CS5321 Assignment 2: Amplification DDoS Attacks (ver. 1.1)

Due Date: 23:59pm (SGT), 16 March 2018. This is an **individual project**. You MUST finish the implementation and report independently.

Late policy. The cutoff for on-time submission is 23:59pm on the due date. Submitting between 00:00am and 23:59pm on the next day is considered one day late, and so on. You are given 3 "grace days" in this semester which you can use to give yourself extra time without penalty.[1] Once you spend all the grace days, your late submissions will *not* be accepted at all.

1 Amplification DDoS Attacks

In this assignment, you will setup a network topology with the provided virtual machines (VMs) that enables you to launch small-scale amplification DDoS attacks. You need to scan a given server to find vulnerable protocols running on the server. After you launch amplification attacks (see Rossow et al.), you can obtain the report of your attack that includes the evaluation of your attack effectiveness (i.e. amplification factor). By submitting the report to the public leader board, you can compete with other classmates in CS5321.

VM images. You will download the four required VM images from the following link: `https://goo.gl/eWqau1` (NUSNET login required). Please download all four `zip` files to your local machine and unzip them to get the `vdi` files. To run these VM images, you need to use VirtualBox (`https://www.virtualbox.org/`).

1.1 Network Setup

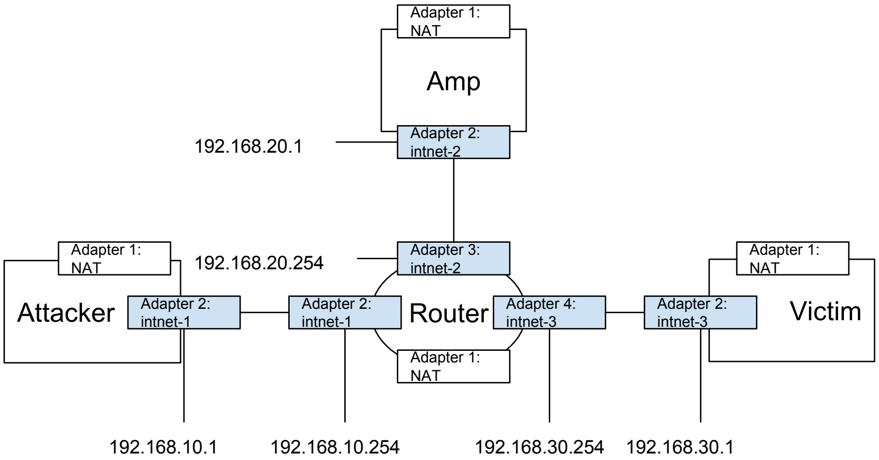

Figure 1: VM setup for amplification attacks. Only the non-NAT network adapters (i.e., blue-shaded parts) need to be manually configured. The rest has been already configured in their `vdi` images.

[1] Grace days are your means to cover real problems: illness, stolen laptops, family problems, etc. It is not intended to be used to cover weekend trips or poor planning, but you can use them as you choose and we don't check up on you. You may use all of your grace days for one assignment, distribute them across several assignments, or not use them at all.

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: cs5321
password: cs5321
```

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

1.2 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. Your submission will be evaluated by the effectiveness of the attacks.

You are asked to write your own attack scripts. You will provide the script and the attack demonstration. You may use some popular libraries (e.g., for IP spoofing) to simplify your attack scripts; however, you are not supposed to use commercial or non-commercial attack tools found on the Internet. If you are not sure whether particular libraries are allowed or not, consult with the TAs.

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 TCP/UDP based protocols are often exploited. An adversary first generate TCP/UDP request packets with source IP address spoofed with the victim's IP address. The request packets are often carefully crafted to create large response packet from the amplification servers. When the amplification servers receive the spoofed packets, they respond with the response packets and send them to the victim's IP address. For more information about amplification DDoS attacks, please refer to the paper by Rossow (2014).[2]

Vulnerability Scanning. Your first task is to find one or more of TCP/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 request packet, you should spoof the source IP with 192.168.30.1. There are multiple ways to spoof the source IP of the outgoing packets and you are free to choose any option.

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

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

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

[2] Rossow, Christian. "Amplification Hell: Revisiting Network Protocols for DDoS Abuse." In *Proc. NDSS*. 2014. Available at `https://dud.inf.tu-dresden.de/~strufe/rn_lit/rossow14amplification.pdf`

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[3] 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: cs5321
password: cs5321
```

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 cs5321 -p cs5321 192.168.20.1:27017/amp
```

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

```
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.

Submiting a record. After you get a valid record in json format (with a valid `hmac` field), you can submit it to our leaderboard server running on `raichu.d2.comp.nus.edu.sg` 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 NUSNET ID and the password is distributed via IVLE Gradebook. You can find the password in the remark entry at IVLE — Gradebook — assignment2-account-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:

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

```
#!/bin/bash

username=e000001
password=9c390ed3
host=raichu.d2.comp.nus.edu.sg

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 by the deadline of this assignment.

Updating your nickname. 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=raichu.d2.comp.nus.edu.sg

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 as many vulnerable TCP/UDP services as possible in the Amp server. Also, you are supposed to demonstrate large BAF. The BAF of your attack will be compared with the BAFs of other students and will be graded relative to the others. Any exceptionally good BAF may get some extra points!

Submission. You need to submit one attack script named `Attack_S_#.{py, java, c, ...}`, where # denotes your NUSNET id. Tar the script (`S_#.tar`) and upload it to IVLE.