

NASA2022 Homework4

b08902149 資工三 徐晨祐

Network Administration

1. CIA Triad & Threat Modeling

(1.)

- a. 在2022年1月，紅十字會遭到網路攻擊，導致51萬筆人道救援對象個資外洩。此事件違反confidentiality，因為紅十字會的系統之自助式密碼管理平台存在漏洞，導致機密資料可以透過未經授權的管道取得。
- b. 在2022年1月17日，Crypto.com發生平臺用戶的加密貨幣被不明人士盜轉。此事件違反了integrity，因為數據遭到了未經授權者的竄改。

(Ref1: <https://www.techtarget.com/whatis/definition/Confidentiality-integrity-and-availability-CIA>)

(Ref2: <https://www.ithome.com.tw/news/149413>)

(Ref3: <https://www.thenewslens.com/article/164125>)

(2.)

Assumption: 使用者登入有設置密碼，並且筆電可正常使用。

Threat Model	Countermeasure
有人趁著筆電擁有者暫時離開電腦，且螢幕還未上鎖時，直接使用其筆電。	使用者一旦要離開筆電身邊，就必須手動上鎖以免有心人士在此時擅自使用筆電。
有人嘗試側錄筆電登入時的密碼。	定期更換密碼，並且在執行登入的同時留意周遭。

(沒有討論對象)

(3.)

Assumption: 手機功能一切正常。

Threat Model	Countermeasure	
有人嘗試以他人的手機掃實聯制。	更改目前掃實聯質設定，改成需要多輸入身分證後四碼。	
有人嘗試以某種方式攔截他人的簡訊封包並修改之後轉傳。	確保實聯制的簡訊傳送都是加密的進行，並且檢視是否同一人發送的簡訊是來自同一支手機。	

(沒有討論對象)

(4.)

Assumption: 學生們有足夠的設備可以進行助教要求的監考設定，助教可以監控學生的所在位置。

Threat Model	Countermeasure	
有人嘗試用任何軟體進行視訊溝通討論。	每組指派一位助教，並且要求每位同學向助教分享螢幕。	
有人嘗試攔截其他小組繳交解答時的資訊。	確保繳交的方式是有加密過的，避免被監聽。	

(沒有討論對象)

2. Web Security

(1.)


OWASP Juice Shop

+

https://nasahw4-juiceshop.herokuapp.com/#/score-board

🔍🛒6🌐

☰

OWASP Juice Shop

🔍🛒6🌐

Score Board15%

Coding Score0%

19/12

26/12

30/22

40/25

50/18

60/11

Show all

🏆Show solved

🎓Show tutorials only

⚠️Show unavailable

📄

🔄

Broken Access Control

Broken Anti Automation

Broken Authentication

Cryptographic Issues

Improper Input Validation

Injection

Insecure Deserialization

Miscellaneous

Security Misconfiguration

Security through Obscurity

Sensitive Data Exposure

Unvalidated Redirects

Vulnerable Components

XSS

XXE

Hide all

(2.)

- **Injection:** Injection包含許多類型，例如Command Injection，SQL Injection，是發生在不安全的資料以command或request的形式送給interpreter
- **Broken Access Control:** 是經常發生的攻擊，對經過身份驗證的使用者沒有實施足夠的access control，常見的例子有透過修改URL或HTML頁面，或是使用攻擊工具修改API請求來繞過Access control的檢查。
- **Security Misconfiguration:** 使用不安全的預設值、錯誤的HTTP標頭配置，或是包含敏感訊息的Error message。常見的例子有：實際上現時應該刪除的檔案未刪除，或保留某些預設帳號。

(Ref1: <https://www.netadmin.com.tw/netadmin/zh-tw/technology/20C88D4CA5C44881A7F745174840D6B4?page=2>)

(Ref2: <https://www.cloudprotector.com/owasp-top-10/>)

(Ref3: <https://segmentfault.com/a/1190000041063058>)

(3.)

CSRF是指cross site request forgery。大部分網站都是使用cookie或session進行登入驗證，當通過驗證後網站就會給五們一個通行證存在cookie或session裏，這樣就無需每個東作都進行驗證，而CSRF就是透過這樣的機制所存在的漏洞進行攻擊。當我們得到某個A網站所發放的憑證且記錄在cookie上之後，如果同時又去瀏覽了其他網站結果不小心瀏覽到了惡意網站，該惡意網站可以透過複製你在cookie中紀錄的通行證來得到你在原來的A網站上的權限。

(Ref: <https://www.imperva.com/learn/application-security/csrf-cross-site-request-forgery/>)

3. Linux Q&A

(1.)

密碼會存在 `/etc/shadow`。當使用 `passwd` command時，我們是在用 `/usr/bin/passwd` 這個執行檔，它是由root所擁有，且有setuid，因此當我們執行 `passwd` 時，effective uid會變成root，我們此時就對 `/etc/shadow` 有 `rw` 的權限。

(Ref1: <https://stackoverflow.com/questions/71839786/in-the-case-of-a-normal-user-with-no-access-to-etc-shadow-file-how-can-the-pass>)

(Ref2: <https://www.796t.com/content/1548998843.html>)

(2.)

存放在 `/var/log/auth.log`。這個檔案紀錄了包含系統授權信息，包括用戶登錄和使用的權限機制等。

(Ref: <https://linux-audit.com/file-permissions-of-the-etc-shadow-password-file/>)

(3.)

User account不只是給人使用，有些account是為了執行系統服務而建立的。會這樣做的原因是，有時系統服務之間的資源區隔會需要和human user之間的資源區隔有同要的機制。此外，相較於全部服務都交給root去執行，當有一個bug出現在其中一項系統服務時，創建專門的使用者來執行某些系統服務也能避免full system attack的問題，因為攻擊者會被限制只能以該服務被賦予的permission來攻擊；但如果是用root來執行所有服務，當bug產生時，就會因為root擁有的權限過大而造成大規模的風險。

(Ref: <https://unix.stackexchange.com/questions/197124/why-are-there-many-accounts-im-the-only-user>)

(4.)

Linux kernel可以透過namespace來進行kernel resource的隔離，使得一部分的process只能使用某部分的resource，另一部分的process也只能看另一部分的resource。

兩個例子：

- Network namespace: 可以允許某些process擁有獨立的網路設備、IP Address、路由、port。
- User namespace: 可以讓同樣一個使用者在不同的user namespace擁有不同的user id，group id。除了系統預設的user namespace之外，所有user namespace都有一個父 user namespace；每個user namespace都可以有零到多個子user namespace。

(Ref: https://philipzheng.gitbook.io/docker_practice/underly/namespace)

4. Cryptography

(1.)

Flag: `HW4{i_came_i_saw_i_conquered_veni_vidi_vici!}`

解密所寫的程式在 `security` 資料夾中的 `decrypt.py`。

由於我沒辦法知道 `encrypt.py` 一開始random到的到底是什麼數字，但知道最初的 `key` 一定是在 $0 \leq key < 26$ 這個範圍，因此我可以暴力的從0開始假設最初的 `key`，並且依照 `encrypt.py` 裡的邏輯去逆推flag，並且把26種可能都印出來，看哪一個是有意義的。

(Ref: 助教提供的 `encrypt.py`)

(2.)

Flag: `HW4{HeLlO_Ev3_can_y0u_h3ar_ouR_v0iCe?}`

解密所寫的程式在 `security` 資料夾中的 `hack.py`。

當 p 是質數且 a 是整數時，由**Fermat's little theorem**可以得知 $g^p \equiv g \pmod{p}$ ，代表 $g^{p-1} \equiv 1 \pmod{p}$ ，也就是說 $g^{\lfloor \frac{p}{2} \rfloor} \cdot g^{\lfloor \frac{p}{2} \rfloor} \equiv 1 \pmod{p}$ ，因此可知 $g^{\lfloor \frac{p}{2} \rfloor} \equiv 1 \text{ or } p-1 \pmod{p}$ 。

由於我知道這題在實作**Diffie-Hellman**時，Alice會將 a 設為 $a = \lfloor \frac{p}{2} \rfloor + i$ ，其中 $0 \leq i < 65536$ ，因此我可以遍歷過所有的 i ，只要 $g^i \equiv g^a \pmod{p}$ 或 $(p-1) \cdot g^i \equiv g^a \pmod{p}$ 成立，就找到 a 是多少。

找到之後透過 `self.a = i + self.p // 2` 以及 `self.genSharedKey(self.g_b)` 即可設定好shared key，此時就可以decrypt了。

(Ref1: <https://zh.wikipedia.org/zh-tw/迪菲-赫爾曼密鑰交換>)

(Ref2: <https://zh.wikipedia.org/zh-tw/费马小定理>)

(3.)

Flag: `HW4{A_5Tr0nG_Pa5sw04D_I5_lmP0RtAn7}`

我先在vm和本機之間建立了一個shared directory，並且讓ubuntu進入recovery mode(開機時長按shift)，藉此將 `/etc/shadow` copy到該shared directory。我在我的本機將該shadow檔中willy的部分複製到另一個檔案，之後用 `hashcat` 去解密，指令為 `hashcat -a 0 -m 1800 passwd.txt wordlist.txt`，其中 `wordlist.txt` 是我在<https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/10-million-password-list-top-1000000.txt>下載的。最後得到密碼為：`yalyasya22`。

```
OpenCL -- -zsh -- 118x26

$6$GYBNlTnG2J0qRHff$POTERafsbpVLOG.qnG5ZjBs1R8XZXlmtcDw/AbAFvYt//dNB4guECBI5yAVx487nx1VCvgqKgi9xM4/TurI3g/:yalyasya22

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target.....: $6$GYBNlTnG2J0qRHff$POTERafsbpVLOG.qnG5ZjBs1R8XZXlmtcDw/AbAFvYt//dNB4guECBI5yAVx487nx1VCvgqKgi9xM4/TurI3g/
Time.Started.....: Fri Apr 29 14:22:57 2022 (8 mins, 5 secs)
Time.Estimated...: Fri Apr 29 14:31:02 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/Users/chrischyxx/Desktop/willy-ubuntu/wordlist)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 1735 H/s (15.15ms) @ Accel:128 Loops:1024 Thr:1 Vec:2
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 856320/999998 (85.63%)
Rejected.....: 0/856320 (0.00%)
Restore.Point....: 856192/999998 (85.62%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4096-5000
Candidate.Engine.: Device Generator
Candidates.#1....: yama1919 -> yalolo
Hardware.Mon.SMC.: Fan0: 100%, Fan1: 100%
Hardware.Mon.#1..: Temp: 78c

Started: Fri Apr 29 14:22:50 2022
Stopped: Fri Apr 29 14:31:04 2022
(base) chrischyxx@kelisixiaoxudeMacBook-Pro OpenCL %
```

(Ref1: <https://github.com/hashcat/hashcat/issues/2270>)

(Ref2: <https://github.com/hashcat/hashcat/issues/3044>)

5. WiFi Hacking

(1.)

因為我使用的是macOS，所以以下指令可能部分只能用在macOS。

Step1: 透過以下command讓我們之後可以直接打 `airport` 來用：

```
sudo ln -s
/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airpo
rt /usr/local/bin/airport
```

Step2: 接著透過 `airport -s` 來搜尋附近的網路。我們可以發現Mysterious Room的Channel為2。

Step3: 透過 `airport en0 sniff 2`，來等待如果有人成功登入，我們可以在登入過程中獲取兩組keys，然後offline的破解它。要注意的是，如果連接wifi的網卡不是 `en0`，需要把 `en0` 替換成連接WiFi的網卡。此外 `sniff` 後面的數字，要和欲破解的WiFi的channel一致。

Step4: 透過 `brew install aircrack-ng` 安裝aircrack-ng。

Step5: `aircrack-ng /tmp/airportSniffxRxlUA.cap`

Step6: 我生了一個dictionary file `dict.txt`，內容是從0900000000~0999999999的所有可能。接著使用 `aircrack-ng -w dict.txt -b 54:3D:37:3D:81:18 /tmp/airportSniffxRxlUA.cap`，其中 `54:3D:37:3D:81:18` 是Mysterious room的BSSID，`/tmp/airportSniffxRxlUA.cap` 是Step2時所生出

的 .cap 檔。

可以得到下列結果：

```
chrischyhx — b08902149@linux1:~ — zsh — 80x24

Aircrack-ng 1.6

[00:22:46] 18274096/100000000 keys tested (13598.04 k/s)

Time left: 1 hour, 40 minutes, 10 seconds          18.27%

KEY FOUND! [ 0918273645 ]

Master Key      : A6 38 BB F2 F3 D9 82 67 0E 19 E1 1B 52 CC AA F9
                  C4 17 08 D1 A6 23 CF C0 9D E4 C9 72 F6 7D 62 69

Transient Key   : A8 03 9F 0F 34 81 02 57 7E 51 6D 5A D4 17 8F 9E
                  34 C5 0F 1C B5 30 64 A2 E9 AD 0E 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC     : B7 78 A6 53 5E FA AE B6 74 60 33 8D 66 0A 18 3C

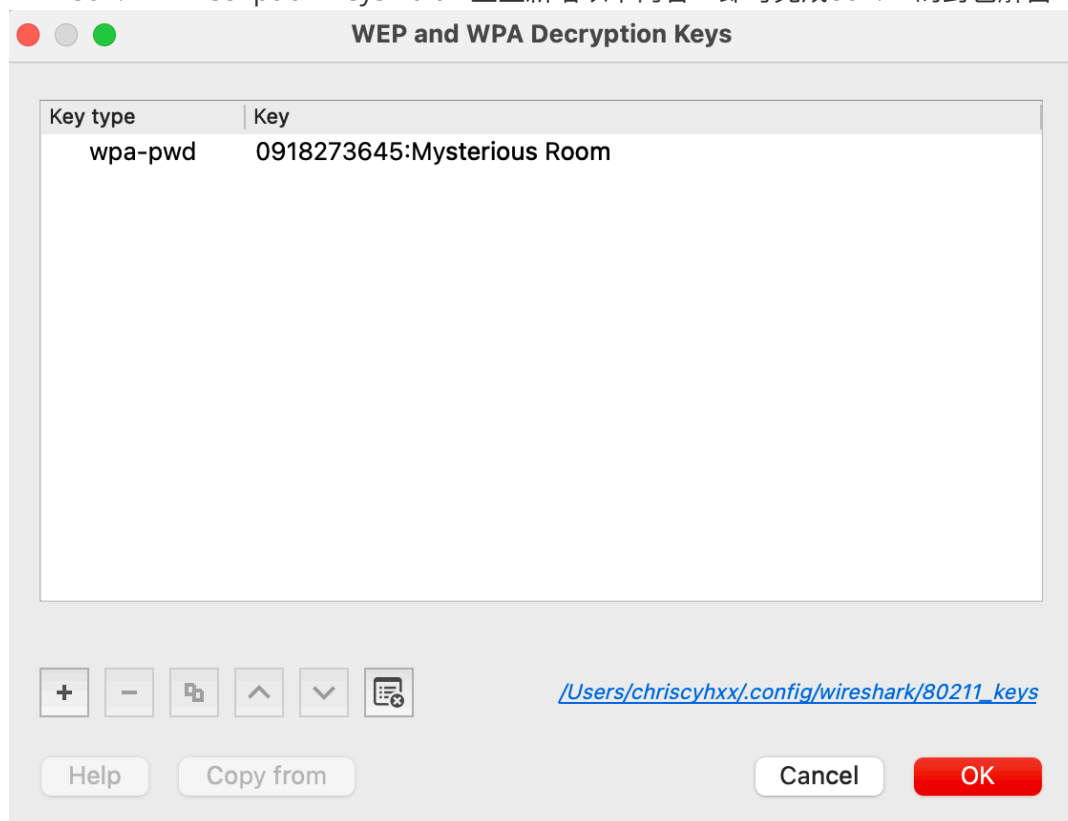
(base) chrischyhx@kelisixiaoxudeMacBook-Pro ~ %
```

可以得到WiFi密碼就是 0918273645。

(Ref: <https://programmer.group/mac-os-cracking-wifi-wpa-wpa2.html>)

(2)

先將上題得到的 .cap 檔轉乘 .pcap 檔，透過wireshark開啟，在wireshark的Preferences中，選擇Protocols > IEEE 802.11 > Decryption Keys Edit。並且新增以下內容，即可完成802.11的封包解密。



(Ref: <https://wiki.wireshark.org/HowToDecrypt802.11>)

System Administration

設定

(1.)

以下為安裝流程：

先在vm上安裝好CentOS並且新增一張網卡attach到bridge adaptor。

接下來輸入指令：

```
yum install -y openldap-servers openldap-clients
systemctl enable slapd
systemctl start slapd
```

用上述指令安裝並啟動 ldap。

接著建立一個 suffix.ldif 檔，內容如下：

```
dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcSuffix
olcSuffix: dc=nasa,dc=csie,dc=ntu
```

使用 ldapmodify -Y EXTERNAL -H ldapi:/// -f suffix.ldif 儲存修改。

建立一個 root.ldif 檔，內容如下：

```
dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcRootDN
olcRootDN: cn=nasa,dc=nasa,dc=csie,dc=ntu
-
replace: olcRootPW
olcRootPW: {SSHA}prTzIXh92BV0MfoOYC8N9HEyOULqggcG
```

其中 olcRootPW 是以 slapasswd 輸入密碼 nasa2022 所得到的結果。

再用 ldapmodify -Y EXTERNAL -H ldapi:/// -f root.ldif 儲存修改。

以下列三個指令新增schemas：

```
ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/cosine.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/nis.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/inetorgperson.ldif
```

再用新增一個 base.ldif，內容如下：

```
dn: dc=nasa,dc=csie,dc=ntu
dc: nasa
objectClass: top
```

```
objectClass: domain

dn: cn=nasa,dc=nasa,dc=csie,dc=ntu
cn: nasa
objectClass: organizationalRole
description: admin

dn: ou=people,dc=nasa,dc=csie,dc=ntu
objectClass: organizationalUnit
ou: people

dn: ou=group,dc=nasa,dc=csie,dc=ntu
objectClass: organizationalUnit
ou: group
```

透過 `ldapadd -x -W -D "cn=nasa,dc=nasa,dc=csie,dc=ntu" -H ldapi:/// -f base.ldif` 儲存修改。

以下為 `ldapsearch -x -b "dc=nasa,dc=csie,dc=ntu"` 的輸出結果：

```
[[root@localhost certs]# ldapsearch -x -b "dc=nasa,dc=csie,dc=ntu"
# extended LDIF
#
# LDAPv3
# base <dc=nasa,dc=csie,dc=ntu> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# nasa.csie.ntu
dn: dc=nasa,dc=csie,dc=ntu
dc: nasa
objectClass: top
objectClass: domain

# nasa, nasa.csie.ntu
dn: cn=nasa,dc=nasa,dc=csie,dc=ntu
cn: nasa
objectClass: organizationalRole
description: admin

# people, nasa.csie.ntu
dn: ou=people,dc=nasa,dc=csie,dc=ntu
objectClass: organizationalUnit
ou: people

# group, nasa.csie.ntu
dn: ou=group,dc=nasa,dc=csie,dc=ntu
objectClass: organizationalUnit
ou: group
```

Reference: NTU CSIE NASA2022 Lab8講義

(2.)

為了提供連線，ldap必須先製作憑證，這邊我們用自定義的CA來為我們簽署憑證。(以下在CentOS那台機器下操作)

```
cd /etc/openldap/certs
# Create the root key
openssl genrsa -out nasahw4_RootCA.key 2048
# create the self-signed root certificate
# 需要填一些資料，我除了common name填nasa.csie.ntu，其他都亂填
openssl req -x509 -new -nodes -key nasahw4_RootCA.key -sha256 -days 1024 -out nasahw4_RootCA.pem
# create a private key for LDAP server
openssl genrsa -out nasahw4_ldap.key 2048
# create a certificate signing request
openssl req -new -key nasahw4_ldap.key -out nasahw4_ldap.csr
# 透過CA的key和CA的certificate來製造憑證
openssl x509 -req -in nasahw4_ldap.csr -CA nasahw4_RootCA.pem -CAkey nasahw4_RootCA.key -CAcreateserial -out nasahw4_ldap.crt -days 1460 -sha256
chown -R ldap:ldap /etc/openldap/certs/nasahw4_*
```

建立一個cert.ldif檔，內容如下：

```
dn: cn=config
changetype: modify
replace: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/openldap/certs/nasahw4_RootCA.pem
-
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/openldap/certs/nasahw4_ldap.key
-
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/openldap/certs/nasahw4_ldap.crt
```

透過 ldapmodify -Y EXTERNAL -H ldapi:/// -f cert.ldif 去更新ldap。可以再透過 slapcat -b

"cn=config" | egrep

"olcTLSCertificateFile|olcTLSCertificateKeyFile|olcTLSCACertificateFile" 來檢查是否有完成設定。

Configure OpenLDAP to listen over SSL:

更改 /etc/sysconfig/slapd 的 SLAP_URLS

```
SLAPD_URLS="ldapi:/// ldap:/// ldaps:///"
```

更改 /etc/openldap/ldap.conf

```
TLS_CACERTDIR /etc/openldap/certs
TLS_CACERT /etc/openldap/certs/nasahw4_RootCA.pem
# 因為是self-signed 所以allow所有憑證
TLS_REQCERT allow
```

重啟服務：`systemctl restart slapd`。

用 `yum install net-tools` 安裝 `netstat` 之後，可以用 `netstat -antup` 去看 `slapd` 跑在哪個port上listen，應該要是636。

設定防火牆：

```
firewall-cmd --permanent --add-service=ldap
firewall-cmd --permanent --add-service=ldaps
firewall-cmd --reload
```

以上為server的設定。

接下來先在vm上將arch linux安裝完並新增一張網卡attach到bridge adaptor之後，以下列指令安裝 `sssd` 和 `openldap`。

```
pacman -Sy sssd
pacman -Sy openldap
# 之後會用到此資料夾，先新增起來
mkdir -p /etc/openldap/certs
```

我們在執行 `sssd` 與ldap server連線時，需要讓 `sssd` 認得對方的憑證所發放的機構，因此我們需要執行以下指令把CentOS上所存的Root CA憑證copy過來。以下指令在CentOS那台機器執行。

```
scp /etc/openldap/certs/nasahw4_RootCA.pem /etc/openldap/certs/nasahw4_RootCA.key
root@192.168.88.130:/etc/openldap/certs
```

上面這個指令的 `root@192.168.88.130` 是Arch Linux上的ip，可用 `ip a` 指令查詢得知。

回到Arch Linux那台機器，改動 `/etc/openldap/ldapconf` 的以下內容：

```
BASE dc=nasa,dc=csie,dc=ntu
URI ldaps://nasa.csie.ntu ldap://nasa.csie.ntu
TLS_REQCERT demand
TLS_CACERT /etc/openldap/certs/nasahw4_RootCA.pem
TLS_CACERTDIR /etc/openldap/certs
```

為了方便，我想讓機器認得 `nasa.csie.ntu` 這個domain name對應到CentOS那台機器的IP，也就是 `192.168.88.124`，在 `/etc/hosts` 加上這行：`192.168.88.124 nasa.csie.ntu`。
之後修改 `/etc/sss/sss.conf` 為以下內容：

```
[sss]
config_file_version = 2
services = nss, pam, sudo
```

```
domains = ldap

[domain/ldap]
cache_credentials = true
enumerate = true

id_provider = ldap
auth_provider = ldap
ldap_uri = ldaps://nasa.csie.ntu
ldap_search_base = dc=nasa,dc=csie,dc=ntu
ldap_tls_reqcert = demand
ldap_tls_cacert = /etc/openldap/certs/nasahw4_RootCA.pem
chpass_provider = ldap
ldap_chpass_uri = ldaps://nasa.csie.ntu
entry_cache_timeout = 600
ldap_network_timeout = 2
ldap_sudo_search_base = ou=SUDOers,dc=nasa,dc=csie,dc=ntu

ldap_schema = rfc2307
ldap_group_member = memberUid
```

記得用 `chmod 600 /etc/sss/sss.conf` 更動permission。

在 `/etc/nscd.conf` 改動以下幾行的內容：

```
enable-cache    passwd    no
enable-cache    group      no
enable-cache    hosts      yes
enable-cache    netgroup   no
```

將 `/etc/nsswitch.conf` 改成以下的內容：

```
passwd:  files sss
group:   files sss
shadow:  files sss
sudoers: files sss

publickey: files

hosts:   files myhostname dns
networks: files

protocols: files
services: files
ethers:   files
rpc:      files

netgroup: files
```

為了改讓 `ldap` 去主管身份驗證的工作，將 `/etc/pam.d/system-auth` 改成以下內容：

```
auth      sufficient      pam_sss.so      forward_pass
auth      required        pam_unix.so      try_first_pass nullok
auth      optional        pam_permit.so
auth      required        pam_env.so

account [default=bad success=ok user_unknown=ignore authinfo_unavail=ignore] pam_sss.so
account  required        pam_unix.so
account  optional        pam_permit.so
account  required        pam_time.so

password sufficient      pam_sss.so      use_authok
password required        pam_unix.so      try_first_pass nullok shadow sha512
password optional        pam_permit.so

session  required        pam_mkhomedir.so skel=/etc/skel/ umask=0077
session  optional        pam_sss.so
session  required        pam_limits.so
session  required        pam_unix.so
session  optional        pam_permit.so
```

為了要讓user可以 `sudo`，做以下兩個檔案的更改。

將 `/etc/pam.d/su` 改成以下內容：

```
auth      sufficient      pam_rootok.so
auth      sufficient      pam_sss.so      forward_pass
auth      required        pam_unix.so

account [default=bad success=ok user_unknown=ignore authinfo_unavail=ignore] pam_sss.so
account  required        pam_unix.so
session  required        pam_unix.so
session  optional        pam_sss.so
password include         system-auth
```

將 `/etc/pam.d/sudo` 改成以下內容：

```
auth      sufficient      pam_sss.so
auth      include         system-auth
account   include         system-auth
session   include         system-auth
```

為了讓user可以自行更改密碼，將 `/etc/pam.d/passwd` 改成以下內容：

```
password      sufficient      pam_sss.so
password      required        pam_unix.so sha512 shadow nullok
```

接著透過以下指令開啟 `sssd` 的服務：

```
systemctl enable sssd.service  
systemctl start sssd.service
```

這樣就成功了，可以用一些 `ldap` 的指令操作看看。

Reference:

LDAP: <https://coodie-h.blogspot.com/2017/09/centos-7openldap.html>

製作憑證: <https://www.itzgeek.com/how-tos/linux/centos-how-tos/configure-openldap-with-ssl-on-centos-7-release-7.html>

SSSD連線LDAP server: https://wiki.archlinux.org/title/LDAP_authentication#Online_and_Offline_Authentication_with_SSSD

sssd.conf的格式設定: <https://man.archlinux.org/man/sss.conf.5.en>

PAM: <https://wiki.archlinux.org/title/PAM>

PAM入門介紹：<https://lagunawang.pixnet.net/blog/post/5206841-pam>入門介紹

討論對象:

b05504066 李旻翰

b09505014 王聖文

(3.)

回到centOS那台機器。

我需要增加一個檔案叫 `sudoschema.ldif`，但我們必須先知道ldap目前

在 `/etc/openldap/slapd.d/cn=config/cn=schema` 有幾分檔案，因為新增的這個檔案內的cn必須有編號，如果是第*i*個檔案就要標*i* - 1。因此我們：

```
$ ls /etc/openldap/slapd.d/cn=config/cn=schema
# output 如下
cn={0}core.ldif  cn={1}cosine.ldif  cn={2}nis.ldif  cn={3}inetorgperson.ldif
```

可知道這個 `sudoschema.ldif` 內要加的編號是 `{4}`，其內容如下：

```
dn: cn={4}sudo,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: {4}sudo
olcAttributeTypes: ( 1.3.6.1.4.1.15953.9.1.1
    NAME 'sudoUser'
    DESC 'User(s) who may run sudo'
    EQUALITY caseExactIA5Match
    SUBSTR caseExactIA5SubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
olcAttributeTypes: ( 1.3.6.1.4.1.15953.9.1.2
    NAME 'sudoHost'
    DESC 'Host(s) who may run sudo'
    EQUALITY caseExactIA5Match
    SUBSTR caseExactIA5SubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
olcAttributeTypes: ( 1.3.6.1.4.1.15953.9.1.3
    NAME 'sudoCommand'
    DESC 'Command(s) to be executed by sudo'
    EQUALITY caseExactIA5Match
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
olcAttributeTypes: ( 1.3.6.1.4.1.15953.9.1.4
    NAME 'sudoRunAs'
    DESC 'User(s) impersonated by sudo (deprecated)'
    EQUALITY caseExactIA5Match
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
olcAttributeTypes: ( 1.3.6.1.4.1.15953.9.1.5
    NAME 'sudoOption'
    DESC 'Options(s) followed by sudo'
    EQUALITY caseExactIA5Match
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
olcAttributeTypes: ( 1.3.6.1.4.1.15953.9.1.6
    NAME 'sudoRunAsUser'
    DESC 'User(s) impersonated by sudo'
    EQUALITY caseExactIA5Match
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
olcAttributeTypes: ( 1.3.6.1.4.1.15953.9.1.7
```



```

NAME 'sudoRunAsGroup'
DESC 'Group(s) impersonated by sudo'
EQUALITY caseExactIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
olcAttributeTypes: ( 1.3.6.1.4.1.15953.9.1.8
NAME 'sudoNotBefore'
DESC 'Start of time interval for which the entry is valid'
EQUALITY generalizedTimeMatch
ORDERING generalizedTimeOrderingMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
olcAttributeTypes: ( 1.3.6.1.4.1.15953.9.1.9
NAME 'sudoNotAfter'
DESC 'End of time interval for which the entry is valid'
EQUALITY generalizedTimeMatch
ORDERING generalizedTimeOrderingMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 )
olcAttributeTypes: ( 1.3.6.1.4.1.15953.9.1.10
NAME 'sudoOrder'
DESC 'an integer to order the sudoRole entries'
EQUALITY integerMatch
ORDERING integerOrderingMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )
olcObjectClasses: ( 1.3.6.1.4.1.15953.9.2.1 NAME 'sudoRole' SUP top STRUCTURAL
DESC 'Sudoer Entries'
MUST ( cn )
MAY ( sudoUser $ sudoHost $ sudoCommand $ sudoRunAs $ sudoRunAsUser $
sudoRunAsGroup $ sudoOption $ sudoOrder $ sudoNotBefore $ sudoNotAfter $ description )
)

```

以 `ldapadd -Y EXTERNAL -H ldapi:/// -f sudoschema.ldif` 新增該schema。

我們要再新增一個檔案 `sudobs.ldif` 內容如下：

```

dn: ou=SUDOers,dc=nasa,dc=csie,dc=ntu
objectClass: top
objectClass: organizationalUnit
ou: SUDOers

dn: cn=defaults,ou=SUDOers,dc=nasa,dc=csie,dc=ntu
objectClass: top
objectClass: sudoRole
cn: defaults
sudoOption: !visiblepw
sudoOption: always_set_home
sudoOption: env_reset
sudoOption: requiretty

dn: cn=%op,ou=SUDOers,dc=nasa,dc=csie,dc=ntu
objectClass: top
objectClass: sudoRole

```

```
cn: %op
sudoCommand: ALL
sudoHost: ALL
sudoOption: !authenticate
sudoRunAsUser: ALL
sudoUser: %op
```

以 `ldapadd -x -W -D "cn=nasa,dc=nasa,dc=csie,dc=ntu" -H ldapi:/// -f sudobs.ldif` 新增base。
我們接下來先設定可以 `sudo` 的user會用哪個 `gid`，我設定為2000，同時也新增一個可以 `sudo` 的user，其 `gid` 要是2000，我們先新增 `superuser.ldif`，內容如下：

```
dn: cn=op,ou=group,dc=nasa,dc=csie,dc=ntu
objectClass: posixGroup
cn: op
gidNumber: 2000

dn: uid=superuser,ou=people,dc=nasa,dc=csie,dc=ntu
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: superuser
uid: superuser
uidNumber: 7775
gidNumber: 2000
homeDirectory: /home/superuser
loginShell: /bin/bash
userPassword: {SSHA}ZvRtcEFPYRyYf1CfN/Qs1mVXytHh6xm5
```

再用 `ldapadd -x -W -D "cn=nasa,dc=nasa,dc=csie,dc=ntu" -H ldapi:/// -f` 來新增user。
接下來，加一個user叫 `no_sudo`，其密碼為nasa2022。新增檔案 `no_sudo.ldif`，內容如下：

```
dn: uid=no_sudo,ou=people,dc=nasa,dc=csie,dc=ntu
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: no_sudo
uid: no_sudo
uidNumber: 9999
gidNumber: 100
homeDirectory: /home/no_sudo
loginShell: /bin/bash
userPassword: {SSHA}17TWrr/nrY1v5moW0bFPyhhwRUGcY8QaY
```

再用 `ldapadd -x -W -D "cn=nasa,dc=nasa,dc=csie,dc=ntu" -H ldapi:/// -f no_sudo.ldif` 完成user的新增。

在 `/etc/sudo-ldap.conf` 新增下列兩行：

```
uri ldap:///
sudoers_base ou=SUDOers,dc=nasa,dc=csie,dc=ntu
```

在 `/etc/nsswitch.conf` 新增下列一行：

```
sudoers:      files ldap
```

這樣就完成了，可以回到arch linux那台機器試試看換到 `no_sudo` 和 `superuser` 之後，可不可以使用 `sudo`。

Reference:

<http://jasee.github.io/2014/03/28/openldap-use-sudo.html>

<https://developer.aliyun.com/article/484098>

討論對象:

b05504066 李旻翰

b09505014 王聖文

(4.)

由於ACL會由上往下去檢查規則，因此我們第一條規則是要設定只讓ldap client去連接到ldap server；而第二條要設定的是不能讓任何人去修改自己的 `homeDirectory`，`uidNumber`，`gidNumber` 的這項規定；最後是要設定讓任何人都可以修改自己的東西，但不能修改別人的。如此一來根據這樣的順序，就能達到題目要求，因此我們先新增一個 `acl.ldif`，內容如下：

```
dn: olcDatabase={2}hdb,cn=config
changetype: modify
add: olcAccess
olcAccess: {0}to dn.base="dc=nasa,dc=csie,dc=ntu"
    by peername.ip=192.168.88.130 manage
    by * none
-
add: olcAccess
olcAccess: {1}to attrs=uidNumber,homeDirectory,gidNumber
    by * read
-
add: olcAccess
olcAccess: {2}to *
    by self write
    by * read
```

再透過 `ldapmodify -Y EXTERNAL -H ldapi:/// -f acl.ldif` 來新增設定，完成。

討論對象:

b05504066 李旻翰

b09505014 王聖文

腳本

(1.)

安裝以下內容：

```
yum install epel-release
yum install python36
yum -y install python3-pip
yum install openldap-devel python-devel
yum install python3-devel
yum groupinstall "Development Tools"
python3 -m pip install python-ldap
python3 -m pip install passlib
```

安裝 `python-ldap` 是為了可以用該函式庫與ldap做互動，而安裝 `passlib` 是為了使用加密密碼的一些函式。
我將script命名為 `addUser.py`，放在 `ldap` 資料夾中。

Reference:

<https://gist.github.com/plugandplay/1401980>

(2.)

安裝以下內容：

```
pacman -Sy python3
pacman -Sy python-pip
pip install ldap3
```

安裝 `ldap3` 是為了可以用該函式庫與ldap做互動。
我將script命名為 `changeName.py`，放在 `ldap` 資料夾中。

Reference:

如何連線ldap: <https://ldap3.readthedocs.io/en/latest/connection.html>

如何modify: <https://ldap3.readthedocs.io/en/latest/modify.html>