it needs to ask other DNS server on the internet for answer. Therefore, the local attack machine has high chance to beat the speed of the real DNS response. After VM3 received the forged DNS response, we can see the answer section is 128.230.18.198 and the additional section change to 128.230.18.190 (screenshot2). Go back to VM1, we also can see the sniffed and spoofed DNS s are printed by netwox 105 (screenshot3). In screesnshot4, we see there are two DNS response sent by 10.0.2.21. Actually, the first one with entry No.5 is the spoofed DNS response which is sent by VM1, and the send one with entry N0.6 is the real DNS response which is sent by VM2. Because the spoofed DNS response arrives first, so VM3 accepts the forged DNS response and discards the real DNS response. Afterwards, we can also try to ping example.net, and we succeeded (screenshot5), the ICMP packers are transferred between 128.230.18.198 and VM3.

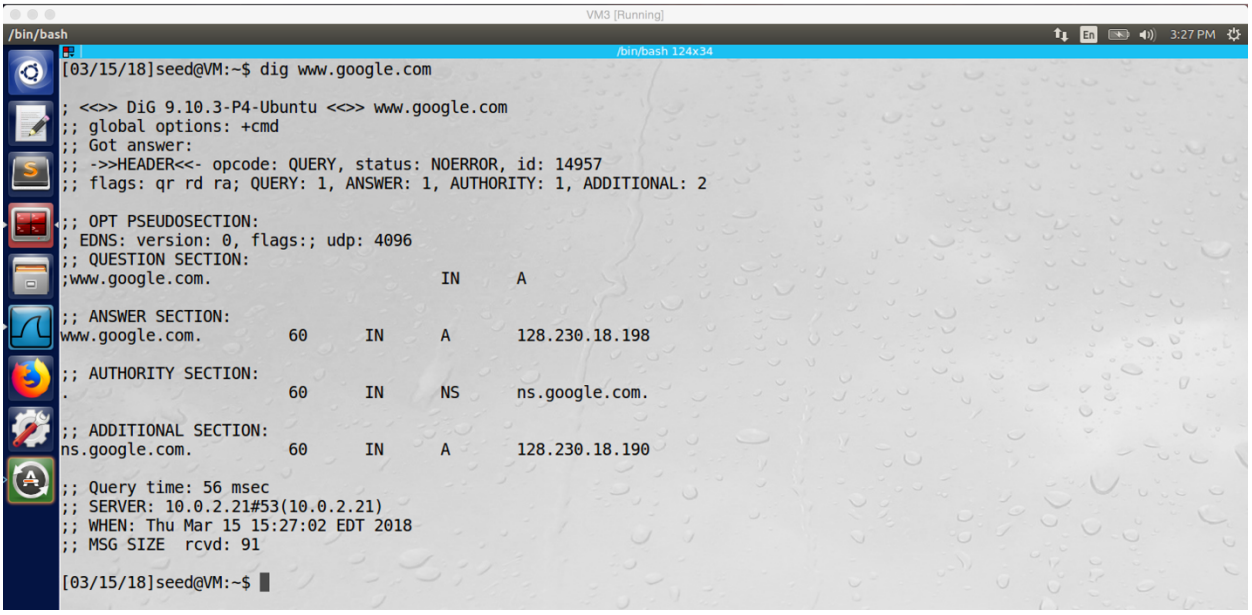
Task3: DNS Server Cache Poisoning



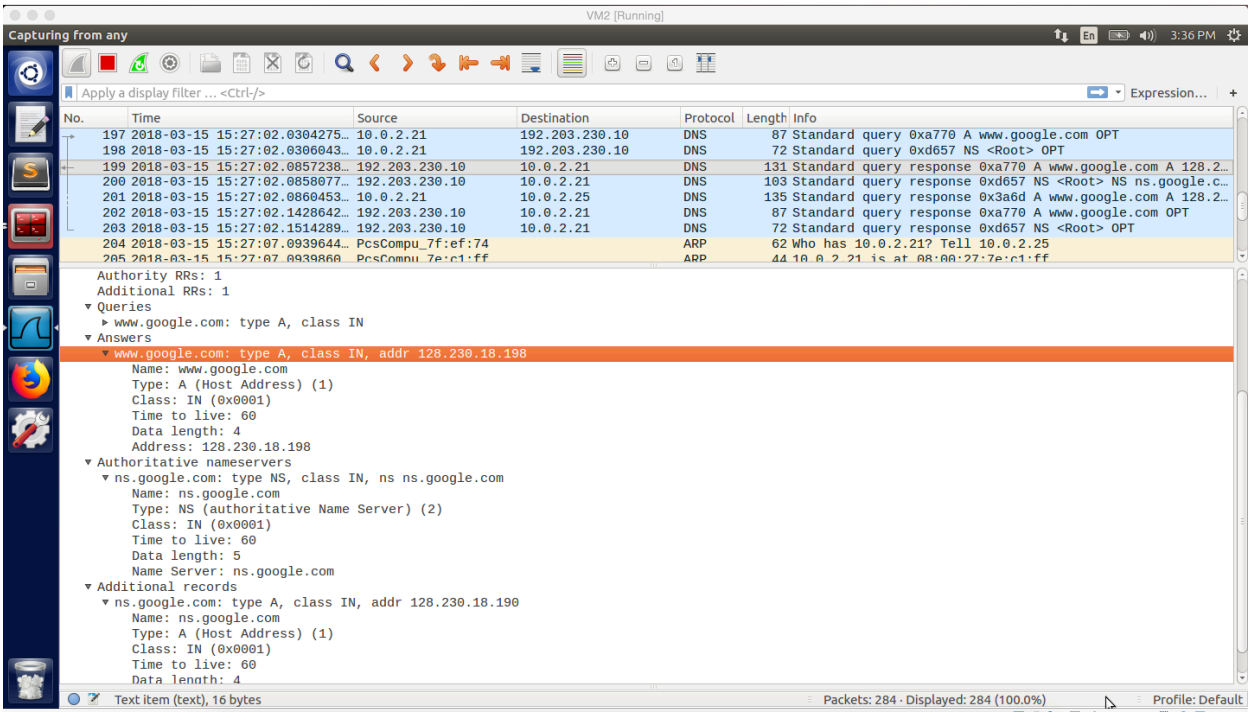
screenshot1 on VM2. Before attack, we clean up the DNS cache first



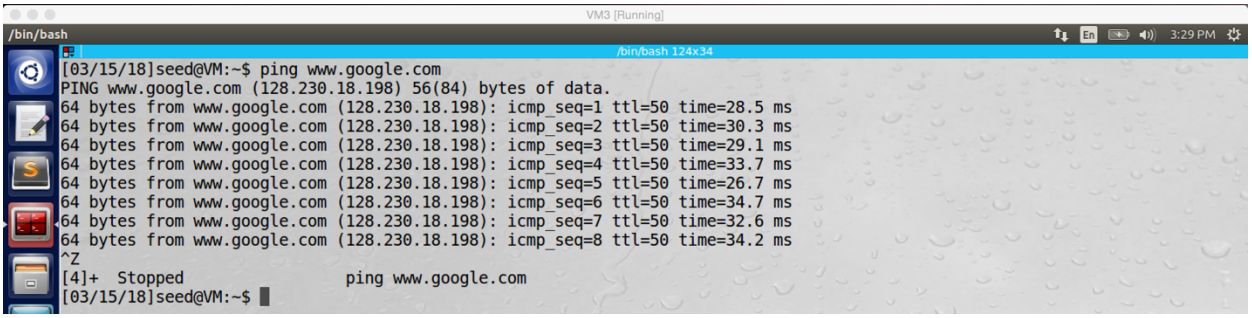
screenshot2 on VM1. In this task we change host name to www.google.com, name server to ns.google.com and target IP address to the local DNS server 10.0.2.21



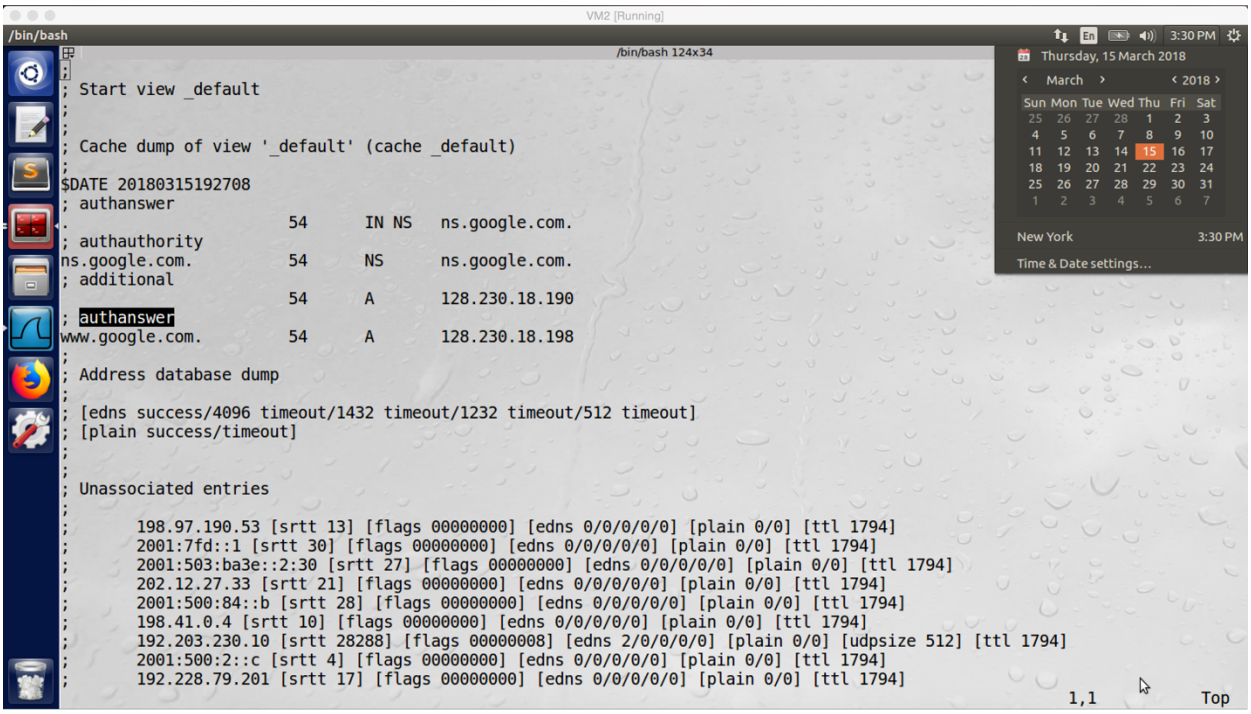
screenshot3 on VM3. We run “dig www.google.com” on VM3, the IP address in answer section is changed to 128.230.18.198, and the IP address in the additional section is changed to 128.230.18.190. The attack is successful.



screenshot4. From the screenshot of Wireshark, we can see; After we run “dig www.google.com” on VM3, the DNS query is sent from VM3 to VM2; and then VM2 send DNS query to 192.203.230.10. Afterwards, there two DNS responses sent from 192.203.230.10 to VM2. The first one (N0.199) is the spoofed DNS response which is sent by VM1. The second one (No.202) is the real DNS response which is sent by the real DNS server with IP address 192.203.230.10. Because the spoofed DNS response arrives first at VM3, VM3 accepts the forged DNS response and discards the real DNS response



screenshot5 on VM3. When we ping google, we actually ping 128.230.18.198



screenshot6 on VM2. We check the dump.db file, the “www.google.com A 128.230.18.198” is cached in the DNS server

Observation and Explanation:

Except to attack the user machine, we also can attack on the local DNS server, it is called local DNS server cache poisoning attack. In this task we will perform such attack. When the user makes DNS query to the local DNS server, the server will look at its cache; if it has the answer, it sends the answer back to the user immediately. Otherwise, it sends DNS query to other DNS servers on the Internet for asking the answer. If attacker’s spoofed DNS response packet can reach the local DNS server earlier than the real response from other servers, then the local server will accept the forged answer. And then it will send the forged answer to the user. Moreover, the local DNS server will cache the forged answer, so any user makes same query to the local DNS server again when the cache is still alive, the user will get the forged answer as well. This is what we want to do in this task: sending spoofed DNS reply from the attacker VM1 to local DNS server VM2, and our goal is to make the local DNS server to cache www.google.com with IP address 128.230.18.198.

Firstly, we clean up the cache of the DNS server (screenshot1). And then we run netwox 105 again, but this time, we change three parameters (screenshot2). We changed hostname to google.com, authoritative name server to ns1.google.com, and filter to “src host 10.0.2.21”. In our case, for the first two parameters, if our attack succeeds, any user machine who make query for asking google.com IP address to VM2 will get SU’s IP address (128.230.18.198) as reply. For the last parameter, because we attack DNS server this time, we need to sniff and spoof any DNS packet of the server, and 10.0.2.21 is the DNS server’s IP address. Afterwards, we go back to VM3 and run “dig www.google.com” (screenshot3), the IP addresses in answer and additional section are 128.230.18.198 and 128.230.18.190, this is what we expect. As the screenshot of Wireshark (screenshot4), after we run “dig www.google.com” on VM3, the DNS

query is sent from VM3 to VM2; and then VM2 sends DNS query to 192.203.230.10. Afterwards, there two DNS responses sent from 192.203.230.10 to VM2. The first one (N0.199) is the spoofed DNS response which is sent by VM1. The second one (No.202) is the real DNS response which is sent by the real DNS server with IP address 192.203.230.10. Because the spoofed DNS response arrives first at VM3, VM3 accepts the forged DNS response and discards the real DNS response. Moreover, after VM2 accepted the forged DNS response, it sends the forged answer to VM3 (No.201). And then I also ping www.google.com, the ICMP packet is transferred between 128.230.18.198 and VM3 as well (screenshot5). Finally, I check the DNS server cache (screenshot6), clearly google.com is cached with SU's IP address - 128.230.18.198. Therefore, our attack succeeds.