DNS Poisoning

Lidor Keren Yeshuah: 213205230

Roy Simanovich: 209396159

In general, DNS poisoning is a cyber attack that injects wrong information (in our case -> wrong ip addresses for domains) to DNS resolvers' cache.
As a result a different site may appear instead of the original.

Description of the attack:

The user wants to get the site www.example.com , therefore, The DNS resolver of the network starts searching (in case it doesn't contain it in its own cache) for the IP address for the requested domain (starts in the root servers -> TLD servers -> auth servers).

In this point we (aka the attackers) are spamming the DNS resolver with DNS responses (spoofed DNS packets) and when the DNS resolver gets our "fake" response instead of the real response from the auth server then we have poisoned the DNS resolver.

And now when the DNS resolver stores the wrong answer in its cache it send the wrong answer to the user which eventually gets a wrong website.

Our Code:

We have 2 python files.

Request.py -> this file contains the DNS request written in python with scapy and writes the packet to a binary file.

replay.py -> this file contains the DNS reply written in python with scapy and writes the packet to a binary file.

Attack.c -> loads both binary python files and on each send of the DNS request we are sending 2500 DNS replies.

writing a packet in python with scapy its way more easier than writing it in C but sending it in C is way more faster than sending it in python.

How to run the program:

1. Download all the files
2. Open a terminal and type: sudo docker-compose up (all the dockers should be running)
3. Open terminal for the user, local-dns and seed-attacker and type: sudo docker exec -it <container id> /bin/bash
4. In the seed attacker docker:
   4.1 apt-get update
   4.2 apt-get install gcc
   4.3 cd volumes (enters to the folder)
   4.4 python3 request.py (run the python prog)
   4.5 python3 replay.py (run the python prog)
   4.6 gcc -o attack attack.c (compile the c prog)
   4.7 ./attack (run the c prog)
5. In the local-dns docker:
   5.1 rndc flush (clears the cache)
   5.2 rndc dumpdb -cache && grep attacker /var/cache/bind/dump.db
   (updates the cache and helps to see the changes)
6. int the user docker:
   6.1 dig www.example.com (performs DNS lookup)

in the photo we can clearly see that we have successfully changed the ip address to 1.2.3.5