Cybersecurity Lab II

Lab 10

DNS Attacks

The goal of the lab is to:

(a) Setup a local DNS server
(b) Attack the local DNS server using the cache poisoning technique.

> 🛇 *Don't attempt to attack public DNS servers!*

1. Environment

Setup. You need to run three VMs to perform this lab: an attacker, user machine, and local DNS server machine. The user machine initiates DNS requests to the local DNS server by using `dig`, and the attacker machine starts the attacks by means of sniffing and spoofing. The environment you created for Lab 9 will be useful here, but you may need to make a modification: **all** DNS traffic from box1 should exit through box2, not just traffic between box1 and box3.

Netwox Tool. You can use this tool to send network packets with different contents. To install `netwox` on Ubuntu:

```
$ sudo apt-get install netwox
```

`Netwox` consists of a set of tools, each of which has a number. For example, you can run a `Netwox` tool as follows:

```
$ sudo netwox <number> [parameters …]
```

You can check the help message for each tool by running this command:

```
$ netwox <number> --help
```

Scapy. Some of the tasks will be conducted using `scapy`. Unlike `Netwox`, `scapy` is well maintained, and has extensive online resources (e.g., documentation, user guides, forums, conferences, etc.)

2. Tasks

Task 1: Setting up the machines (30%)

The goal of this task is to configure a local DNS server and create a DNS zone for "example.com".

Setting up the User machine

You need to configure the user machine to use a specific local DNS server (which will be another VM). At a high level, you need to configure the /etc/resolv.conf configuration file. Depending on your system configuration, there are different ways to achieve this. For instance, if your VM uses DHCP to obtain IP addresses and local DNS servers, DHCP clients will overwrite the /etc/resolv.conf.
One way to get the nameserver information into /etc/resolv.conf without worrying about the DHCP is to add an entry to the /etc/resolvconf/resolv.conf.d/head file:

```
nameserver <DNS_Server_IP>
```

Then, you run the following command:

```
$ sudo resolvconf -u
```

Another way is to stop the NetworkManager from overwriting nameserver configurations. However, you need to carefully perform these steps as they may impact other networking aspects in your VM. First, you need to configure the file /etc/NetworkManager/NetworkManager.conf and comment out the entry for dns. Then, you remove the /etc/resolv.conf symbolic link, and create a new file. Finally, you write your /etc/resolv.conf version including the required nameserver configuration.

Setting up the local DNS server

Setting up a local DNS server is straightforward. You need to install the bind software, configure bind, and create zone files. In the following, we assume that the used OS is Ubuntu, however, the same configurations apply for other systems (potentially in different paths).

First, to install bind, you can run the following commands:

```
$ sudo apt-get update
$ sudo apt-get install bind9 bind9utils bind9-doc
```

Note: Different package managers have different package names for bind. For instance, in yum, you should install the bind package.

Second, we need to configure the DNS server to disable DNSSEC and set the cache dump file. In bind, the main configuration file is /etc/bind/named.conf, and this file includes other configuration files that contain the actual configurations.
In our scenario, we need to configure /etc/bind/named.conf.options as follows:

```
options {
    dump-file "/var/cache/bind/dump.db";
    // dnssec-validation auto;
    dnssec-validation no;
};
```

Everytime you configure the DNS server, you need to restart it as follows:

```
$ sudo service bind9 restart
```

The following two commands are useful for the purposes of this lab as well:

```
$ sudo rndc dumpdb -cache // Dump the cache to dump-file
$ sudo rndc flush // Flush the DNS cache (i.e., remove the cache entries)
```

Third, we will host a sample zone in the DNS server. For simplicity, the required configuration files are attached as .txt files. You need to create the required configuration files in your VM, and copy the contents of the txt files to the corresponding configuration files. For instance, the contents of “zone.txt” should be added to the `/etc/bind/named.conf` file. And the contents of example.com.db.txt are to be added to `/etc/bind/example.com.db` file.

Testing the Setup

After you are done with all configurations, make sure to restart the DNS server and flush its cache. Using the user machine, your **first task** is to ping a server such as www.sfu.ca, and describe your observation. You should use tcpdump/Wireshark to show the DNS query triggered by your ping command.

Your **second task** is to fetch the “www.example.com” DNS records using the `dig` command. Similar to the first task, describe and explain your observations using proper screenshots from Wireshark.

Task 2: Cache Poisoning: Targeting a Single Hostname (30%)

The goal of this task is to spoof a DNS response to poison the DNS cache with an attacker-controlled IP address. Specifically, when the local DNS server receives a request from the user VM, it forwards it to the root nameservers. Your goal is to sniff the network, create a spoofed DNS response when a request is sent out by the DNS server, and send the spoofed response to the local DNS server. If the constructed response is *valid*, it will be accepted by the DNS server. Recall to flush the cache content before you start!

For this task, you will use the “`netwox 105`” tool to sniff the network and spoof the DNS responses for “www.example.net”. Notice that we do not spoof responses for “www.example.com” as they are already managed by the local DNS server (i.e., it will not send DNS requests). You can check the tool parameters by running this command: `netwox 105 --help2`
You need to set the spoofed IP addresses for www.example.net and ns.example.net to “192.168.0.5” and “192.168.0.6”, respectively.
For this task, as you need to sniff requests sent by the DNS server, you need to use the right `filter` argument in the `netwox` command. In particular, the `filter` should include “`src host <DNS_Server_IP>`". Also, you need to set the `--spoofip` argument to “`raw`” to spoof at the IP level.

You need to include the used `netwox` command in your report, and include proper screenshots from the `dig` and tcpdump commands as well as from the cache dump file to show the success of your attack.

Task 3: Cache Poisoning: Targeting a Whole Domain (40%)

For this task, you are asked to perform a cache poisoning attack that targets a whole domain instead of a single hostname.

Remember to flush the cache content of the local DNS server before you start!

Your goal is to launch a DNS cache poisoning attack to control the whole domain “example.net” without the need to spoof responses for every hostname, e.g., www.example.net.
To achieve this, you need to sniff the DNS request, and send a spoofed DNS response as you did in Task 2. However, you need to control two sections in this attack. First, you need to spoof the Answer Section for “www.example.net” so that its IP address becomes “10.0.0.24”. Second, you need to spoof the Authority Section for “example.net” to point to the attacker-controlled nameserver. Specifically, you need to add the following record when you spoof the response:

```
;; AUTHORITY SECTION:
example.net. 259200 IN NS ns1.cmpt783.org.
```

If this entry is cached by the local DNS server, ns1.cmpt783.org will be the nameserver for future requests of any hostname in the example.net domain. Since ns1.cmpt783.org is controlled by the attacker, it can be configured to reply with specific IP addresses.

You need to use `scapy` (and Python3) to perform the sniffing and spoofing as we discussed in the lecture.

First, you need to show a screenshot of the outputs of a `dig` command for “www.example.net”.
Second, you need to show that the above Authority Section entry was cached by the local DNS server. After the cache is poisoned, run a `dig` command on any hostname in the example.net domain, e.g., mail.example.net, and use tcpdump to observe the DNS requests and responses. Include proper screenshots to show the success of your attack (from `dig`, tcpdump, and `ping` commands).

3. Submission

You are required to submit:
(1) All source code you implemented to complete the tasks.
(2) A detailed report.
The files should be compressed in a single (.zip) archive.