Assignment 2

CS448 Introduction to Information Security, KAIST
(2021 Spring)

Due: 11:59 PM (KST), April 14, 2021

INSTRUCTIONS TO STUDENTS:

- Any collaboration or assistance of any kind is strictly prohibited; all work must be your own.
- Your submission *will surely* be compared with the submissions for your peers for plagiarism detection (e.g., MOSS). Any academic dishonesty will be directly reported to the university.
- You have total five grace days in this semester.

1 True-False Questions (30 points)

1. True or false: The KAIST IT Department may not notice when there exist a certificate for the KAIST web domain that is never issued by KAIST.
(A) True, (B) False
2. True or false: The use of cipher suites in TLS that offer perfect forward secrecy guarantees the secrecy of session keys that will be generated in the future.
(A) True, (B) False
3. True or false: DNS-over-HTTPS prevents DNS recursive resolvers from learning the queried domains but cannot handle DNS spoofing attacks.
(A) True, (B) False
4. True or false: Unlike Bitcoin's proof-of-work that uses a single common puzzle for all, DDoS defense puzzles must be different for different clients.
(A) True, (B) False
5. True or false: We can fix the UDP servers, which are exploited for amplification DDoS attacks, by preventing them from returning more bits than they receive.
(A) True, (B) False
6. True or false: DDoS threats would disappear in a hypothetical world where a system admin can increase his server capacity indefinitely.
(A) True, (B) False

1.1 Submission

Submission. Write a text file `p1_S_#.txt`, where # denotes your student id. In the text file, write your answer letters (i.e., A or B) for six questions and use `newline` to separate them. For example,

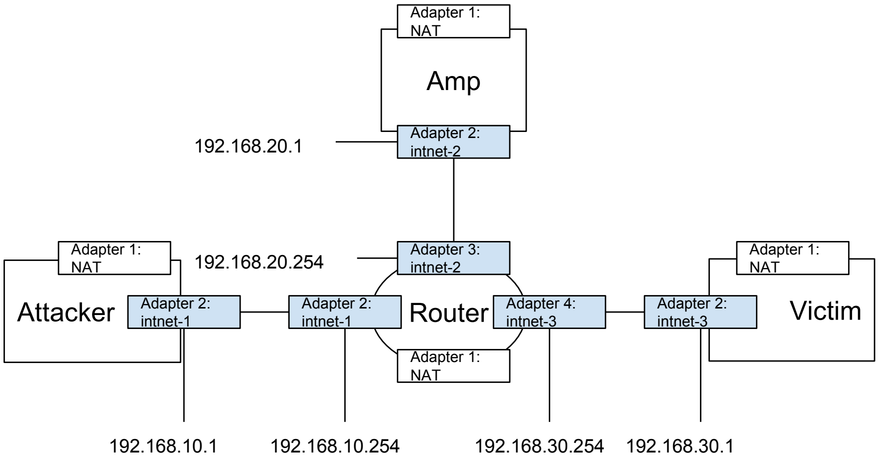

Figure 1: VM setup for amplification attacks. Only the non-NAT network adapters (i.e., blue-shaded parts) need to be manually configured. The rest has been already configured in their `vdi` images.

```
In p1_S_20210001.txt file:
    A
    B
    A
    B
    A
    B
```

Include this text file into the final tar file of your amplification DDoS attack; see the next section.

2 Amplification DDoS Attacks (70 points)

In this section, you will setup a new network topology with the provided virtual machines (VMs) that enables you to launch small-scale amplification DDoS attacks.

Warning. Do ***not*** test your attack scripts against services running on the Internet as it could lead to real DDoS attacks in the real world. Always make sure that all your attack activities stay within the virtual environment prepared for this assignment.

VM images. You will download the four required VM images from the following link: `https://drive.google.com/drive/folders/1gTzcEvHWgg5NVyScObHVMPGV8gXpewkb`. Please download all four `vdi` files to your local machine. To run these VM images, you need to use VirtualBox (`https://www.virtualbox.org/`).

2.1 Network Setup

The four VMs are standalone Ubuntu servers and their local network settings (e.g., IP addresses) have already been configured. However, you are responsible for their interconnection and create the topology in Figure 1. The following steps explain the topology setup in VirtualBox.

1. In VirtualBox Manager (i.e., the main GUI panel), create 'New' virtual machine. Use appropriate names (e.g., Attacker, Router) and choose Linux for the type and Ubuntu (64-bit) for the version. When choosing the hard disk option, choose 'Use an existing virtual hard disk file' and select one of the `vdi` files you have downloaded. Then, finally 'create' the VM.

2. After creating the four VMs needed for this assignments, take a look at Figure 1. Notice that Router is connected to all the three VMs and each of their connections is distinguished by the name of their internal network. For example, Attacker and Router is connected through 'intnet-1,' Router and Amp through 'intnet-2,' and Router and Victim through 'intnet-3.'
3. Go to 'Setting' of Attacker VM. Go to 'Network.' You will see that only Adapter 1 is enabled with the NAT option. You need to enable Adapter 2 with 'Internal Network' option and the name 'intnet-1.'
4. Go to 'Setting' of Amp VM. Go to 'Network.' You need to enable Adapter 2 with 'Internal Network' option and the name 'intnet-2.'
5. Go to 'Setting' of Victim VM. Go to 'Network.' You need to enable Adapter 2 with 'Internal Network' option and the name 'intnet-3.'
6. Go to 'Setting' of Router VM. Go to 'Network.' You need to enable three additional adapters: enable Adapter 2 with 'Internal Network' option and the name 'intnet-1,' Adapter 3 with 'Internal Network' option and the name 'intnet-2,' and Adapter 4 with 'Internal Network' option and the name 'intnet-3.'
7. All done. Start all four VMs. Then, they will be automatically form the topology with the assigned IP addresses as shown in Figure 1.

Login credentials. You will need to access the Attacker VM to design and launch your own attacks. You will also need to access the Victim VM to verify the effectiveness of your attack. Use the following login credentials for both servers:

```
username: cs448
password: cs448
```

The Router or Amp VMs have different login credentials and you are not supposed to access them.

2.2 Launching Amplification Attacks

Overview. The goal of the attack is to exploit the amplification (or Amp) server and make it send large traffic volume to the victim server (i.e., Victim). The effectiveness of the attack is measured by the ratio of the attack traffic volume received by Victim to the volume generated by Attacker.

You are asked to write your own attack scripts. You will provide the script and the attack demonstration. You may use some common libraries that come with typical Linux distributions for your attack scripts; however, you are ***not*** supposed to use highly automated tools, such as `scapy`. If you are not sure whether particular libraries are allowed or not, post a question on the github issues or KLMS Q&A.

Amplification Attacks. As discussed in the lecture, the amplification DDoS attack is one of the most widely used DDoS attack vectors for two main reasons: (1) attacker's machine (e.g., Attacker in our setup) is not directly visible to the victims as the amplification servers send attack traffic on be half of the attackers; and (2) attack traffic volume is significantly (e.g., from 10x to 1000x) amplified and thus large-scale attacks can be launched with a low attack cost.

For amplification attacks, several UDP based protocols are often exploited. An adversary first generate UDP request packets with source IP address spoofed with the victim's IP address. The UDP request packets are often carefully crafted to create large response packet from the amplification servers. When the amplification servers receive the spoofed UDP packets, they respond with the response UDP packets and send them to the victim's IP address.

Vulnerability Scanning. Your first task is to find one or more of UDP based services running on the Amp server that can be exploited for amplification attacks. You may use any tool of your choice (e.g., `nmap`) to perform this.

IP Spoofing. As shown in Figure 1, the victim server's IP is 192.168.30.1. Thus, in your UDP request packet, you should spoof the source IP with 192.168.30.1.

Bandwidth Amplification Factor Maximization. The bandwidth amplification factor (BAF) is the important metric for measuring the effectiveness of amplification attacks. BAF is defined by

$$BAF = \frac{\text{len(UDP payload) amplifier to victim}}{\text{len(UDP payload) attacker to amplifier}}.$$

The larger the BAF, the more powerful amplification attacks can be launched. Your task is to craft your UDP request packets that maximize the BAF of your attack. You are free to use any options or parameters for your request packets. If you refer to any external references, you must cite your sources in your attack script.

Attack Demonstration. Now, you are ready to launch your attack and send request packets to the Amp server, which then will send larger response packets to the Victim server. You are ***not*** required to actually flood the Victim server but only to demonstrate that (1) you can make the Amp server to send some attack packets to the Victim server; and (2) the BAF of your attack is measured reliably.

There is a monitoring service on the Amp VM that tracks the attack traffic in the following manner: (1) After the VM is launched (or rebooted), whenever the first request packet for certain protocol (e.g., TCP) and port (e.g., 80) with a spoofed source IP address arrives, a measurement session is launched and terminated after 60 seconds; (2) a bandwidth amplification factor per protocol is calculated for the sum of all the requests and responses when there exist at least 10 requests per protocol during the measurement session.

Note that there will be only one measurement session per VM's launch (or reboot). You can still send packets to Amp VM after the session, but no record will be generated. You may want to have your own monitoring approach for development and testing (e.g., measuring bandwidth at Attacker and Victim). If your attack is ready and a new record is required, you can reboot the Amp VM to reset the measurement sessions.

Once a session is considered ended after 60 seconds, the detail is written to a mongodb server running on the Amp VM, which is essentially a JSON with the following fields:

1. `proto`: The protocol name (either `TCP` or `UDP`)
2. `port`: The port number of certain service on Amp (e.g. 80 for HTTP)
3. `timestamp`: The beginning UNIX timestamp of the session
4. `reqbytes`: The numerator of BAF formula
5. `resbytes`: The denomiator of BAF formula
6. `hmac`: The HMAC tag for submission verification

Retrieving records. The attack session records are saved[1] in the collection `attack_session` of the db `amp` on the Amp VM's mongodb server, which can be accessed by the following account remotely via the default port 27017:

```
username: cs448
password: cs448
```

The account has full write and read access to the mongodb database (db `amp`), allowing you not only reading the records but also cleaning up excessive records generated by experimental run. Taking the mongodb client `mongo` as an example, you can login the db from any VM (e.g. Attacker or Victim VM) using the following command:

```
mongo -u cs448 -p cs448 192.168.20.1:27017/amp
```

Once entering mongo shell, you can use find() command to get all the records:

[1] Note that you may need to wait at least 60 seconds to get the json record until the attack session is considered ended.

```
db.attack_session.find({})
```

Or you can use remove() to remove all of them:

```
db.attack_session.remove({})
```

More commands and parameters of `mongo` shell can be found in its official documentation.

Leaderboard submission. Although it is *not* required for evaluation, you can submit your own attack record json files to a public *leaderboard* to compare your attack effectiveness with other peers. After you get a valid record in json format (with a valid `hmac` field), you can submit it to our leaderboard server running on `172.10.6.117:3000` (via kcloudvpn) for verification and ranking. The read access to the leaderboard requires no login, however submitting a json record to the leaderboard requires authentication. The username of the leaderboard account is the student ID and the password is distributed via KLMS > Grades > Leaderboard Credential.

The submit API (/api/submit) accepts the json record via POST with basic HTTP authentication. An example of submission script using curl command is shown as follows:

```
#!/bin/bash

username=e000001
password=9c390ed3
host=172.10.6.117:3000

url=${username}:${password}@${host}/api/submit

json='{
  "proto" : "TCP",
  "port" : 59013,
  "timestamp" : "1519717537464979376",
  "reqbytes" : 204,
  "resbytes" : 0,
  "hmac" : "69dc5aa04b34ecf9d5efe203d0345c3486a06fa3e4b43f5ee2f5ac1bc757089e"
}'

curl -H "Content-Type: application/json" -X POST -d "${json}" ${url}
```

Once you submit a valid record (i.e. the one with correct `hmac`), the leaderboard will be updated with the submitted BAF for specific protocol and port. You can keep submitting higher BAF records. deadline of this assignment.

Updating your nickname. To offer some level of privacy via anonymity, the leaderboard shows the student records with a nickname, which can be updated via the `rename` API (`/api/rename`). The script for updating your nickname is similar:

```
#!/bin/bash

username=e000001
password=9c390ed3
host=172.10.6.117:3000

url=${username}:${password}@${host}/api/rename

json='{
  "nickname" : "yourname"
}'
```

```
curl -H "Content-Type: application/json" -X POST -d "${json}" ${url}
```

Evaluation.

- You are supposed to find at least two vulnerable UDP services in the Amp server. Showing any BAF value larger than 1.0 in the two protocols guarantee 20 points.
- You are also supposed to demonstrate a large BAF. If you demonstrate a BAF > 100 with any UDP protocol, you earn additional 30 points.
- The last 20 points will be awarded to those who demonstrate larger BAF values and it will graded relatively in comparison with the BAF values of other students. Any exceptionally good BAF may get some extra points!
- **(Important)** Your attack effectiveness will be evaluated based on your direct submission to KLMS (see below). Your submission to the leaderboard will *not* be evaluated. The leaderboard is provided only for fun!

Submission. You need to submit one or more attack scripts named `p2_S_#_seq.py` (or `c`, etc) and one or more attack report files `p2_S_#_seq.json`, where # denotes your student id and `seq` denotes the id of files when submitting two or more files of the same type. Tar the script along with the true-false answer file and upload it to KLMS.