CSCI 4430 Tutorial Project 2 (Part I)

Mi Zhang

mzhang@cse.cuhk.edu.hk

(Acknowledgement: Helen Chan)

Outline

- Project Overview
  - Goal
  - Environment
  - Main Ideas
- Prepare the VMs
  - Access to VMs
  - Setup NFQUEUE on VM A
- Header Checksum

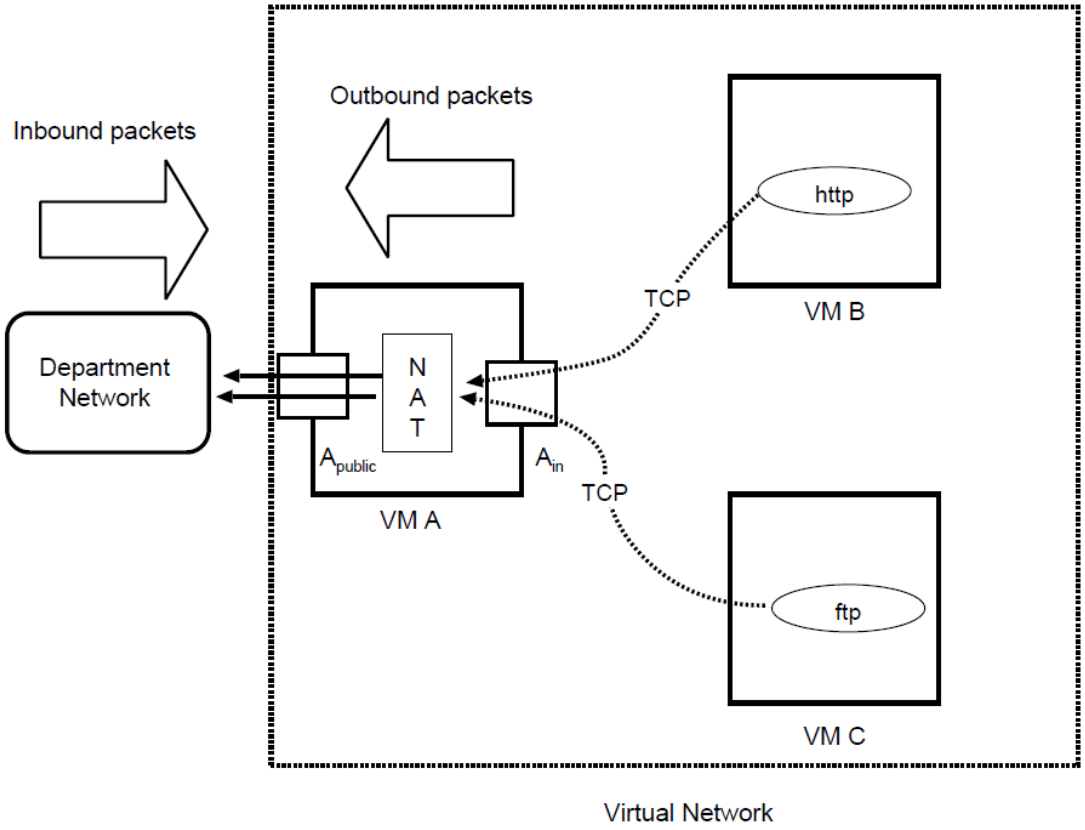
Outline

- Project Overview
  - Goal
  - Environment
  - Main Ideas
- Prepare the VMs
  - Access to VMs
  - Setup NFQUEUE on VM A
- Header Checksum

Goal & Environment

- Goal:
  Build an **NAT program** in C/C++ to forward **TCP** traffic
  - NAT: lecture notes “Network Layer -IP”, pp.60-68
  - TCP: lecture notes, “Transport Layer - TCP”
- Environment:
  - VM A: **2** network interfaces
    - eth0: Virtual, internal network
    - eth1: Department network
  - VM B, VM C: **1** network interface
    - eth0: Virtual, internal network only

Goal & Environment



Goal & Environment

1. Direct all outbound traffic of VM B and VM C to VM A
   - Refer to the specification for setting the default gateway on VM B and VM C - `sudo route add default gw $A_{in}$`
   - **Part of the setup before NAT program is run/tested, NOT to handle it in your program**
2. Trap packets for your NAT program on VM A
   - Refer to Table 1 in the specification - run_iptables.sh
   - **Part of the setup before NAT program is run/tested, NOT to handle it in your program**
3. Run your NAT program on VM A

Main Ideas

- Program input :

```
$ ./nat <public ip> <internal ip> <subnet mask>
```

  - Public IP: `10.3.1.[vm group id]`
  - Internal IP: `10.0.[vm group id].[0-255]`
  - Subnet mask: `24`

- What to do:
  1. Monitor packets on the NFQUEUE
  2. Identify packets matching the criteria, e.g. protocol, inbound vs. outbound
     a. Lookup or create NAT entries if necessary
  3. Modify the packets if necessary
     a. Recalculate checksum
  4. Decide whether to accept or reject packets

Assignment Overview

- **NFQUEUE** (libnetfilter-queue)
  - Online Documentation:
    - http://www.netfilter.org/projects/libnetfilter_queue/doxygen/index.html
  - Installation on Ubuntu
    - sudo apt-get install libnetfilter-queue-dev

Outline

- Assignment Overview
  - Goal
  - Environment
  - Main Ideas
- Prepare the VMs
  - Access to VMs
  - Setup NFQUEUE on VM A
- Header Checksum

Access to VMs

- Email on VMs assignment
  - Contains **VM Group ID**, and other information
  - *Note that "Group No." is not necessary the same as "VM Group ID"*
- Accessible within CSE network
- Power up the VMs via vSphere Client
- Set up your own password
  - Or you risk your homework being accessible by others
- IPs of VMs
  - **`eth0: 10.0.[vm group id].(1|2|3)`**
    - 1 for VM A, 2 for VM B, 3 for VM C
  - **`eth1: 10.3.1.[vm group id]`**
    - Only available on VM A

Access to VMs

- Access via vSphere Client
- Access via SSH
  - VM A: within the CSE network,

    ```
    $ ssh –p[13000 + vm group id] csci4430@projgw.cse.cuhk.edu.hk
    ```

  - VM B, VM C : **on VM A**,

    ```
    $ ssh csci4430@10.0.[vm group id].(2|3)
    ```

Setup NFQUEUE on VM A

- Install NFQUEUE (for development)

```
$ sudo apt-get install libnetfilter-queue-dev
```

- Compile the example program provided in lecture
  - Download nfq.zip from the course webpage
  - Add rules to trap packets, e.g.

```
$ sudo iptables -A OUTPUT -p icmp -j NFQUEUE --queue-num 0
```

  - Run the program: `$ sudo ./nftest`
  - Generate some packets, e.g.

```
$ ping -c 3 10.0.[vm group id].2
```

Same as the queue no. supplied to `nfq_create_queue()`.

Outline

- Assignment Overview
  - Goal
  - Environment
  - Main Ideas
- Prepare the VMs
  - Access to VMs
  - Setup NFQUEUE on VM A
- Header Checksum

Header Checksum

- Recalculate the checksum of a packet whenever the packet is modified
  - Packets with incorrect checksum will be dropped
- How to calculate checksum?
  - Refer to the lecture notes “Transport Layer - Introduction, UDP, Checksum”
- We provide the implementation to generate checksums for **IP** and **TCP** headers:
  - "*`checksum.h`*" and "*`checksum.c`*"
  - You may use them directly in the assignment

Today's Tutorial

- Assignment Overview
  - Goal
  - Environment
  - Main Ideas
- Prepare the VMs
  - Access to VMs
  - Setup NFQUEUE on VM A
- Header Checksum

Next week's tutorial …

- More details on packet processing (using NFQUEUE)
- Some recommended materials to go through before next week's tutorial
  - Example program on NFQUEUE in class
  - Lecture notes on routing and NAT
  - Assignment Specification

Next week tutorial: Project 2 (Part II)

Extra

VM Hostname

- Same hostname for VM A, VM B and VM C …
  - Prompt: `csci4430@csci4430-0:~$`
- Change hostname:
  - On VM A, assume the new hostname is "vm-a":

    ```
    $ echo "vm-a" | sudo tee /etc/hostname
    ```

    - Edit "*/etc/hosts*", replace "csci4430-0" with "vm-a" :

      ```
      127.0.0.1        localhost
      127.0.1.1        vm-a
      ```

    - Re-login to VM
  - Similar on VM B and VM C

(Acknowledgement: Qin Liu)

VM Hostname

- Entering IP every time when log onto VM B and VM C …
- Associate IPs to hostnames
  - Append the entry to "*/etc/hosts*",

```
10.0.[vm group id].1      vm-a
10.0.[vm group id].2      vm-b
10.0.[vm group id].3      vm-c
```

  - May do it on VM B and VM C as well
  - Access VMs by hostname,
    - On VM A, `$ ssh vm-b`

(Acknowledgement: Qin Liu)

Useful Commands

- `ifconfig`
  - Display information of network interfaces
    - For all the network interfaces (by default) `$ ifconfig -a`
    - For a specific interface "eth0", `$ ifconfig eth0`

- `nslookup`
  - Query IP addresses by domain names
    - e.g. find the IP address of workstation `linux1` on VM A or other workstations, `$ nslookup linux1`

```
Name:    linux1.cse.cuhk.edu.hk
Address: 137.189.88.145
```

Useful Commands

- `iptables`
  - User-level tool for maintaining packet filter rules in kernel
    - Display rules in a table "*table*",

      ```
      $ sudo iptables –t [table] –nL
      ```

    - Drop all the rules in a table "*table*",

      ```
      $ sudo iptables –t [table] –F
      ```

- `nc`
  - Netcat: a simple utility for establishing and listening TCP or UDP connections, see man page for details
    - Note: this command is not available on sparc1-sparc20