Network Administration and System Administration

Final Examination

Time: 2023/06/05 09:10 - 12:10

Instructions and Announcements

- 考試時間共三小時，三人一組考試。分組連結為 https://docs.google.com/spreadsheets/d/1EA5O_59-jkJAjzi_sTBCrJ7w6xodPEhGwv_e3NhxFVU/edit?usp=sharing。
- 考試期間禁止使用手機、電話、任何通訊軟體等與同組成員外任何人聯繫，也禁止組與組之間**一切討論與合作**，如被發現視為作弊行為，**期末考 0 分**，並依校規懲處。
- 本考試允許使用 ChatGPT。
- 作答過程中請自行斟酌備份，避免電腦發生意外，損失過多進度，可考慮準備隨身碟或雲端空間備份進度。
- 為避免發生重大意外，請自行注意 VM 用量，同時開啟過多 VM 可能導致電腦當機，我們恕不負責。
- 線上考試的 announcement 將會更新在 https://docs.google.com/document/d/1jwSibeldxdNvs94KwXKQ0EzjMjNIIS770AJr2hKL4kU/edit?usp=sharing。
- 題目檔案請至 https://drive.google.com/drive/folders/1zy36ceP_cvVjE7jryGnrx0vPV6uIdSzc?usp=sharing (考試開始後公開) 下載。
- 完成題目時請至 https://forms.gle/7gkvqDtAcnC2k9LZ9 上傳作答內容，每組每個 subtask **最多上傳 3 次**。
- 計分板連結 https://docs.google.com/spreadsheets/d/1zTNZSnieYxQRFbbDr2qUsalcqr7UKlip4aA-9jG4Lr0/edit?usp=sharing。
- 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。
- 題目可能難免有疏誤之處。若發現有解不開題目、題敘不清的狀況，請盡快跟助教或老師反應，或斟酌時間先解別的題目。
- 滿分為 **230** pts。

1 HTTPS ★★ ∼ ★★★★ (30 points)

Resources

- server.ova (Server VM) (可以用 GUI 登入，account/password: Guest/nasanasa)
- client.ova (Client VM) (可以用 GUI 登入，account/password: Guest/nasanasa)
- alpine-standard-3.17.3-x86_64.iso
- OPNsense-23.1-OpenSSL-dvd-amd64.iso.bz2
- server_sample.py (用 chatGPT 寫的 python HTTPS server 範例)
- client_sample.py (用 chatGPT 寫的 python HTTPS client 範例)

Note: ova 檔測試過可在 204 電腦 + Windows + VirtualBox 上運作。

Description

還記得作業五的 AnJ 與他破爛的 HTTPS server 嗎？AnJ 自從作業五發現大家都把他的秘密看光光之後很害羞，決定重新架一個新的服務。可惜他死性不改使用的仍是 self-signed certificate，而跟一般情形一樣，client 端也都不進行確認。於是，你決定要來教教他什麼是中間人攻擊。

在這個題目裡，你需要實作一個 MitM (Man-in-the-middle) attack，將 server VM （以下簡稱 server）與 client VM（以下簡稱 client）放到同一個網路之後去研究它們之間的連線。

Note: 你可以使用任何方式把 client.ova、server.ova 放到同一個網路底下，但請注意：

1. server 的 IP 需設為 192.168.1.101。
2. client 會嘗試連線 server.lisa.anj.terrance ，所以請依據各小題的需求設定 DNS record。
3. server 跟 client 會用 crontab 每一分鐘去嘗試連線。server 會把連線的紀錄寫在 /tmp/server.log；client 會把連線的紀錄寫在 /tmp/connection.log。
4. 本題是 CTF 的形式，各小題都會有一個格式為 nasa{...} 的訊息（以下稱為 flag），請找到 flag 來作為達成該小題的依據。

Tasks

(1) (5%) flag1: 請把網路架起來，讓 server VM 跟 client VM 在同一個網路底下（沒有特別的 LAN、VLAN 條件限制，只要兩台機器間能連到對方就好了）。Server VM 的 IP 為 192.168.1.101，而 server.lisa.anj.terrance 要對應到 server VM 的 IP 192.168.1.101。架好之後到 /tmp/connection.log 跟 /tmp/server.log 檢查一下，就可以找到格式為 nasa{...} 的訊息了喔～

Hint: 推薦使用 OPNSense 。但如果想要使用 pfsense、iptable 搭配自己架設 DNS 服務也沒有問題。

(2) (5%) flag2: 請架一台 Eve VM ，放在同一個網路底下，然後對 server 進行連線。網路架設正確時，server 的服務會自動開在 192.168.1.101 的 port 2037。請發一個簡單的 GET request 看看 server 會回覆你什麼？

Hint: 建議 Eve VM 可以用 alpine linux 作為 OS（比較輕量、電腦比較不會因為虛擬機過多當機，但記得要 apk install 需要的東西），然後使用 python 寫簡單的 https client（可參考範例的 python 檔案）。但要用其他方式來做 https client 也都可以。

(3) (10%) flag3: 在 Eve VM 架一個簡單的 HTTPS 服務，來接受 client 的資料。把內網底下 server.lisa.anj.terrance 的 DNS record 竄改成你 Eve VM 的 IP，看看 client 想跟 https://server.lisa.anj.terrance:2037 說什麼？

Hint: 建議 Eve VM 可以用 alpine linux 作為 OS（比較輕量、電腦比較不會因為虛擬機過多當機，但記得要 apk install 需要的東西），然後使用 python 寫簡單的 https server（可參考範例的 python 檔案）。但要用其他方式來做 https server 也都可以。

(4) (10%) flag4: 在 Eve 上實作：先讓 client 對你進行連線之後，把 client 的資料轉傳給 server，讓你神不知鬼不覺的成為那個中間人。成功的話你就會在 client 跟 server 的訊息間看到 flag4。

Submission

請透過 Google Forms 回答 flag（格式：nasa{...}），各小題可以分開上傳、也不強迫解題順序，但請注意不要傳錯題。

2 ID: nasa2023 ★ ∼ ★★★★★ (35 points)

Resources

- `bot.sh`: Task 3 中的 BOT。
  https://drive.google.com/drive/u/2/folders/1_w5tjukRGbD1nAXMfYXCx3WFoFkSGnem

Rules

- 請不要協助或干擾其他組別作答，違者將視情節嚴重程度進行懲處。
- 在 3 個 Task 中，皆不需要取得 `nasa2023*` 的帳號權限。
- Flag 的形式為 `NASA{[printable ascii characters]}`。

Description

西元 2203 年 6 月，一個名為「NASA 期末考」的生存遊戲襲捲台大資工系。身在其中的你，意外得知一個特殊的帳號 `nasa2023`，只要成功登入帳號，就能得到 `nasa2023` 留下來的特殊寶物（flag），換取更多分數，增加通關的機會。

Task 1 - PHP Login? (1) ★ (15 points)

請連線至 https://www.csie.ntu.edu.tw/~nasa2023/login1.php 並嘗試以 `nasa2023` 的身份登入。成功登入後網頁會顯示 flag 的內容。

Hint: 有沒有什麼方法能看到 PHP 的原始碼呢？

Task 2 - PHP Login? (2) ★★ ∼ ★★★ (11 points)

請連線至 https://www.csie.ntu.edu.tw/~nasa2023/login2.php 並嘗試以 `nasa2023` 的身份登入。成功登入後網頁會顯示 flag 的內容。

- Hint: 這題和 Task 1 有什麼不同？PHP 是由哪個使用者執行？
- Hint: https://www.php.net/manual/zh/function.exec.php 可能對解題過程有幫助。

Task 3 - Linux Login? ★★★★★ (9 points)

以下是你可能會在登入工作站時看到的畫面：

```
############################################################
#         Public Domain Workstation Lab (R217).            #
############################################################
#   UNIX Login Service:                                    #
#                        ...                               #
#                                                          #
############################ Last Update: Mar  30 2023 ###
Last login: Wed May 31 10:49:42 2023 from 127.0.0.1
mail: /var/spool/mail/nasa2023: No such entry, file or directory
```

西元 2023 年 05 月 31 日 (週三) 10 時 50 分 14 秒 CST

為了確認你有能力登入 nasa2023 的帳號，nasa2023 留了一個 BOT (見附檔) 來擷取 `Last login` 的資訊。如同 `Description` 所述，這是一場在 2203 年 6 月的遊戲，因此 BOT 要求 `Last login` 行中要有 2203 和 Jun，但是，怎麼可能有辦法從未來登入呢？

請在 `linux1~linux10` 中選擇一台機器作答，並嘗試讓 `ssh nasa2023t${teamID}` 的輸出資料通過 BOT 的驗證，其中 `teamID` 是你的組別編號（執行範例： `bash bot.sh nasa2023t01 linux1`）。

- Hint: 不需要真的能登入 `nasa2023t${teamID}`。
- Hint: 作答次數有限，建議先使用自己的帳號嘗試。

Submission

Task 1/2: 請透過 Google Forms 回答 flag。
Task 3: 請透過 Google Forms 回答使用的機器（`linux1~linux10`），助教會執行附檔的 `bot.sh` 來檢查。

3 DNS ★ ∼ ★★ (20 points)

Reminder

- 請使用 nasa2023-final-dns.ova 來作為 primary 及 secondary server 來完成此題，user 和 password 皆為 nasa2023。
- 為節省同學們的時間，不用另外安裝 client 的 VM，在測試 primary 及 secondary server 時，我們會直接使用 dig @localhost
- 每一題都需要以下的設定：
  - zone : group[GroupID].com (for example: group 30's zone = group30.com)
    - nasa2023.group[GroupID].com -> 1.2.3.[GroupID]
    - www.nasa2023.group[GroupID].com -> nasa2023.group[GroupID].com
  - dig @localhost -x 1.2.3.[GroupID] 要可以得到 nasa2023.group[GroupID].com

Task 1 - Primary/Secondary (10 points)

許多 server 都會有 primary 跟 secondary 的機制，這樣的好處是假設有任一 server 出現問題而停止運作，也還是有其他 server 能提供相同的服務。Bind 很貼心的幫我們解決了這樣的需求，只要我們正確設置好參數，primary server 就能定時將指定的 zone file 備份到 secondary server！

- 請利用本題的 ova 檔再開啟一個 VM 來當作 secondary，而我們已經架設完成的 server 則作為 primary，設定好參數，讓 primary 上創建好的 zone file 可以自動備份到 secondary server 上
- 成功設定後，不需要新增 zone file 即可在 secondary server 自動備份 zone file
- 除此之外，此時在 secondary 透過

  ```
  dig @localhost nasa2023.group[GroupID].com
  dig @localhost www.nasa2023.group[GroupID].com
  dig @localhost -x 1.2.3.[GroupID]
  ```

  等等指令應該能正確查詢 record。

Task 2 - DNSSEC (5 Points)

NASA TAs 在批改作業時，發現 DNSSEC 頻繁出現於答案中，在知道它是甚麼後，現在我們來實際操作，讓同學更了解 DNSSEC 背後的運作。
請完成以下要求：

- 請使用 Task 1 所設定的 master 和 slave，為它們設置 DNSSEC
- Sign "group[GroupID].com" 這個 zone
- 成功設定後，不管在 primary 還是 secondary 上用

  ```
  dig @localhost nasa2023.group[GroupID].com
  dig @localhost www.nasa2023.group[GroupID].com
  dig @localhost -x 1.2.3.[GroupID]
  dig DNSKEY @localhost group[GroupID].com
  dig +dnssec @localhost nasa2023.group[GroupID].com
  ```

  等等指令應能顯示正確 output。

Task 3 - DNS over HTTPS (5 Points)

為了讓使用者的 DNS query 不會被網路上的其他人看到，你決定為你的 server 設定 DNS over HTTPS。
請完成以下要求：

- 請設定 primary DNS server 在 port 443 上接受 DNS over HTTPS 的 query
- 如果你使用的憑證是 `server.crt`，用

  ```
  dig @localhost nasa2023.group[GroupID].com
  dig @localhost www.nasa2023.group[GroupID].com
  dig @localhost -x 1.2.3.[GroupID]
  dig +https +tls-ca=server.crt @localhost nasa2023.group[GroupID].com
  dig +https +tls-ca=server.crt @localhost www.nasa2023.group[GroupID].com
  dig +https +tls-ca=server.crt @localhost -x 1.2.3.[GroupID]
  ```

  等等指令應該能正確查詢 record。

Submission

- 請找助教 demo

4 (Cisco Switch) Bocchi's バイト総集編 ★ ∼ ★★ (20 points)

為了慶祝孤獨搖滾宣布製作總集篇劇場版（ ^ ∨ ^ ），我們推出 **Bocchi's バイト総集編**，在這一題裡你將會複習到之前所用到的指令以及一些新的東東？

Resources

- `soushuuhen.pka`
- Packet Tracer 8.2.0 安裝檔
- Packet Tracer account: cisco.packet.tracer@yopmail.com
- Packet Tracer password: Cisco.packet.tracer0

Task - `soushuuhen.pka` (20 points)

- 目標：
  1. (5 points ★★) 系上剛收購了一台二手的 Cisco Switch，連上去發現上面還留有前一個使用者的舊設定，請重設 Switch 的所有設定。
  2. (5 points ★) 請在 1F-A 上新增一個使用者 (Username: Bocchi、Secret: Bocchi)，並在透過 console 連接 1F-A 時，驗證使用者身分。(不需要處理 vty 的部分)
  3. (5 points ★) 請在 1F-A 上設定 PC10-X 屬於 VLAN10，PC20-X 屬於 VLAN20。
  4. (5 points ★★) 請在 1F-A 上設定 NA-Server 屬於 VLAN99 後，將 running-config 存入 startup-config，並透過 TFTP 存到 NA-Server 上，檔名為 1F-A.2023-06-05。(NA-Server 上的 TFTP 服務已經開啟了)
- 注意：
  - 沒有 Activity Check，需自行檢查設定結果。
  - 過程中如需設定 1F-A 的 Interface IP 可自行設定。
  - 4 個小題各自獨立，完成該小題的要求即可得到分數。

Submission

- 請上傳完成後的 `.pka` 檔

5 RADIUS in Wireless Network ★ ∼ ★★ (20 points)

Description

在 lab 中，我們簡單介紹過系上的 WiFi 網路架構，其中提到 RADIUS 是負責進行驗證的重要服務。以下會有 3 大題與 RADIUS 有關的題目，請根據題目的要求回答問題。

Task 1 - RADIUS 問答 (6 points)

簡答題，請根據題目回答以下問題。

1. (1 point) RADIUS 主要提供的服務為 AAA，請問 AAA 分別是什麼？
2. (1 point) RADIUS 伺服器主要使用的 port 有兩個，請找出這兩個 port 的預設值，並分別回答他們負責的功能及使用的傳輸層 (transport layer) 預設協定。
3. (2 points) RADIUS 伺服器的客戶端 (clients) 通常是 NAS (Network Access Server)，請簡單說明什麼是 NAS(最多三句話)，並回答在資工系的無線網路架構中，作為 RADIUS 伺服器的客戶端的 NAS 設備為何？**注意這題的 NAS 並不是指 Network Attached Storage！**
4. (2 points) 大家平常在校園內應該可以看到名為 eduroam 的 SSID，eduroam 讓不同學校的 WiFi 使用者 (例如訪客等) 能透過同一組帳號密碼在所有提供 eduroam 服務的校園內都能驗證並登入使用無線網路。eduroam 是透過 proxy RADIUS 提供服務的，請簡單說明什麼是 proxy RADIUS，並簡單描述 eduroam 是怎麼使用 proxy RADIUS 以達到跨校漫遊的 (總共最多五句話)。

Task 2 - RADIUS 設定檔 (6 points)

我們系上所使用的 RADIUS 伺服器是 FreeRADIUS，必須透過正確的設定才能讓服務正常運作。我們會提供某台 FreeRADIUS 伺服器的設定檔，請根據以下題目要求從設定檔中找到對應的答案。

Resource

- radius_conf.zip，這個壓縮檔裡面包含某一台 FreeRADIUS 伺服器的完整設定檔，這些設定檔案在這台 FreeRADIUS 中原本的路徑為 /etc/freeradius/3.0；另外，伺服器的 FreeRADIUS 版本是 3.0.20，作業系統為 Ubuntu 22.04

Problems

1. (3 points) 使用 RADIUS 驗證時，必須到儲存使用者帳號密碼的資料庫確認正確性。已知在本題中的這台 FreeRADIUS 驗證時使用的是 SQL 類的資料庫，請從提供的設定檔中找到答案並回答問題。
   (a) (1.5 points) 請問使用的資料庫是什麼？(請寫軟體名稱)
   (b) (1.5 points) 用來登入這個資料庫軟體的帳號密碼分別是什麼？
2. (3 points) Figure 1 是某一台 AP 的 RADIUS 驗證設定畫面，我們想要讓這台 AP 能透過我們的 FreeRADIUS 伺服器進行驗證。已知這台 AP 與 FreeRADIUS 為不同機器，且 FreeRADIUS 端的設定已經完成。請從提供的設定檔中找到答案並回答問題。
   (a) (1.5 points) 請問 Figure 1 中的第三個欄位 (連線密碼)，應該要填入什麼字串？
   (b) (1.5 points) 這台 AP 的 IP 位址可能是什麼？請給出一個範圍，例如：172.16.0.0 ∼ 172.16.255.255。

Hint

- 這題中的兩個小題各自都只需要查看一個設定檔案，因此可以先試著找到哪一個設定檔是跟題目相關的，再從中找答案。

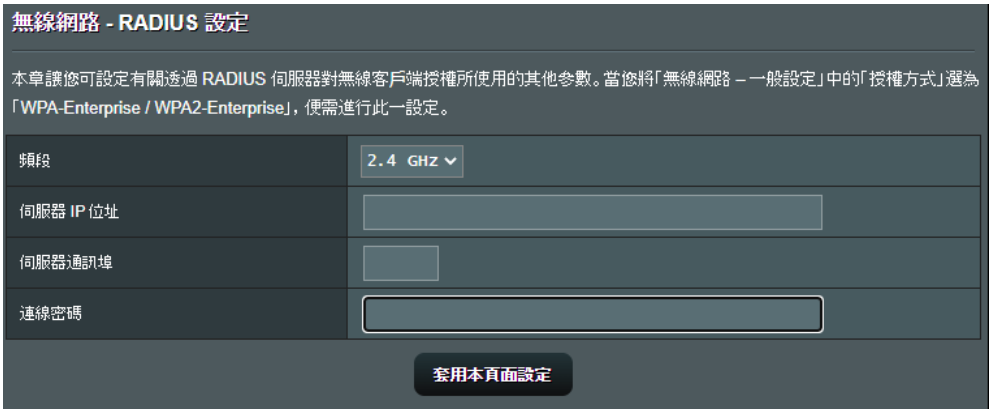


Figure 1: AP RADIUS Setting

Task 3 - RADIUS log **搜尋** (8 points)

Resource

- radius_log.json

Description

在 NASA 的日常工作中，常常會需要看 log 來了解並解決突發狀況，因此如何整理並找出 log 有問題的地方是很重要的課題。附檔是系上 RADIUS 伺服器的 log 的一部分，並且經過特別處理以保護隱私。

對於個別欄位的說明：

- InputPackets: 使用者下載的封包數
- OutputPackets: 使用者上傳的封包數
- InputPackets: 使用者下載的 byte 數
- OutputOctets: 使用者上傳的 byte 數
- UserEndRSSI: 該使用者最後觀測到的 RSSI 值，單位是 AP 廠商訂定的分數

1. (3 points) 請找出 EndRSSI 最大的使用者以及 RSSI 值。
2. (2 points) 承上題，假設 RSSI 減去 100 就可以換算成 dBm，並且環境的 noise 是 −50 dBm，請算出當時的 SNR 值。
3. (3 points) 請分別找出 input packets 和 output packets 最大的使用者 (**注意同樣的使用者可能出現在多筆記錄中！**)

Submission

- 所有題目皆請透過 Google Forms 上傳文字回答。所有小題皆可以分開上傳 (包含 Task 2 的 a, b 小題)，並且次數分開計算。

6 Raise Cats in Web Server ★ ∼ ★★ (35 points)

Background

還記得你們為了進到 NASA 寫的第一份作業 HW0 嗎？那時你被 SA 成員們爆打一頓，昏倒在地，雖然你好不容易進來了，也成功熬過了一學期（為你自己喝采吧！），但優秀的你遭受一個學期的荼毒與折磨，想必你已經對這群人心生恐懼。但突然有一天，你偶然發現他們養了好幾隻貓，大家都會看著貓貓，歪頭燦笑，你以為大家是少女心萌發，結果深入調查才發現，這群人把貓貓囚禁在不同的 web 世界裡，並且強迫他們勞動.....

本題基本上是改自 Lab 12 與作業 6，希望大家能好好把握之前所學來答題拿分數！

Before you start

- 請利用 P6_cats_server.ova 製作一個新的 VM，用來操作以回答接下來的題目。
- VM Account
  - Distribution: Debian 11.6.0
  - Default username & password: nasa2023
  - nasa2023 具有 sudo 權限
- 繳交時請勾選要繳交的題目，第 4 小題會由助教過去看 demo，其他題目在表單內檢附上截圖即可，如果是瀏覽器的畫面一定要包含完整清晰的網址、終端機的畫面要包含你下的指令，否則不予計分。

Tasks

1. Basic Setup (3%)
   - 請確認虛擬化的網路設定確保你的本機能夠連得到貓貓們工作的 VM
   - 請安裝 Nginx server 所需的相關套件，並啟動 Nginx Service。
   - VM 內有開啟 ufw 防火牆，請修改 rules 使你想要的服務能正常運作 (e.g. sshd / nginx service)。
   - 請附上連線至 `http://{your_vm_ip}` 的畫面截圖，應該要會出現顯示 Welcome to nginx! 的預設畫面。
2. Access Log (3%)
   - 為了紀錄貓貓們的工作狀況（Server 的連線狀況），請將 Nginx Server 的 access log 建立並保存在以下的路徑中：/var/log/nginx/catracking.log 。
   - 請附上執行

     ```
     cat /var/log/nginx/catracking.log
     ```

     的畫面截圖。如果 log 是空白的請多連線你的網頁幾次。
3. 403 Error (5%)
   - 你想在 Server 上面存放一些秘密資料。請建立 /var/www/html/secrets 資料夾，如果有任何人不小心連到這個 directory，請使用下圖的貓貓來回覆 403 Error response（可參見 `https://http.cat/403`）。

- 請附上連線至 http://{your_vm_ip}/secrets 的畫面截圖，網頁無需排版，有出現貓貓讓我們看到它有正常工作即可。

4. User Directory & Redirect (5%)
   - 請設定你的 Nginx Server，當連上 http://{your_vm_ip}/~{path} 時，可以被 redirect 到https://http.cat/{path}
   - path 是任何非空的字串
   - 請附上在終端機執行 curl -v http://{your_vm_ip}/~500 的畫面截圖。
5. PHP on Server (5%)
   - 讓 my-website.ntu.edu.tw 這個 domain name 指到你的 VM。請於本機設定 hostname 跟 ip 的配對檔案（像在 linux 中是 /etc/hosts)，使得該 domain name 會對應到 VM 的 IP address。(記得考試結束要改回來喔 !)
   - 請寫一個簡單的 PHP cat.php 程式到你的 Server 上。當連上 http://my-website.ntu.edu.tw/cat.php 時，可以顯示"Help us MEOW..." 字樣。
   - PHP 內容能顯示字樣即可，如果你要額外發揮創意我們也很歡迎。
   - 請附上連線至 http://my-website.ntu.edu.tw/cat.php 的畫面截圖。
6. Reverse Proxy (7%)
   - 作業已經讓你有能力在同一個 Nginx Server 上面設定多個網站，用不同的網域名稱連上時會有不同的內容，我們再來複習一次吧。
   - 用前一小題的做法，讓 other-website.ntu.edu.tw 這個 domain name 也指到你的 VM。
   - 在前一小題的結果存在的情況下，當連上 http://other-website.ntu.edu.tw/nasa.php ，可以顯示" NASA harms cats..." 字樣。
   - 請創建新的 dir (e.g. /var/www/other_html) 存放另一個網站的內容（php file)。PHP 內容能顯示字樣即可。
   - 請附上連線至 http://other-website.ntu.edu.tw/nasa.php 的畫面截圖，前一小題尚未完成者，此題亦不計分。
7. N Web Servers?! (7%)
   - 你知道同一台機器上可以同時存在 N 個 web servers 嗎？也就是說，貓貓們要不斷把 request 傳來傳去，被更加剝削！
   - 在實務中，不同的 web server 開發上會有不同的考量面向，例如 Nginx 強大在速度、Apache 強大在功能的齊全，因此有些人會用 Nginx 接收 request，簡單的網站就直接回傳靜態網頁、複雜的則轉交給 Apache 處理；甚至不只功能上的考量，在「同一台」實體機上，因為

網頁的開發語言不同，我們可能要讓 Nginx 把 request 轉傳給另一個 server 處理，比方說這個網站的後端是用 JavaScript 開發的，可能就要轉給 Nodejs server，因此這題要來讓你嘗試簡單地建置雙 server！

- 請你另外安裝 Apache，將它開在 8080 port，並且讓 Nginx 在看到 `http://{your_vm_ip}/apache` 時導向 Apache server，它應該要能顯示 Apache 預設的歡迎畫面。
- 請附上連線至 `http://{your_vm_ip}/apache` 的畫面截圖。
- Hint: 如果你連上去出現的是 Apache 顯示的 404 畫面，代表你超級接近了，也許可以再多想想 Nginx redirect 時多加什麼資訊可以讓 Apache 知道你要的是 root index page (/)？

7 Can't get my system right ★ ~ ★★★ (20 points)

Resource

- VM 檔案：`ldap.ova`, `workstation.ova`

Environment

此題所有使用者、LDAP 密碼均為 `nasa2023`

LDAP Server

- 作業系統：Debian 11 (bullseye)
- 介面卡 1: NAT
- 介面卡 2: Internal Network (10.217.44.21)
- 使用者名稱：`root`
- LDAP admin dn: `cn=admin,dc=nasa,dc=csie,dc=ntu`
- LDAP 使用者: `uid=ldap-user,ou=people,dc=nasa,dc=csie,dc=ntu`

Workstation

- 作業系統：Arch Linux
- 介面卡 1: NAT
- 介面卡 2: Internal Network (10.217.44.10)
- 使用者名稱：`root`

7.1 LDAP Migration ★★ (10 points)

原先的 LDAP server (未附上) 位於 10.217.44.20。由於作業系統過舊，NASA 成員們決定用 Debian 11 安裝新的 server, 並設定 IP 為 10.217.44.21。不過，工作站上的 SSSD 似乎沒辦法與新 server 連線。請找出無法連線的理由，並修復它。在 Demo 時，請簡短說明你改變了哪些設定。助教會將 Workstation VM 重新開機，並嘗試用 ldap-user 帳號登入。

7.2 Too Many Logins ★ (5 points)

在檢查 journal 時，你發現有奇怪的 IP 似乎正在暴搜機器上的使用者。這個動作在機器上 journal 上製造了非常多無用的 log。請將兩天以前的 log 全數刪除，並設定 journald 使得這些 log 最多只會佔用 16MB 的大小。

7.3 No notes, no curve ★ (5 points)

在工作站上執行 `pacman -S python-matplotlib` 時會產生大量錯誤。請找出原因並安裝此 package。

Submission

請找助教 demo。

8 Secure VNC ★ ∼ ★★★ (20 points)

Resource

- VM 檔案：`vnc.ova`
- 指定的 VNC server 和 client：TigerVNC

Environment

- 作業系統：Debian 11 (bullseye)
- VM 網路介面：bridged
- 使用者名稱：`anna`
- 使用者密碼：`nasa2023`
- 為確保運作順暢，請使用 R204 電腦 Windows 10 的 Virtual Box 或 VMWare Workstation 完成作業。我們不保證能順利運行，也不會提供在其他環境運行的幫助。
- 在各小題的規範下，你可以在 VM 安裝任何必要或喜歡的套件和軟體。

Task 8.1 ★ (5 points)

如果你有用過工作站開 VM 寫作業的話，你可能知道 VNC 是一個可以用來做到遠端桌面的系統。本題請幫助 Anna 在提供的 VM 中啟動一個 VNC Server，並用 SSH port forwarding 來達到 VNC over ssh tunnel 的效果。

- 需求
  - 在 VM 中啟動一個 VNC server (command: `tigervncserver`)
  - VNC server **只在** `localhost:5999` 接受連線
  - 客戶端要以密碼驗證身分
  - 客戶端連上後**應該執行 firefox，而不是完整的桌面環境**
  - 從 host 端（e.g. 204 電腦）透過 SSH port forwarding 成功連上 VNC server

- 提示
  - VM 中已經安裝並啟用 SSH server
  - 你需要安裝的套件是 `tigervnc-standalone-server` 和 `firefox-esr`

Task 8.2 ★★ (5 points)

Anna 發現前一小題的設置雖然可以做到安全的遠端桌面，但因為必須連到 VM 的 ssh server，所以如果離開 VM 所在的 LAN 就沒辦法連上了。請幫忙 Anna 設置 VNC server，使得 Anna 就算在不同 LAN 也能連到 VNC server。

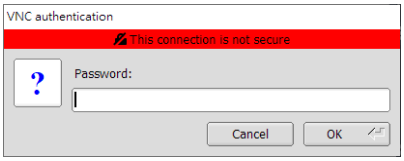
- 需求
  - 在 VM 中啟動一個 VNC server
  - VNC server **只在** `localhost:5999` 接受連線
  - 客戶端要以密碼驗證身分
  - 客戶端連上後**應該執行 firefox，而不是完整的桌面環境**
  - 選擇任意一台 CSIE Workstation 和一個非 5999 的 port `X`，使得往該台工作站 `localhost:X` 的連線被轉送到 VM 的 `localhost:5999`
  - 使用類似前一小題的指令 SSH port forward 到你選擇的工作站，並連上 VNC server。

Task 8.3 ★★★ (10 points)

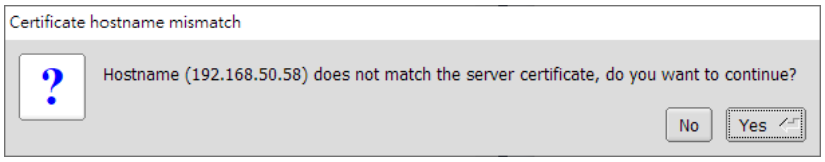
Anna 在試用 VNC 一段時間後很滿意，因此決定將 VNC 部署到一台有對外 IP 的機器。請幫 Anna 產生一組自簽憑證並用於 VNC server，避免壞人假冒 Anna 的機器。

- 需求
  - 在 VM 中啟動一個 VNC server
  - VNC server 在**所有網路介面**的 port 5999 接受連線
  - VNC server 要以 TLS 加密傳輸的資料，並搭配自簽的 X509 憑證
  - **憑證請選用 RSA-4096 演算法，並設定在 30 天後就到期**
  - **在 `/.vnc/tigervnc.conf` 寫入正確內容，讓 Anna 不需要在指令裡面指定憑證和私鑰的檔案路徑**
  - 客戶端要以密碼驗證身分
  - 客戶端連上後**應該執行 firefox，而不是完整的桌面環境**
  - 連上 VNC server 要輸入密碼時，應該會有”This connection is secure” 的字樣，且不會跳出憑證 hostname 相關的警告。”Unknown certificate issuer” 的警告則是允許的。
- 提示
  - 設定成功時，輸入密碼的介面應該是：

而不是：



– 像是這樣的警告是不允許的：



Submission

請找助教 Demo。

9 Shell me the CorGI ★ ∼ ★★ + 0.5★ (30 points)

Resource

- web-scripting-release.zip
- 內含：
  - Dockerfile
  - data 資料夾，供 Task 9.3 測試用
- Build 執行 Script 用的 Docker 環境：
  - `docker build -t shell-cgi - < Dockerfile`
- 期末好運貓貓

Description

你知道用 Shell Script 也可以寫網頁後端嗎！下方是一個用 Bash 寫的簡單範例：

```
#!/bin/bash
printf "Content-type: text/plain\n\nMeow!"
```

你可以將其存於工作站家目錄下的 `~/htdocs/cgi-bin/test`，將其設為可執行，並打開瀏覽器連到 `https://www.csie.ntu.edu.tw/~<username>/cgi-bin/test`，就會看到一個可愛的 `Meow!` 了！

透過「通用閘道器介面（Common Gateway Interface, CGI）[1]」，任何能夠輸入、輸出、以及讀取環境變數的程式語言基本上都能用來撰寫網頁後端。在這個大題中，你要使用 Shell Script 來撰寫一些簡單的 CGI 網頁應用。

[1] https://en.wikipedia.org/wiki/Common_Gateway_Interface

Environment

我們會在上述提到的 Docker 環境中透過 CGI 來執行你的 Script。該環境已經安裝好以下的幾種 Shell：`sh`、`bash`、`zsh`、`fish`、`ksh`、`tcsh`。你的 Script 會被放在 `/usr/local/apache2/cgi-bin/` 裡，並且**該位置的上一層（i.e., `/usr/local/apache2/`）會是可寫的**。

如果要在自己電腦上測試，你可以執行下方的指令。所有在目前資料夾裡面的檔案皆會在 `http://localhost:8080/cgi-bin/` 底下。

```
docker run --rm -p 8080:80 -v $(pwd):/usr/local/apache2/cgi-bin -d shell-cgi
```

註：你也可以參考上面的 Example，把 Script 放在工作站上測試。但請注意，放在工作站的 Script 實際上是在其他機器上執行的，**該機器上缺少許多 Shell 跟指令**，所以可能會有無法正常執行的狀況。另外，為避免可能造成的資安漏洞，**請記得在結束後將不必留存的 Script 刪除或移出，若未執行本步驟，將審酌情況扣期中考之總分**。

Rules

- **此題限用 Shell Script。原則上 Docker 環境裡有的所有工具跟指令都可以使用**，但請不要嘗試使用網路連線或是安裝其他套件。
- 所有 Request 中，我們能控制的地方（像是 Request Path 跟 `username`）可能會出現的字元集合是 `[a-zA-Z0-9/-_&=?.:]`
- Response Body 請盡量遵守題目說明中的格式（每一行的行尾都要有換行），不過原則上些微的差異不會影響給分與否。

Hint

- Script 開頭記得要加 *shebang*
- Request 的各種資訊會在環境變數裡。另外如果是 POST Request 的話，Request Body 會在標準輸入（stdin）。
- `env` 這個指令可以印出所有環境變數，裡面也包含各種需要的輸入，請仔細檢查看看。
- `exec 2>&1` 把這個放在 Script 的前面可以讓你在 Response 裡面看到各種錯誤時的訊息。

Task 9.1 - Hello, CGI! ★ (5 points)

先來暖身一下吧！請在 Response 的第一行輸出 `Hello, CGI!`，並在第二行輸出該 Request 的 Path，在第三行輸出 Query string。舉例來說，當我們對 `<url to your script>/123/abc?meowmeow` 發送 GET Request 時，你的 Response 要長的像下面這樣：

```
Hello, CGI!
Path: /123/abc
Query: meowmeow
```

請注意，如果是對 `<url to your script>` 發送 Request，你的 Path 應該要是一個空字串（冒號後面還是要有恰好一個空格）。

Task 9.2 - Page View Count ★ (7 points)

接下來請實作一個瀏覽次數統計吧！Response 應該要是一個數字，代表該 Path 被瀏覽了幾次。另外，如果該次 Request 的 Query String **恰好（exactly）**是 `reset`，則請將該 Path 的瀏覽次數歸零重新計算。底下是一個可能的互動過程：

```
# curl "<url to your script>/path1"
1
# curl "<url to your script>/path1"
2
# curl "<url to your script>/path2"
1
# curl "<url to your script>/path1?reset"
1
# curl "<url to your script>/path1"
2
# curl "<url to your script>/path2?a"
2
# curl "<url to your script>/path2/"
1
```

Task 9.3 - Webserver inception ★★ (8 + 10 points)

當然，我們可以用 CGI 來寫個簡單的 Webserver。雖然說我們這邊是用 Apache，所以根本沒必要這樣做，不過管他的。此題分為兩個部份。

Task 9.3a (8 points)

請實作以下 endpoint:

- `GET /view`：從 `./data` 資料夾中取得相對應的檔案，傳給瀏覽器使其能夠顯示。須支援以下功能：
  1. 根據檔案副檔名更改 Content Type (MIME Type)。本題只需支援以下副檔名：.html，.css，.jpg, .mp3，可以假設不會出現其他的副檔名。
  2. 若檔案不存在，或者為一個目錄，回傳 Status Code 404 (Not found)。
  3. 若試圖取得的檔案位置在 `./data` 資料夾之上，如 `/view/../a.html`，回傳 Status Code 403 (Forbidden)。(這種攻擊手法叫做 Path Traversal)

實作完成後，將範例的頁面 (`<url to your script>/test.html`) 用瀏覽器打開，應能看到一個包含 CSS 格式、圖片和音樂播放器的網頁。

Task 9.3b (10 points)

鑑於安全考量，你可能會不想讓任何人都能看到你的網頁，因此需要驗證功能。接續 9.3a 實做以下 endpoint：

- 更改 `GET view`：檢查 cookie，若 cookie 中不存在 `totallylegituser` 一項，用 Status Code 302 (Found) 導向到 `/login`，否則正常取得檔案印出。

- `GET login`: 顯示 `login.html`。
- `POST login`：包含一個參數 `password`，檢查密碼的 sha256 hash 是否與 `secret.hash` 內一致，若驗證成功則設定 cookie `totallylegituser=1` 並印出任意的成功訊息，失敗則導向到 `/login`。

下面是一個用瀏覽器測試此 task 的可能過程：

```
GET http://localhost:8080/cgi-bin/view/test.html -> Redirect to /cgi-bin/login
GET http://localhost:8080/cgi-bin/login -> Display ./data/login.html
POST http://localhost:8080/cgi-bin/login -> Validate user and set cookie
GET http://localhost:8080/cgi-bin/view/test.html -> Display ./data/test.html
GET http://localhost:8080/cgi-bin/view/../test.html -> 403 Forbidden
```

Task 9 Hints

- 可以利用 `realpath` 來處理路徑。
- 使用 curl 傳送 POST 的語法如下：`curl <url> -X POST -d "password=meow"`
- curl 也可以傳 cookie: `curl --cookie "Name=Value" <url>`
- 若有 Cookie 的話，會出現在 CGI 的環境變數裡頭，設定 cookie 可以用 `Set-Cookie` 的 header。你可以利用各瀏覽器的 Cookie Manager 擴充元件來測試 cookie。
- 重新導向可以用 `Location` header 來設定目標。
- HTTP Status Code 可以用 `Status` header 來設定。對 HTTP status code 有興趣的話可以看 https://http.cat/ （跟考試無關）
- 允許用 ChatGPT 等 AI 工具寫這題，Modern problems require modern solutions.

Submission

此題限用 Shell Script。請直接上傳完整的 Script。