Kaminsky Attack – CS528 - Lab 4

VM names and IP Addresses:

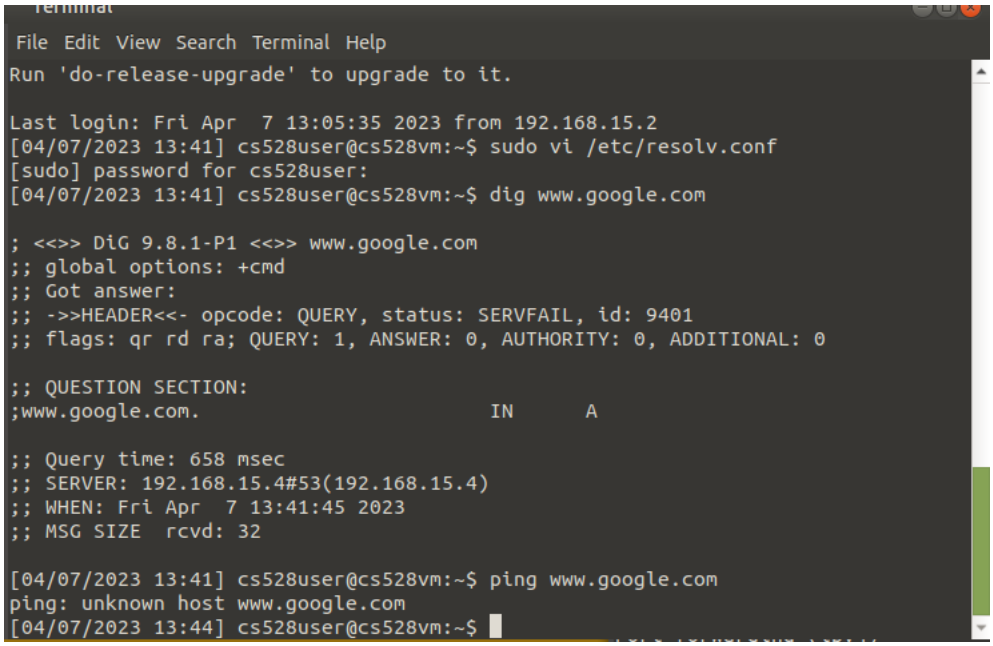
DNS SERVER : CS528_vm1 : 192.168.15.4

USER : CS528_vm2 : 192.168.15.5

Attacker : 192.168.15.6

2.2 Check that the DNS configuration is working by running:

dig www.google.com

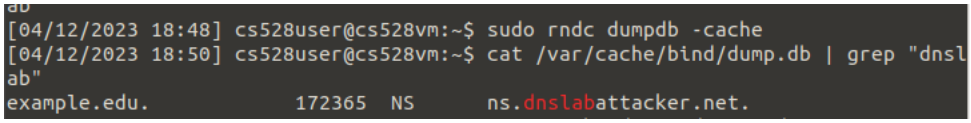


Task 1 : Poison the cache of Apollo

The file provided (UDP.c) is starter code for continuously fabricating DNS request packets. To fabricate a correct DNS response packet, I first captured the DNS response from the correct example.edu name server (found using dig www.example.edu) and observed what fields were required and how they changed.

- Using the skeleton code UDP.c, we construct a DNS packet to spoof from the client to the server. Here we fix the source port as 33333 for the sake of convenience, set the destination port as 53 (DNS query port), and fill in the packet with a DNS query field starting with twysw.example.com and keep changing 1 character at random.
- dnsResponseBuilder() : Here we create the reply packets using the response function, we construct a DNS response packet by filling in the query same the query spoofed in the previous task. Reply with the correct IP and the target the authority section using ns.dnslabattacker.net as the name server.

- we send multiple replies to every query such that we poison the cache by attacking the authority section and as shown below in my dump.db the name server corresponding to example.edu has been changed to ns.dnslabattacker.net



Task 2:

In this section, I verified that the attack was successful as mentioned in the lab handout. The general idea of the verification process was to manually set the IP address of the ns.dnslabattacker.net server at Apollo to be that of the dns_attacker machine so that whenever a dns_usr machine queried for a *.example.edu URL, Apollo would look up its cache and find that the name server corresponding to *.example.edu is ns.dnslabattacker.net and return the corresponding IP as that of dns_attacker. Once the usr_machine has the IP of the (attacker) DNS server, it queries that and the dns_attacker responds with the spoofed IP of the URL. In this way, the attack is shown to be successful. output of running the dig command at dns_usr. It can be seen that the IP returned for www.example.edu is 1.1.1.1 (the spoofed IP). Moreover, note that the IP of the name server is shown as 102.168.15.9, which is dns_attacker's IP. Finally, note that the IP of the DNS server queried is 192.168.15.7 (Apollo's) IP. This verifies that the DNS server has been compromised and the attacker now has control over what IP to respond with.

Answer for the question under task 2:

Apollo will not accept this because we are dealing with two different zones here: the.edu zone and the.net zone. When sending forged response packets to Apollo, we explicitly specify that the authoritative name ns.dnslabattacker.net refers to the server for the domain example.edu. Apollo accepts the response and updates its record since we are able to spoof the IP of the true name server for example.edu, which is recognized as having administrative authority to specify values for the example.edu domain. However, as we are not spoofing to the ns.dnslabattacker.net and cannot, the information provided in the supplementary section that provides the IP of the ns.dnslabattacker.net is ignored. We do not have the power to change the IP address for