Network Administration and System Administration

Midterm Examination

Time: 2017/4/24 (Mon.) 09:10 - 12:10

Instructions and Announcements

- 考試時間共三小時，三人一組依照座位分配坐在一起。
- 除了第一題外，所有題目均可在 **R204 的電腦**上或**自己的筆電**上完成，**第一題只能在 R204 電腦上作答**。
- 使用助教提供的 VM image 或 Arch, CentOS 安裝 iso 之前，請自行確認檔案的正確性。可使用 sha1sum 這個指令計算出檔案的 hash 值，並跟官方比對。
  - CentOS 7 的 hash：http://centos.cs.nctu.edu.tw/7/isos/x86_64/sha1sum.txt
  - Arch Linux 的 sha1sum：https://www.archlinux.org/download/
- 為避免發生重大意外，請自行在過程中斟酌是否需要備份 VM。若真的不幸發生悲劇，助教可以提供原始狀態的 VM，但傳輸或下載 VM 需花費一些時間，請盡量避免此一情況。
- 完成題目時可請助教過去評分，如果還沒完成就偷跑先召喚助教，會給予 penalty 2pts。若題目內有若干小題，可分階段完成。有結果可以 demo 但 demo 失敗不在此限。助教有權判斷並決定是否給 penalty。
- 組與組間**禁止討論**，如被發現視為作弊行為，**期中考 0 分**。
- 各題後面黑色星號數目代表我們估計的難度。請參考，可用來決定解題順序。
- 滿分 200 pts。
- Good Luck!!

1 pfSense (25 pts) ★★★★★

Set up a pfSense virtual machine on your R204 desktop computer. The pfSense machine should have 3 VLANs: 7, 22 and 99, such that:

1. pfSense should have an IP issued by the same DHCP server as the host machine assigned to its WAN interface. (6 pts)
2. The DHCP server is only set up to offer IP and respond to DHCP requests in VLAN 99. (3 pts)
3. Only VLAN 99 can access the administration interface of pfSense (e.g., web, SSH). (4 pts)
4. Although machines in VLAN 22 are not directly accessible from 204 machines, every 204 machine is able to ssh "the only" machine in VLAN 22 using `ssh [IP of pfSense's WAN interface]`. (7 pts)
5. Machines in VLAN 7, 22 and 99 are able to access outside machines (e.g., `8.8.8.8`). (3 pts)
6. Same as in HW3, IP addresses should remain unchanged during transmission between VLAN 7, 22 and 99. (2 pts)

You are required to set up **4 virtual machines** (1 pfSense and 3 other whatever-you-liked OS).

Hints

- pfSense should have WAN interface "bridged" to host.
- There is something called "port forwarding".
- Remember to refresh the MAC address of your virtual machines.

Resources

- pfSense ova, password=nasa
  - CSIE
  - Google Drive
- Lubuntu ova, password=nasa
  - CSIE
  - Google Drive

2 Wireshark (25 pts) ★★☆☆☆

Please solve the following questions with the trace file NASA-midterm-wireshark.pcapng.

1. Give the displaying filter to show only the packets associated with DNS queries sent toward any DNS servers here in the CSIE department. (10 pts)
2. Find all MAC addresses from the devices that have initiated a DHCP request but not yet completed the process of obtaining the IP by the end of the trace file. (8 pts)
3. Find all hosts (domains) that were visited by Mac OS clients via HTTP protocol. (7 pts)
    - Hint: user_agent

3 Packet Tracer (25 pts) ★★★☆☆

Hi there, you are our newly hired IT support technician. Your predecessor was pissed off, so he disabled all the freakin' devices we have...

There are two offices, each with a PC, a laptop and a switch, and one of the switches has a terminal that allows net admin to access directly with particular interface. PCs are located on VLAN 7, while laptops are located on VLAN 8.

1. Your first task is to re-establish connections between our internal switches.
   Create trunk between Switch 0 and Switch 1 with LACP.
2. Please reconnect all the PCs and laptops to their respective VLANs.
3. Admins tend to perform sensitive tasks. Therefore, we should place them on an isolated VLAN.
   Place the terminal PC 2 on VLAN 99.
4. Allow the admin terminal to telnet the switch. It's unsafe, but a legit compromise in our scenario.
   Allow PC2 to telnet to Switch 0 through 192.168.99.2/24.
5. Though we use a shitty protocol to establish connections, we still have to use passwords to show some respect to cryptography.
   Use enable secret `q_mao` **and login username** `admin` **with secret password** `admin`**.**

That will be all on your first day of work!

Cisco Packet Tracer

- Username: ciscopt@yopmail.com
- Password: cisco.PT.217

You will need to use this pka file.

sha1sum: 04dc8a97f6a27c73ca7a5baf1deb1f4fdcf2fd8c

4 Strange SSH (15 pts) ★★★☆☆

Ms. Meow wanted to set up public-key-based ssh login on her remote Linux server. She connected to the server using a naive telnet-like communication protocal and issued all the `ssh-keygen` commands you also did in NASA week 2 exercise session. Unfortunately, the communication protocal was unencrypted and a bad guy who sniffed Meow's network collected all the activities between Meow and her remote server. The bad guy hacked Meow's server and prepared a welcome message, for the sole purpose of mockery, for Meow when she tries to log in. You are given the same `pcap` file collected by the bad guy. The task is to collect relevant clues in the `pcap` file and log in to that server as if you were Meow. If you do everything correctly, you will see a welcome message just as Meow did. Demonstrate the message to TAs.

Here are some clues:

- Meow's username is `nasa_meow`.
- Find Meow's private key in the `pcap` file
- Find the private key's passphrase in the `pcap` file
- Find ssh server's listening port in the `pcap` file
- Find the hostname/IP of the remote server in the `pcap` file

Here is the `pcap` file.

5 Stupid encodings (15 pts) ★★☆☆☆

The following string: "`195a30a1d1561cbc0ae7c488b93d037f6b713354`" is the result of applying either `base64` or `base32` for **five** times over the string "`Base{32,64}_Is_Stupid_But_Sometimes_Useful`", followed by a `sha1sum`. In other words, it is the output of the following commands:

```
echo "Base{32,64}_Is_Stupid_But_Sometimes_Useful" | \
    base{32,64} | \
    base{32,64} | \
    base{32,64} | \
    base{32,64} | \
    base{32,64} | \
    sha1sum
```

`base{32,64}` means either `base32` or `base64`.

Your task is to find out the exact sequence that produces such output, using shell script (neither `Python` nor `Perl` is allowed). It suffices to enumerate all 32 possibilities using bruteforce. Print the correct sequence in the following manner:

```
base64 -> base64 -> base32 -> base32 -> base64
```

6 Debian Mirror (15 pts) ★★★☆☆

Debian, the popular Linux distribution, is distributed among hundreds of servers worldwide. These servers are called "mirror", which basically serve binary packages, package list and other miscellaneous components. The mirrors with URL `ftp.<country>.debian.org` are called **primary mirror**. They have high network capacity and are top choices of Debian users in that country. However, only a few countries have Debian primary mirrors. In this problem, you are asked to find out the DNS records and the real providers of these primary mirrors. You are given a country list: country.csv and the following commands:

- Get the provider of an IP address:
  `curl ipinfo.io/$IP 2> /dev/null | grep '"org":'`
- Get DNS records of a hostname:
  `dig +short ftp.[CountryCode].debian.org`

You need to loop through these countries and determine if ftp.$country.debian.org is resolvable; if it is, print it in the following manner:

```
[CountryFullName] ftp.[CountryCode].debian.org:
[IP], provided by [Provider]
[Empty line]
```

If an IP address is not resolvable, just print "unresolvable". For example, in the case of Bosnia & Herzegovina:

```
Bosnia & Herzegovina ftp.ba.debian.org:
mirror.debian.com.ba. => unresolvable
```

A DNS response might contain chaining CNAME records, followed by the final IP address. You must print them in the following manner:

```
[CNAME1] => [CNAME2] => [Final IP], provided by [Provider]
```

For example, in the case of Japan:

```
Japan ftp.jp.debian.org:
cdn.debian.or.jp. => jp.cdn.araki.net. => 133.5.166.3, provided by AS2508 Kyushu University
```

A DNS response might contain multiple A records, you must list them line by line. For example, in the case of the United States:

```
United States ftp.us.debian.org:
64.50.233.100, provided by AS4181 TDS TELECOM
128.30.2.26, provided by AS3 Massachusetts Institute of Technology
128.61.240.89, provided by AS2637 Georgia Institute of Technology
64.50.236.52, provided by AS4181 TDS TELECOM
208.80.154.15, provided by AS14907 Wikimedia Foundation Inc.
```

The desired output is here.

Sidenote: If you are a Debian or Ubuntu user, don't forget to checkout `ftp.tw.debian.org` and its provider. It's an awesome mirror! You might also want to take a look at the official mirror list.

7 Simple Password Generator (15 pts) ★★★☆☆

Write an apg-like shell script. Either POSIX shell scripts or bash shell scripts are acceptable once you have the correct shebang. You should implement it in shell scripts. If you call apg in your script then you will get no point. Your script should support the following options of apg:

- -n num_of_pass
- -m min_pass_len
- -x max_pass_len

Note these options can be given in any order. Every option will only be given once. The default values and behaviors of these options should be the same as those of apg. That is, you should generate *num_of_pass* passwords, with lengths between *min_pass_len* and *max_pass_len*, each printed in a line.
If any other options are given, or the argument after those commands are not integer, exit your script **immediately**, printing some error messages (free format, a simple "error" is okay) to **stderr**, with an exit code **other than 0**.
Your generated passwords should contain only characters from [0-9a-zA-Z]. Instead of implementing some password generating scheme, you can just generate random strings from these characters.

Examples:

```
$ ./my_apg.sh
OPhuwBNG
7oScQs6g
vN5IChFf
jnGzDozu
ixYe5K2b
Kndte5Vr

$ ./my_apg.sh -x 20
mBLiNsvHAzR3ctm54e
GsZbHMaMskqKjb1v
QRQTnqSsa6vpv2nOb
AQTa6VyM1B
irpyaHQKkls
JOxwcCfGxe7zTO0RBYG

$ ./my_apg.sh -m 10 -x 15 -n 2
zyEvDNyPHC
vCZxRuRSKY8TQZ

$ ./my_apg.sh -x 1 -m 10 -n 1
NchYKMeJ61

$ ./my_apg.sh -h
Unknown argument: -h # printed to stderr

$ ./my_apg.sh -x abc
Expected an integer after -x # printed to stderr
```

8 Where are those attackers from? (15 pts) ★☆☆☆☆

Everyday there are many attackers trying to break into CSIE workstations. We have fail2ban installed to prevent them from brute forcing our users' passwords. Now, given the fail2ban log, write a shell script to find out which countries those banned attackers are from.

The ban notice has the following pattern:

```
<timestamp> fail2ban.actions        [<number>]: NOTICE  [<filter name>] Ban <ip>
```

The output should be something like:

```
$ ./from.sh
    348  VN, Vietnam
    273  CN, China
    131  TW, Taiwan
    115  US, United States
     72  AR, Argentina
     56  FR, France
     42  NL, Netherlands
     34  IR, Iran, Islamic Republic of
     27  IN, India
     25  UA, Ukraine
```

Note that:

- Either POSIX shell scripts or bash shell scripts are acceptable once you have the correct shebang.
- Unlike HW1, please print the list of countries **sorted by the frequency in descending order**.
- Each IP should only be counted once when calculating the frequency of each country. That is, if somehow we ban an IP 50 times, it should still be counted once, not 50 times.
- You don't need to check if an IP is unbanned.
- You don't need to match the whole pattern. You can write a script that only works on the input we give you. If you produce the correct output, you will get the points.

9 Yet Another Arch (20 pts) ★★★★★

Install a VM that satisfies the following requirements.

- Can choose to boot into Arch in Grub.
- Can choose to boot into CentOS in Grub.
- Only one LVM volume group is on the VM.
- The LVM volume group should be constructed on two (virtual) disks.
- The two root partitions of the two OS should be based on LVM.
- The above two OS should have the same home directory.
- Your partition for home directory should have an exact size of 87 * 1024 * 1024 bytes.
- You must know how to demonstrate that you satisfy the above requirements.

Hints:

- You can move repositories in Taiwan to higher priority in `/etc/pacman.d/mirrorlist` so you can do `pacstrap` faster.
- https://wiki.archlinux.org/index.php/LVM#Configure_mkinitcpio

Installation ISOs:

- Arch Linux
  - CSIE
  - Google Drive
  - Official Site
  - sha1sum: `71a7aa147877b413497cdff5b1e0aa5bc0c9484f`
- CentOS
  - CSIE
  - Google Drive
  - Official Site
  - sha1sum: `8d7a0370e46bfc7a73cf24696fd8982410932e1f`

10 Enlarge (15 pts) ★★★★☆

You are given a VM, please enlarge its home partition to 2G without adding a new disk.
PLEASE DO NOT REINSTALL IT!!

- VM Image
  - CSIE
  - Google Drive
  - sha1sum: `3d50813f8b1706082f04cdf699a225e69a5e83a0`
- Username: root
- Password: root

11 Virtualize (15 pts) ★★★★☆

Simply install KVM on CentOS as you did in the homework. It should satisfy the following requirements:

- With virt-manager, non-root user can create VM without root permission of the VM host.
- Support hardware virtualization.
- The ssh key of the client should be password-protected (when generated). Note that this is not telling you to manually encrypt it with things like openssl, tar.
- When you demonstrate, the window prompting for asking password should not appear. That is, you should know how to "cache" the password while keeping the private key encrypted.

And also, you should know how to demonstrate that these requirements are satisfied. Please demonstrate as you did in the lab class. That is, there are two VMs. One is the VM host and the other one is the client with GUI.

Note that we strongly recommend you to solve this problem with VMWare Workstation Player.

You need to unzip those files before you can open them with VMWare Workstation Player. You can ask TAs how to import them.

VMs:

- Clean CentOS VM without GUI
  - CSIE
  - Google Drive
  - sha1sum: `94a277cf9c98df1505adeb151a3efa6206d37899`
  - Username: `root`, Password: `root`
- CentOS VM with GUI
  - CSIE
  - Google Drive
  - sha1sum: `0278d4bf2ed2f10289f7a1d4de559ef3a8a088a2`
  - Username: `root`, Password: `root`
  - Username: `user`, Password: `user`