HOMEWORK 4

Web Services & NFS FireWall

phkoan, liyi

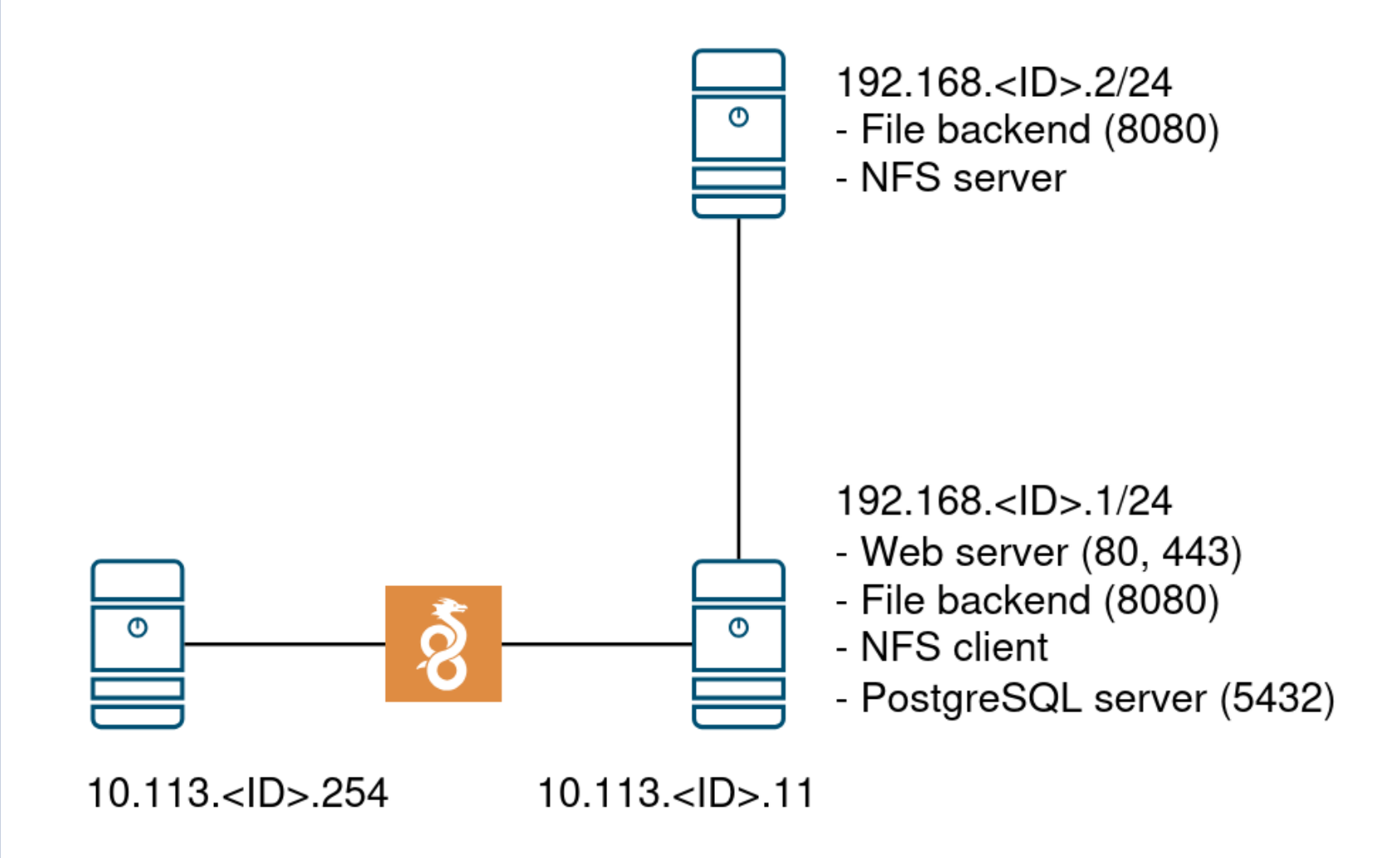
國立陽明交通大學資工系資訊中心

Information Technology Center, Department of Computer Science, NYCU

Outline

- HTTP Server (90%)
  - Virtual Hosts (3%)
  - Common
    - Hide Server Information (3%)
    - HTTPS and PKI (12%)
    - Access Control (3%)
    - Normal Logging (8%)
    - Verbose Logging (16%)
    - Log Rotate (5%)
  - file.{ID}.cs.nycu (20%)
  - Database & adminer.{ID}.cs.nycu (20%)
- FireWall (20%)
  - General rules (8%)
  - SSH failed login (12%)
- NFS (10%)
  - Server (4%)
  - Client (6%)

Architecture



HTTP Server

國立陽明交通大學資工系資訊中心
Information Technology Center, Department of Computer Science, NYCU

Reminder

- No matter which OS, use 10.113.{ID}.11 to do this homework
- You can use any web server to complete this homework, but only Nginx is guaranteed to pass.
- Be sure to clean all your log file before judge

Virtual Host (3%)

- Set up a name-based virtual host.
- Show different contents based on different domain
  - Your Domain Name: `nasa.{ID}.cs.nycu`
    - {ID} is your wireguard ID
- Make `[nasa|file|adminer].{ID}.cs.nycu` can be resolved in your machine

Hint:
You can use hosts file to map ip to your domain.

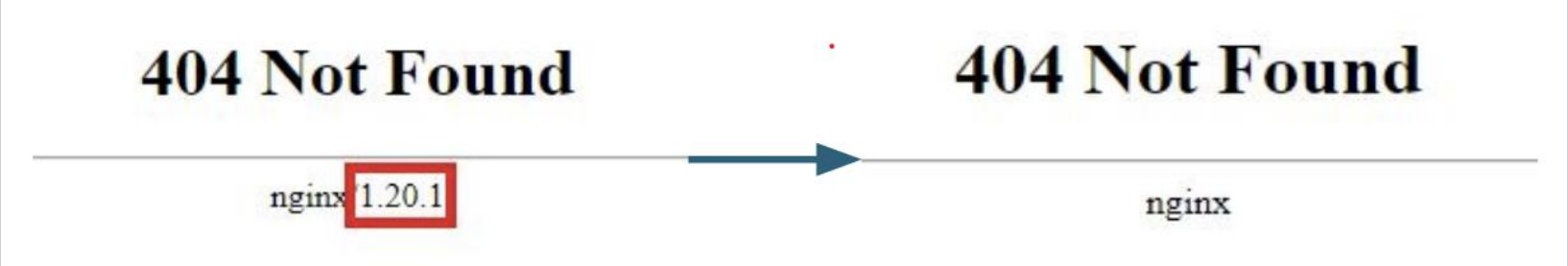
- On FreeBSD, CentOS, Ubuntu: /etc/hosts
- On Windows: C:\Windows\System32\drivers\etc\hosts

Virtual Host: Content (3%)

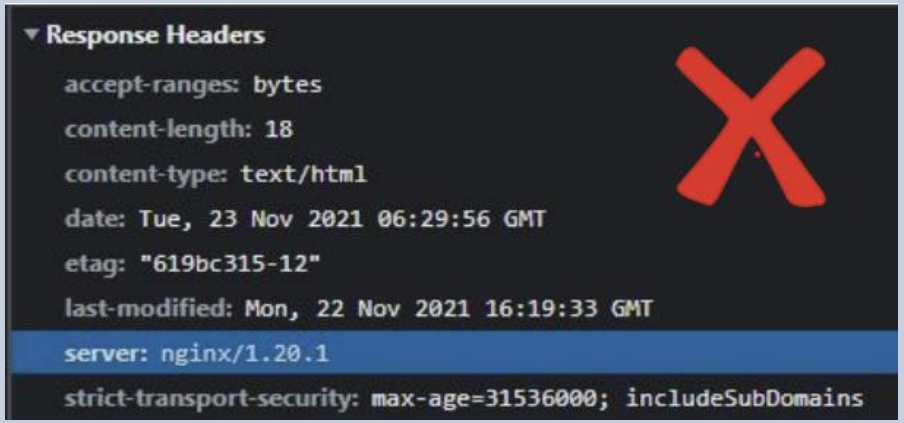
- nasa.{ID}.cs.nycu (1%)
  - Raw text: 2024-nycu.sa-hw4-vhost
- file.{ID}.cs.nycu
  - The backend mentioned later
- adminer.{ID}.cs.nycu
  - The adminer service, also mentioned later
- *.{ID}.cs.nycu (2%)
  - For the wildcard subdomain that not mentioned above, redirects the user to https://http.cat/404

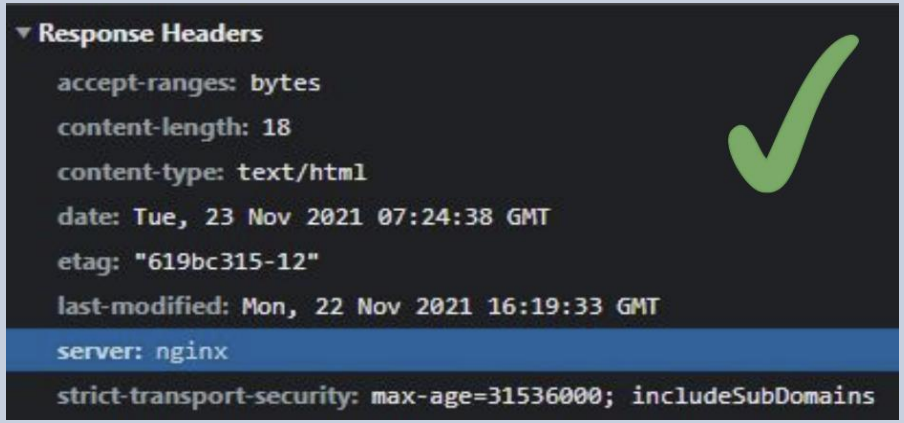
Common: Hide Server Information (3%)

- On virtual host nasa.{ID}.cs.nycu
  - Do not show the server version on error pages.



  - Hide Nginx/Apache version in header.





Common: HTTPS & PKI (12%)

- Redirect HTTP to HTTPS in all virtual host (nasa, file, adminer) domains (2%)
- Enable HTTP Strict Transport Security (HSTS) in all virtual host domains (2%)
- Sign your own SSL Certificate

Common: HTTPS & PKI (12%)

- Create your own CA (certificate authority) and generate A certificate for all virtual hosts
- The CA should be signed with the following arguments (3%)
  - Country name: TW
  - State name: HsinChu
  - Organization name: NYCU
  - CN: {ID}.cs.nycu
- The CA certificate file should be placed at `/home/judge/ca.crt`
- Trust the CA on your system (1%)
- The certificate should be signed from the CA (4%)
  - All fields except CN are the same as the CA
  - Hint: wildcard certificate
- Install openssl tool on your system

Common: Access Control (3%)

- On virtual host `nasa.{ID}.cs.nycu`, when it's accessed, the user is required to provide credentials (HTTP Basic Authentication). (3%)
  - Username: `sa-admin`
  - Password: Your {IP} without dots. (e.g. `101132011`)
- If not, the web server should return "HTTP Unauthorized"

Common: Normal Logging (8%)

- Write the access log of your web server to **`/home/judge/webserver/log/access.log`**

- Log Format (4%)
  - Combined: default in Nginx, needed to be identical when using any other web server
- No logging when request user agent contains “no-logging” (4%)

Common: Verbose Logging (16%)

- Write the verbose log to **`/home/judge/webserver/log/access.log`**, append to the same file of normal logging file (16%)

- Log Format
  - STATUS: `{STATUS}\t{base64 encode result of the log struct}`
  - The structure of log will be shown in next slide
  - One log entity should be written in same line
  - example: **`STATUS: 404 UmVxdWVzdCBIZWFk……`**
  - Hint: https://github.com/openresty/lua-nginx-module

Common: Verbose Logging (16%)

- Logging structure after decode with base64

```
Request Headers:
x-forwarded-for: 127.0.0.1
x-forwarded-proto: https
connection: Keep-Alive
user-agent: Mozilla/5.0 (compatible; Discordbot/2.0; +https://discordapp.com)

Request Body:


Response Headers:
connection: keep-alive
x-frame-options: DENY
content-type: text/html; charset=utf-8
content-language: en-us
content-length: 3636
referrer-policy: same-origin
cross-origin-opener-policy: same-origin
vary: Accept-Language, Cookie, origin
x-content-type-options: nosniff

Response Body:
{"status":"OK"}
```

Common: Verbose Logging (16%)

- Each logging entity need to contains
  - All Request Headers and Response Headers
  - Entire Request Body and Response Body
- Order in each logging entity does not matter
- Print empty line if header or body not exist
- Print entire base64 encode result into access.log file
- Your access.log should look like below but not identical

```
STATUS: 200 UmVxdWVzdCBIZWFkZXJzOgp4LWZvcndhcmRlZC1mb3I6IDM1LjIzNy.......
127.0.0.1 - - [06/Nov/2024:10:42:47 +0000] "GET / HTTP/1.1" 200 524 "https://localhost/" "Mozilla/5.0 (Linux; Android 10; K) AppleWe
STATUS: 404 UmVxdWVzdCBIZWFkZXJzOgp4LWZvcndhcmRlZC1mb3I6IDM1LjIzNy.......
127.0.0.1 - - [06/Nov/2024:10:42:47 +0000] "GET /notfound HTTP/1.1" 404 12 "https://localhost/" "Mozilla/5.0 (Linux; Android 10; K)

```

Common: Log Rotate (5%)

- Rotate your web server's log files (5%)
  - You can use any tool you want to rotate the log
  - Rotate target: **`/home/judge/webserver/log/access.log`**
- Requirements
  - Rotate if file size is greater than 300 bytes
    - If less than 150 bytes, do nothing
  - Preserve only 3 recently rotated files
  - Compress with gzip
  - Store your compressed log to **`/home/judge/webserver/log/compressed.log.[1-3].gz`**

Database (10%)

- Deploy the PostgreSQL service on `192.168.{ID}.1` (6%)
- Set up an user which username is "`root`" and the password is "`sa-hw4-{ID}`" with appropriate permission. (4%)
- Ensure psql (terminal-based PostgreSQL frontend) is installed on the machine `192.168.{ID}.1` and that the user '`judge`' can access to it.
  - You may lose all points if it's not installed
  - Remember to set `PGPASSWORD` system environment variable
- Retrieve the user information query by name from table named `user` (schema below) in database `sa-hw4` when processing a GET request to `/db/{name}`.

Column	Type	Comment
id	integer *Auto Increment* [**nextval('user_id_seq')**]	
name	text	
age	integer	
birthday	date	

file.{ID}.cs.nycu

- Choose whatever backend server implementation you want
- With following api endpoints
  - `GET /ip` (2%)
  - `GET /file/{fileName}` (shared 8%)
  - `POST /upload` (shared 8%)
  - `GET /db/{name}` described in database section (6%)
- The detailed API spec is here
  - You can use https://editor.swagger.io/ for visualizing the API spec

file.{ID}.cs.nycu

- The backend server should be served at port 8080 on BOTH hosts in total two replicas (4%)
- The web server should act as proxy server to distribute the traffic arrived file.{ID}.cs.nycu to the backend server using round robin algorithm with same weight (2%)
  - No points will be given if this failed
  - Note that even though user uploads file when connecting to `192.168.{ID}.1`, the file can be retrieved when connecting to `192.168.{ID}.2`
  - We may shutdown one of your backend server during judge, make sure your web server can route the traffic to the functional one
  - I.e., the files are synced between two hosts

adminer.{ID}.cs.nycu: adminer service (10%)

- Adminer is a full-featured database management tool written in PHP
- Deploy Adminer using any preferred method, ensuring it's accessible through your web server and can establish a connection to the PostgreSQL database server with user “root”. (10%)
- The PostgreSQL database server should be accessible by adminer via the following URI: `192.168.{ID}.1:5432`.
  - Hint: integrating PHP into web server, docker, handcraft

Firewall

國立陽明交通大學資工系資訊中心
Information Technology Center, Department of Computer Science, NYCU

Firewall: General Rules

- ICMP (ping) (3%)
  - Only internal network IP (`192.168.{ID}.0/24`) can send ICMP echo request packets to server. (will NOT respond ICMP ECHO-REPLY packets)
  - I.e., allow ping for `192.168.{ID}.0/24` but deny all other sources
- HTTP (5%)
  - The web server (port 80 & 443) should be accessible from anywhere
  - The file service (port 8080) should only accept traffic from `192.168.{ID}.0/24` and deny all other sources

Firewall: SSH

- If someone attempts to login via SSH but failed for 3 times in 5 minute, their IP will be banned from SSH for 60 seconds automatically. (12%)
  - There are many software can do this, e.g. Blacklisted, DenyHosts, Fail2Ban, etc. (See appendix.)
  - Banned SSH still have access to other services.

NFS

國立陽明交通大學資工系資訊中心
Information Technology Center, Department of Computer Science, NYCU

NFS Server & Client

- You should setup NFS server on `192.168.{ID}.2` (the new host) (2%)
  - It should export `/data` and allow `192.168.{ID}.0/24` to mount (2%)
- Client is `192.168.{ID}.1`
  - Client should mount `/data` to `/net/data` with rw permission (1%)
  - Client should mount it after reboot automatically (2%)
  - Root on client should NOT have full access to the mount directory (3%)

Rules

- TAs reserve the rights of final explanations
- **Open from 11/14 (Thu) 23:59**
- **Deadline: 12/19 (Thu) 23:59**
- Late submissions will **NOT** be accepted

Attention

- Your work will be scored by Online Judge system
  - Only the **LAST** submission will be scored
  - Late submission will **NOT** be accepted
- **ALWAYS BACKUP** your system before submission, as we may do malicious actions
- Make sure everything works after reboot

Tips

- Install your system on virtual machines to benefit from flexibility
  - Easy install and backup
- Try to make your VM hardware configuration better
  - Disk controller
    - IDE → SATA, NVMe, …
  - NIC: paravirtualized net, ...

Appendix:
How to use Online Judge

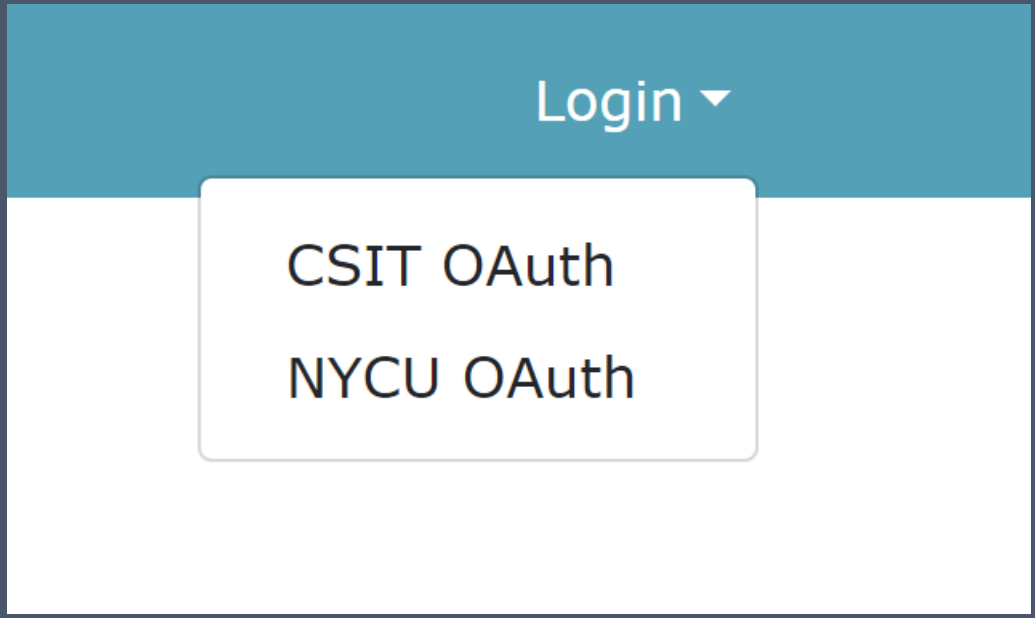
國立陽明交通大學資工系資訊中心
Information Technology Center, Department of Computer Science, NYCU

Online Judge

https://nasaoj.cs.nycu.edu.tw

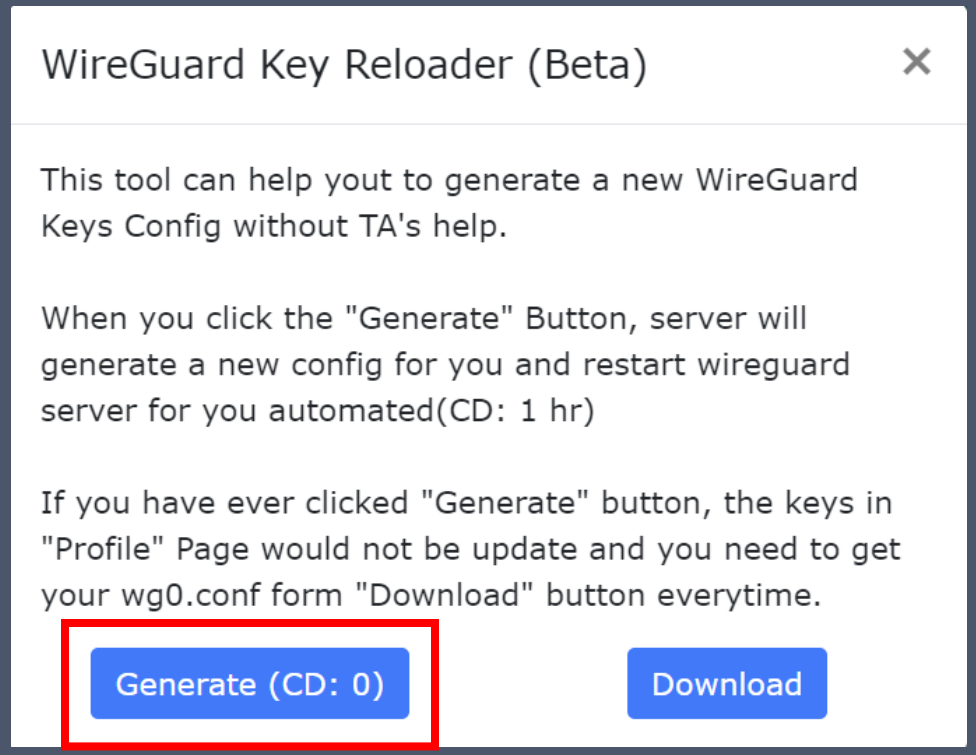
Login

Log in NASA Online Judge via either OAuth methods



Troubleshooting WireGuard® Issues

Use key reloader to regenerate configuration to solve VPN-related issues



Appendix: Blacklistd & DenyHosts

國立陽明交通大學資工系資訊中心
Information Technology Center, Department of Computer Science, NYCU

Appendix: Blacklistd

- Blacklistd is a daemon listening to sockets to receive notifications from other daemons about connection attempts that failed or were successful.
- Since FreeBSD 11 imported blacklistd from NetBSD.
- Enable Blacklistd
  - The main configuration for blacklistd is stored in blacklistd.conf(5).
  - sysrc blacklistd_enable=yes
  - service blacklistd start

Appendix: DenyHosts

- DenyHosts is a utility developed by Phil Schwartz and maintained by a number of developers which aims to thwart sshd (ssh server) brute force attacks.
- Installation
  - /usr/ports/security/denyhosts
  - pkg install denyhosts
- Enable DenyHosts
  - sysrc denyhosts_enable=yes

Help

- Join NCTUNASA google group
  - If you have any question, you can post your problem in this group, TAs and Students will help you.
  - https://groups.google.com/g/nctunasa
- UNIX 常見指令教學
  - https://it.cs.nycu.edu.tw/unix-basic-commands
- How To Ask Questions The Smart Way
  - https://github.com/ryanhanwu/How-To-Ask-Questions-The-Smart-Way