Project 4: SOCKS 4

NP TA 奎廷

12/26 23:55

Project 4 Deadline

Demo: 12/29 Wed.

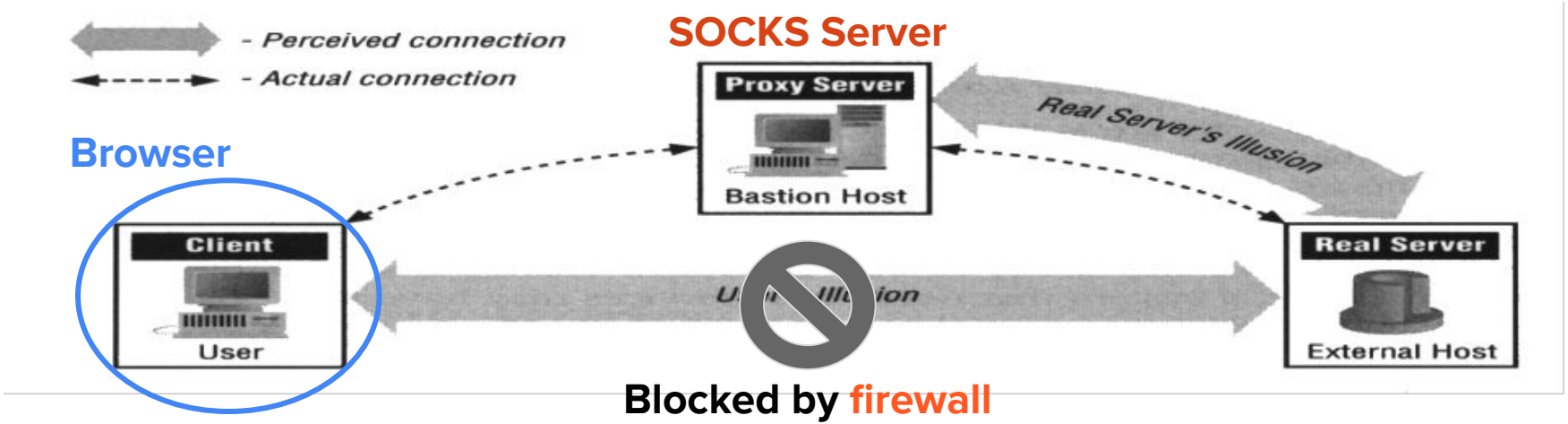
Project Requirements

I. SOCKS 4 Server **Connect** Operation
II. SOCKS 4 Server **Bind** Operation
III. CGI Proxy
IV. Firewall

I. Connect Operation

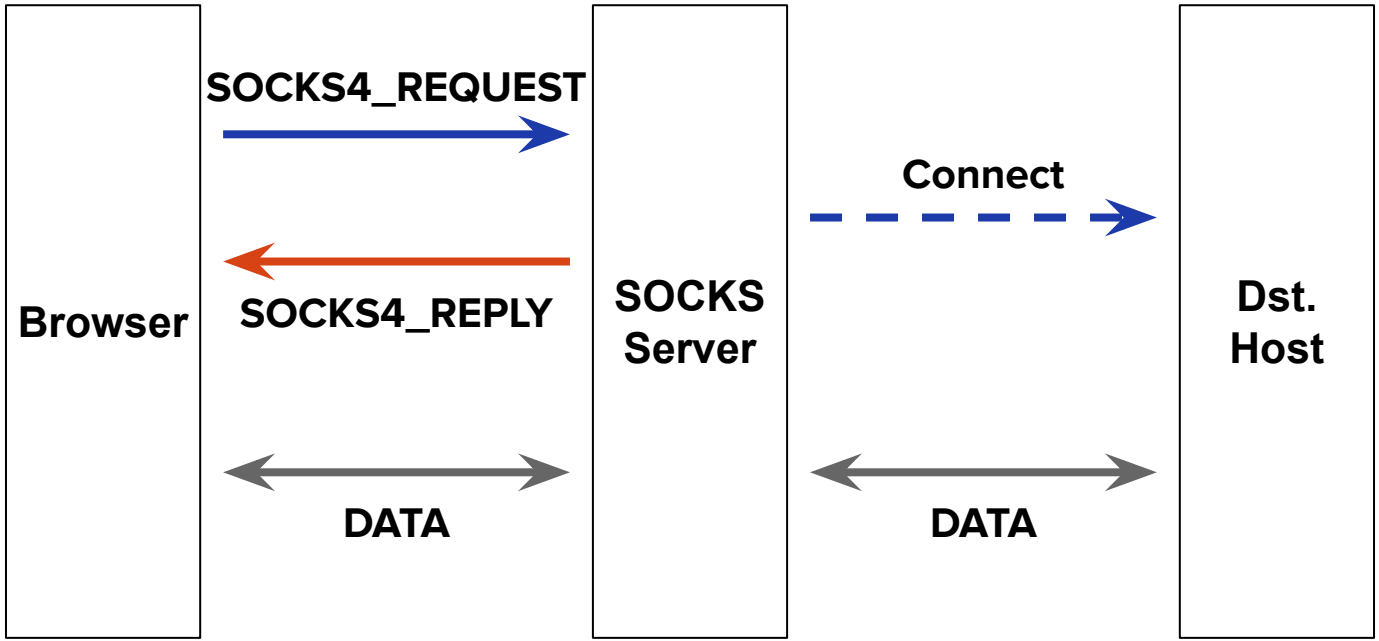
Connect Request

- A client wants to establish a connection to an application server



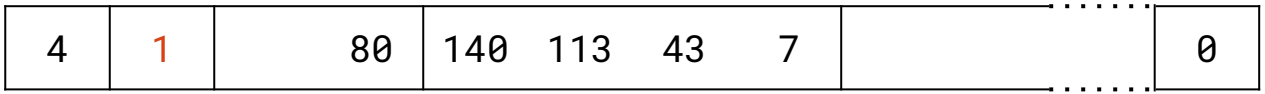
ref. proxy.ppt, p.2

Connect Operation (HTTP Example)



SOCKS4_REQUEST

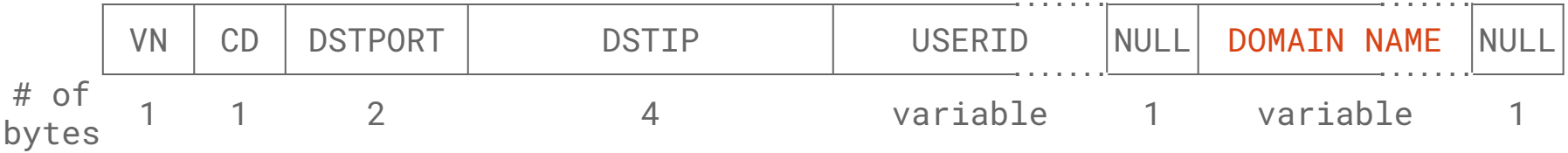
Example:(Connect)



- VN is the SOCKS protocol version number and should be **4**
- CD is the SOCKS command code and should be **1** for **CONNECT** request
- NULL is a byte of all zero bits

SOCKS4_REQUEST (SOCKS 4A)

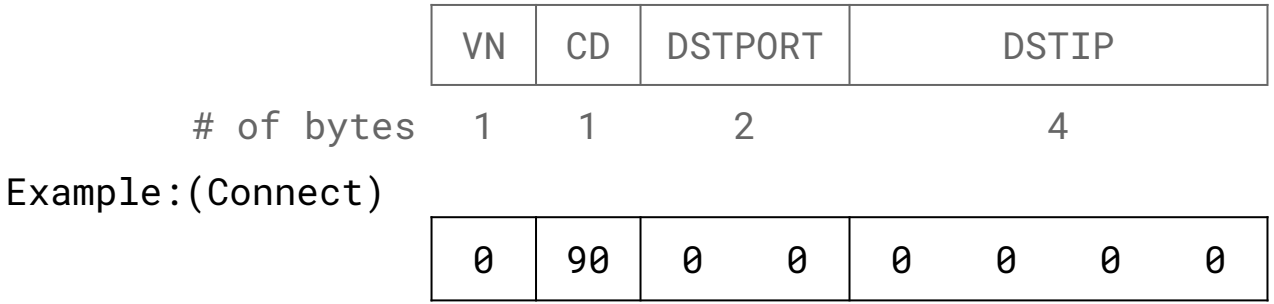
- If the client cannot resolve the destination host's domain name itself



Example:(Connect)

- DSTIP should be `0.0.0.x` with nonzero x
- The SOCKS server resolves the domain name
- You may test with `` `curl --socks4a <host[:port]> <URL>` ``

SOCKS4_REPLY

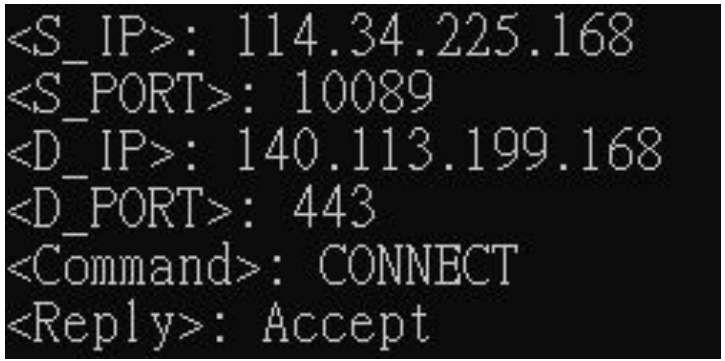


- VN is the version of the reply code and should be **0**
- CD is the result code:
  - **90**: request granted
  - **91**: request rejected or failed
- DSTPORT and DSTIP fields are ignored in CONNECT reply

SOCKS Server Messages

Your server should print messages in the following format:

- `<S_IP>:` source ip
- `<S_PORT>:` source port
- `<D_IP>:` destination ip
- `<D_PORT>:` destination port
- `<Command>: CONNECT or BIND`
- `<Reply>: Accept or Reject`

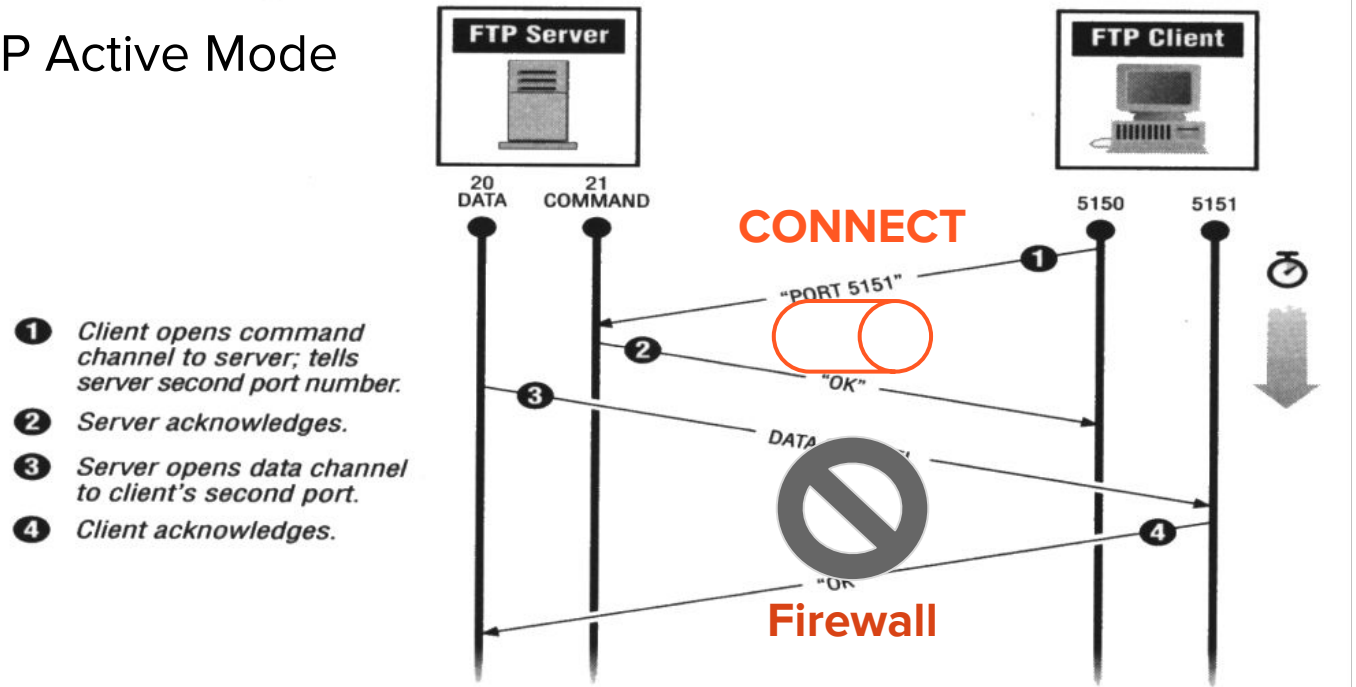


II. Bind Operation

Bind Request

- A client wants to prepare for an **inbound** connection from an application server

FTP Active Mode

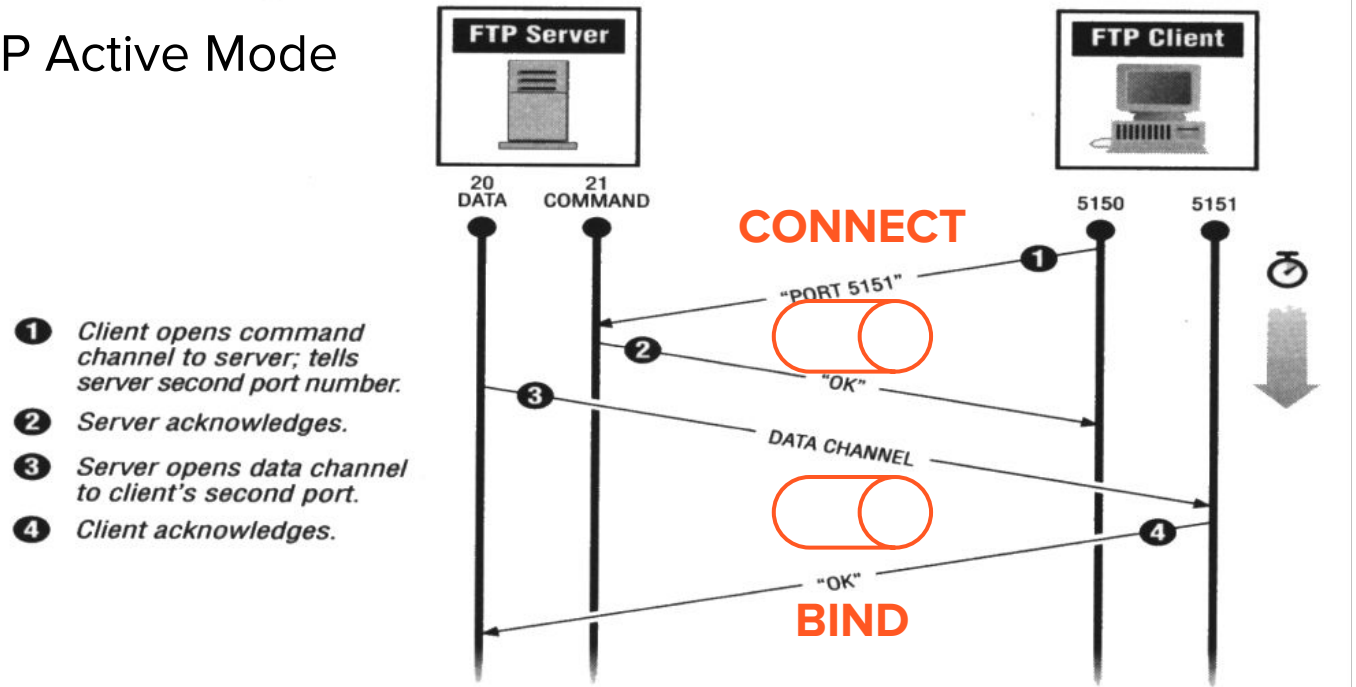


ref. services.ppt, p.13

Bind Request

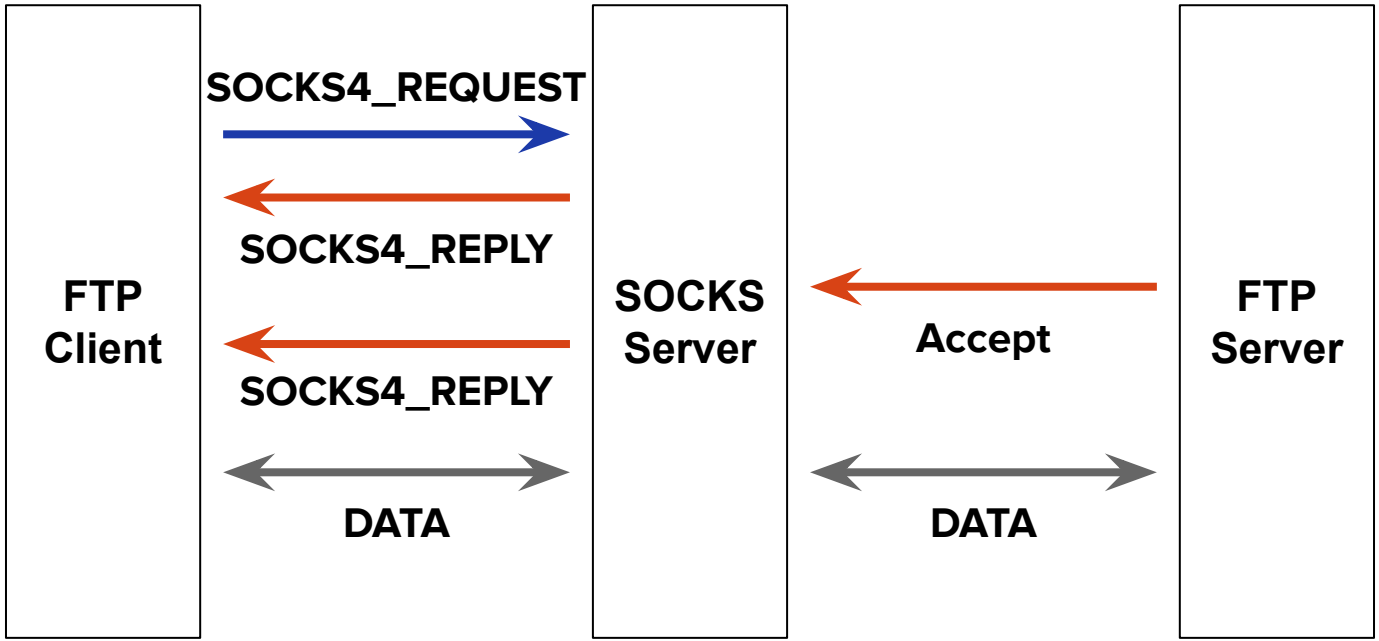
- A client wants to prepare for an **inbound** connection from an application server

FTP Active Mode



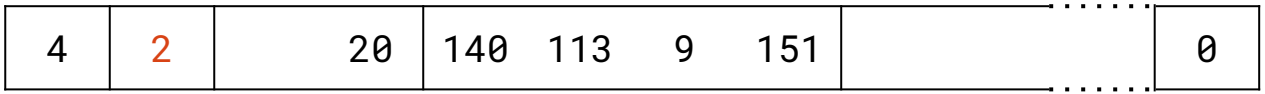
ref. services.ppt, p.13

Bind Operation (FTP Example)



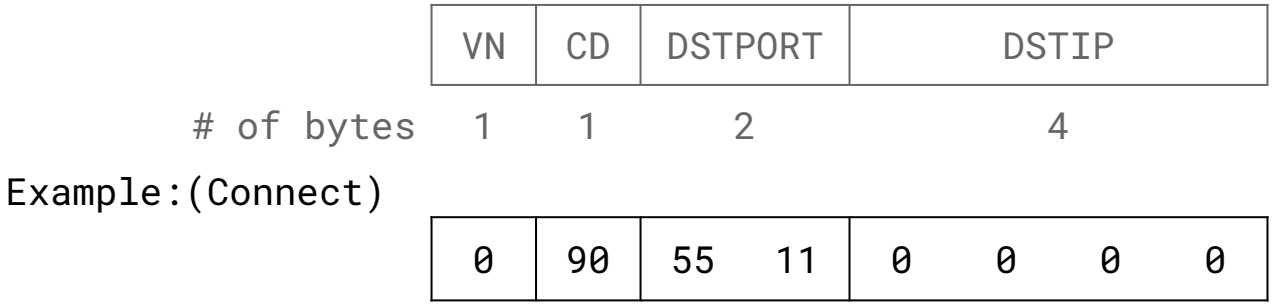
SOCKS4_REQUEST

Example:(Bind)



- VN is the SOCKS protocol version number and should be **4**
- CD is the SOCKS command code and should be **2** for **BIND** request
- NULL is a byte of all zero bits

SOCKS4_REPLY

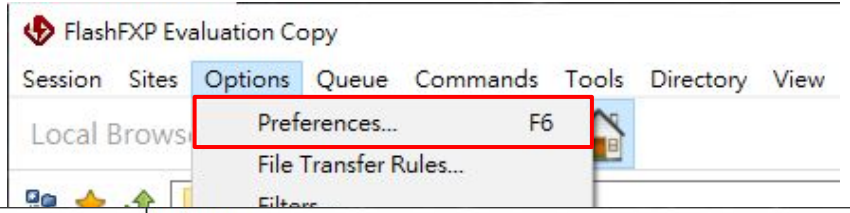


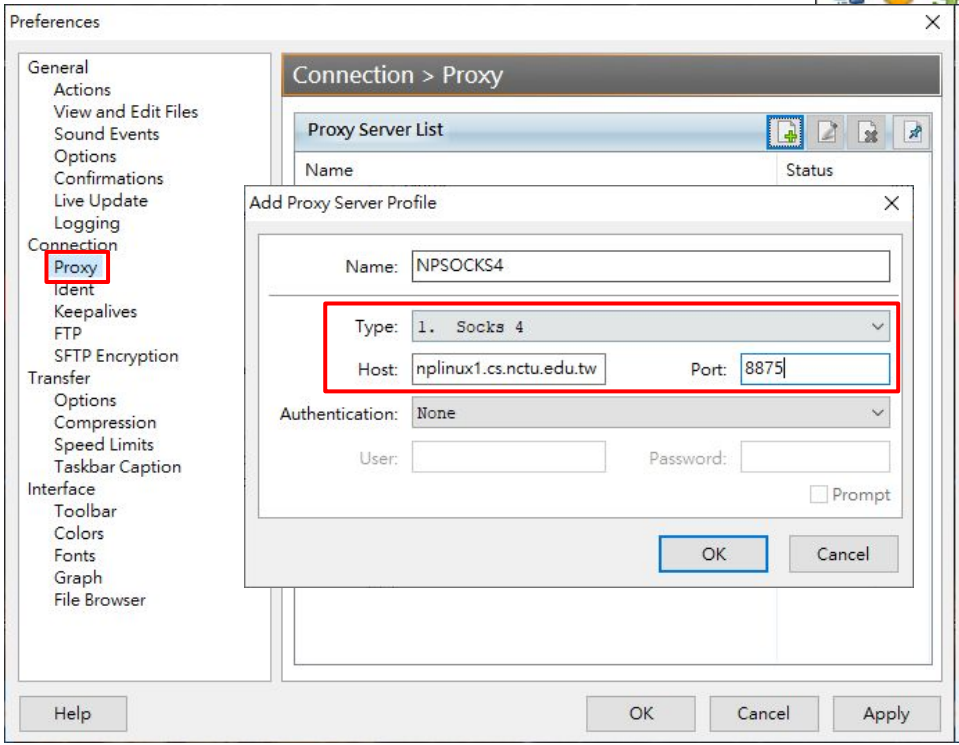
- VN is the version of the reply code and should be **0**
- CD is the result code:
  - **90**: request granted
  - **91**: request rejected or failed
- DSTPORT and DSTIP fields are **meaningful** in BIND reply

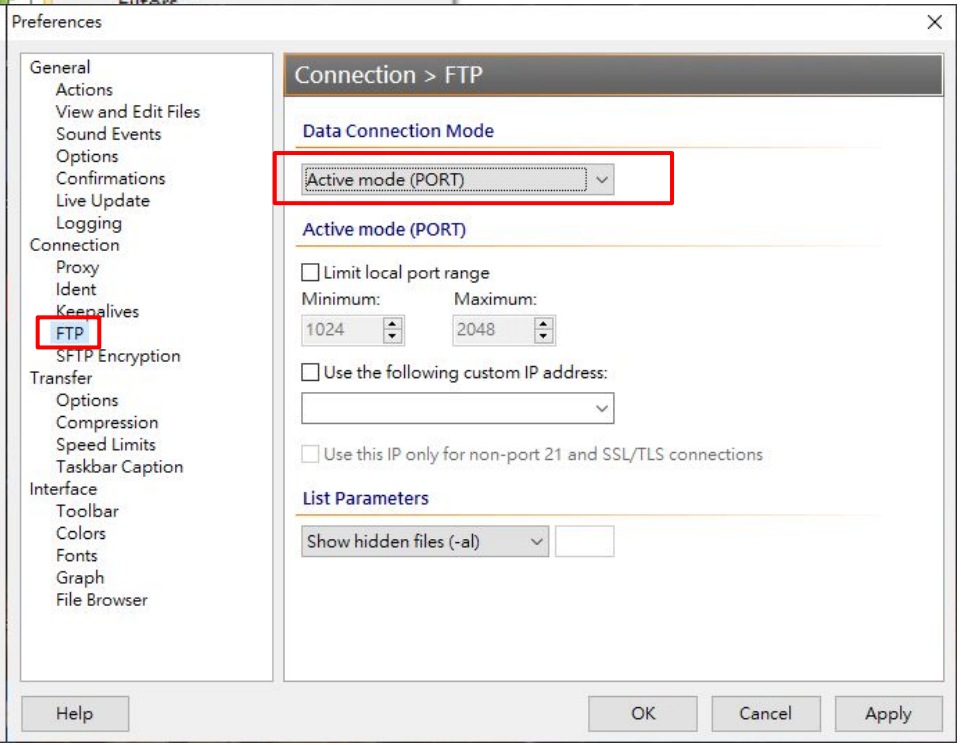
FTP Server / Client

- FTP Server
  - You should setup your own FTP server for testing
  - E.g., FileZilla Server
- FTP Client
  - We will use **FlashFXP** (link) as FTP client
  - The client has to support FTP Active Mode with Proxy on

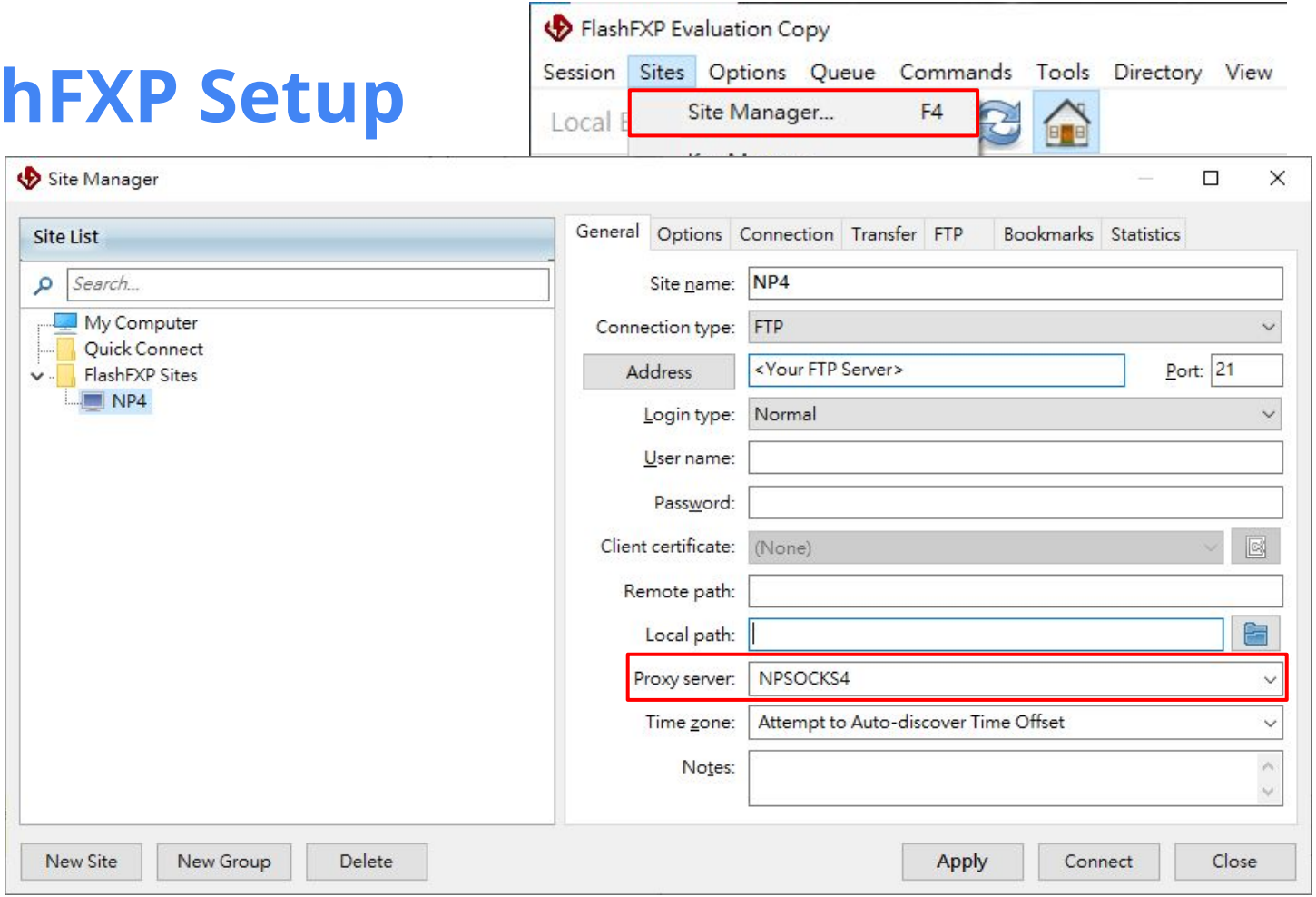
FlashFXP Setup





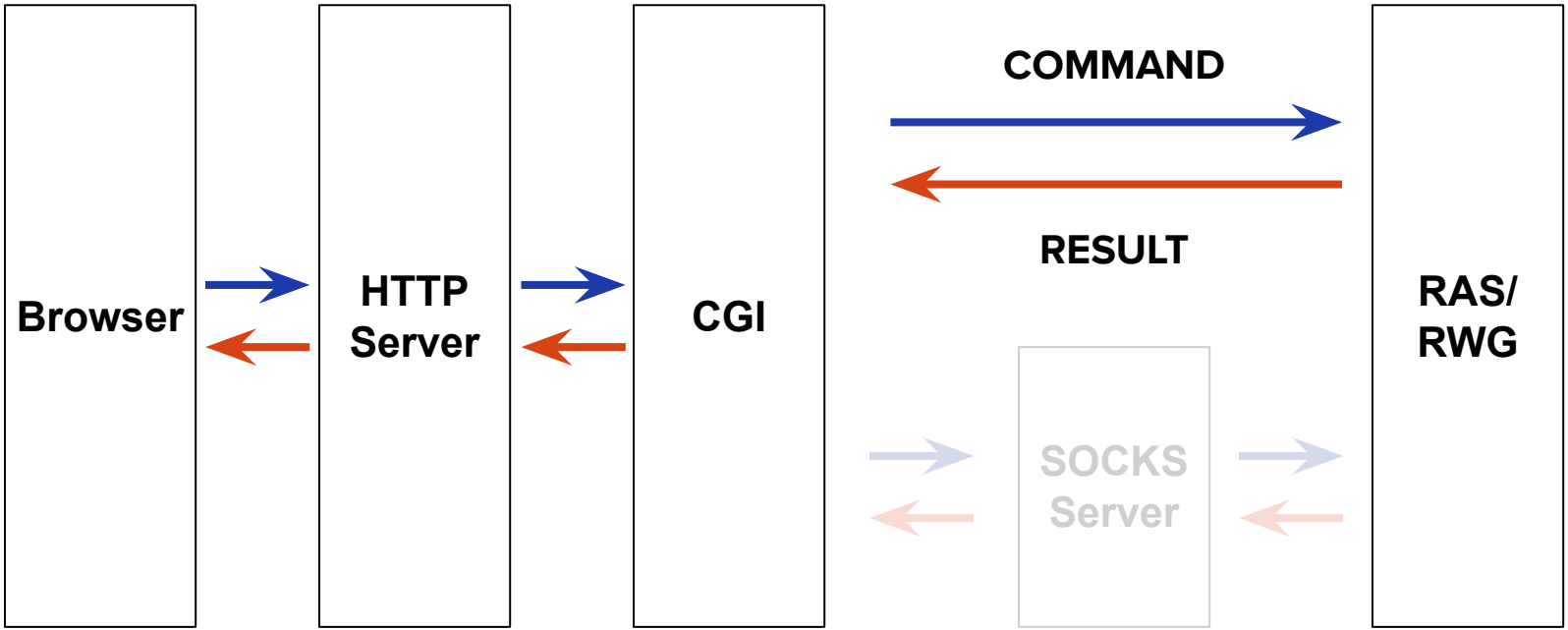


FlashFXP Setup

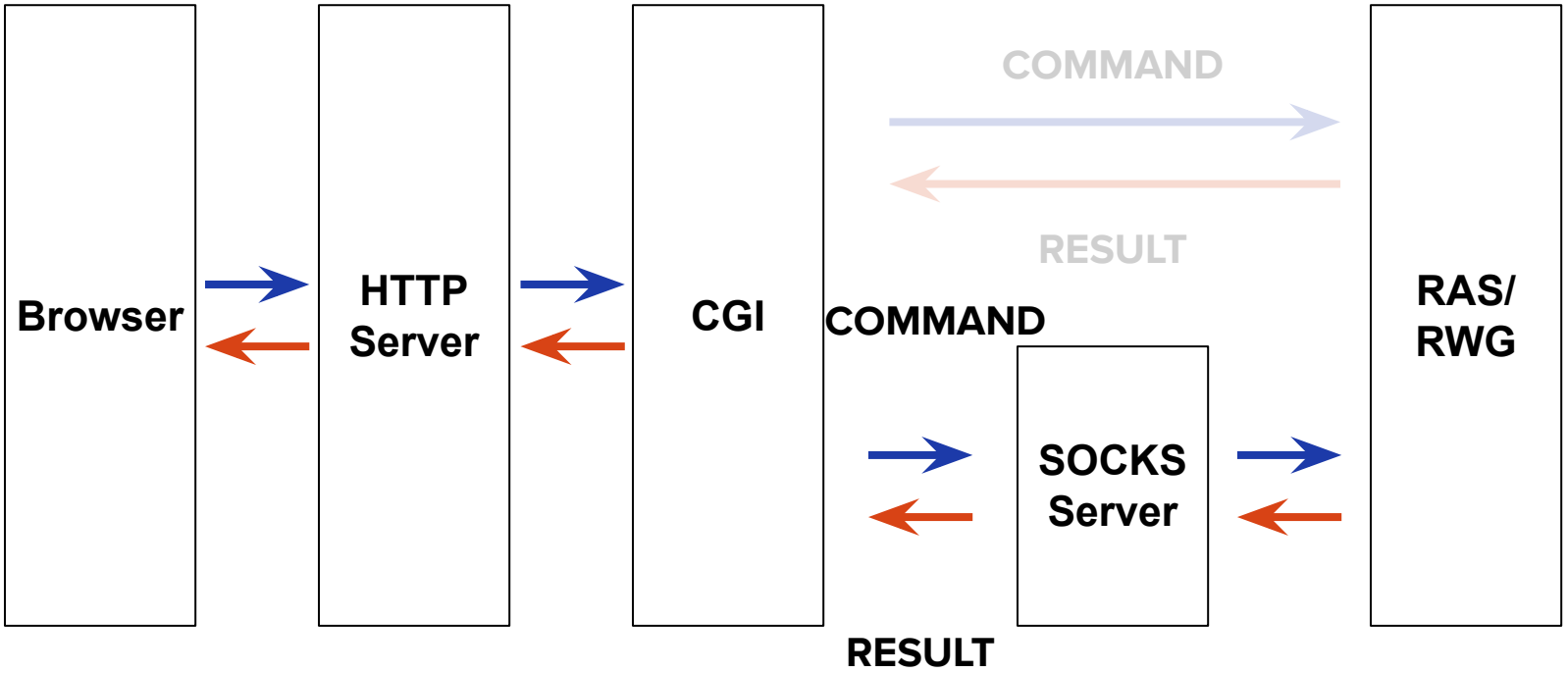


III. CGI Proxy

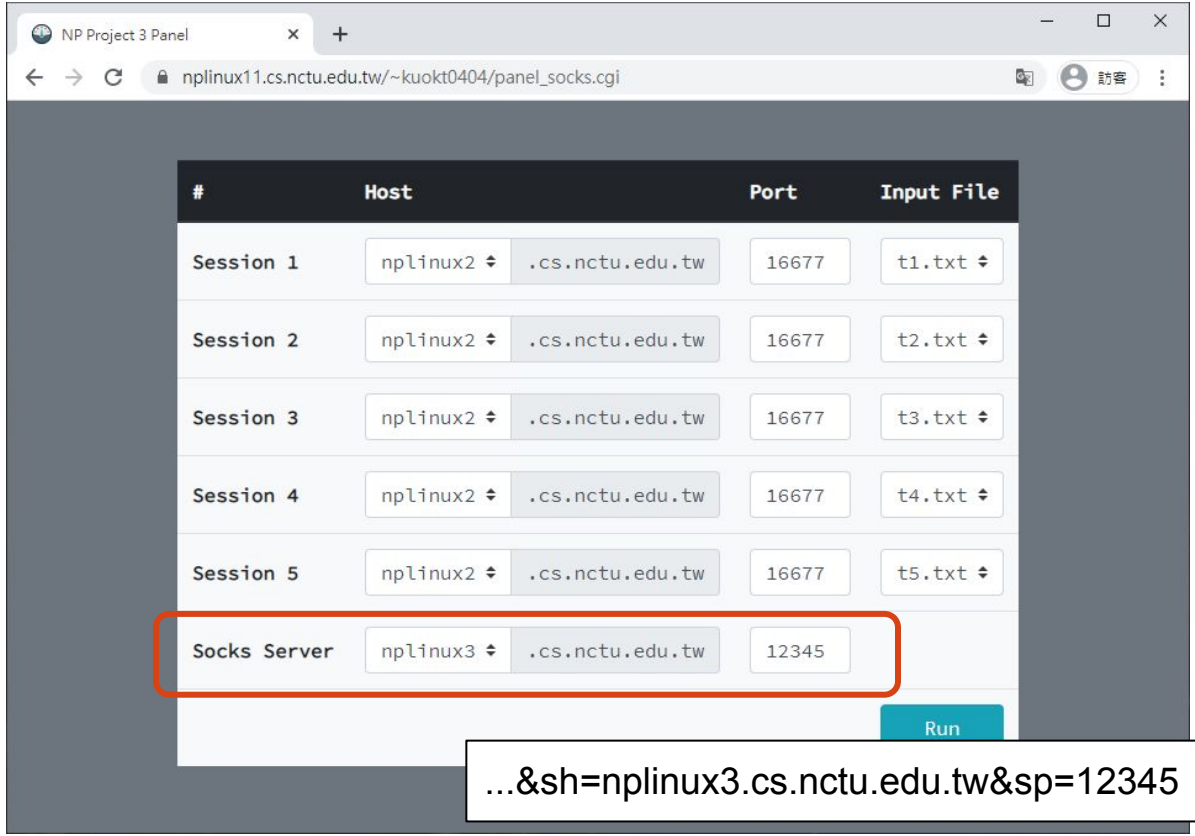
CGI Connection (Project 3)



CGI Connection with SOCKS



panel_socks.cgi

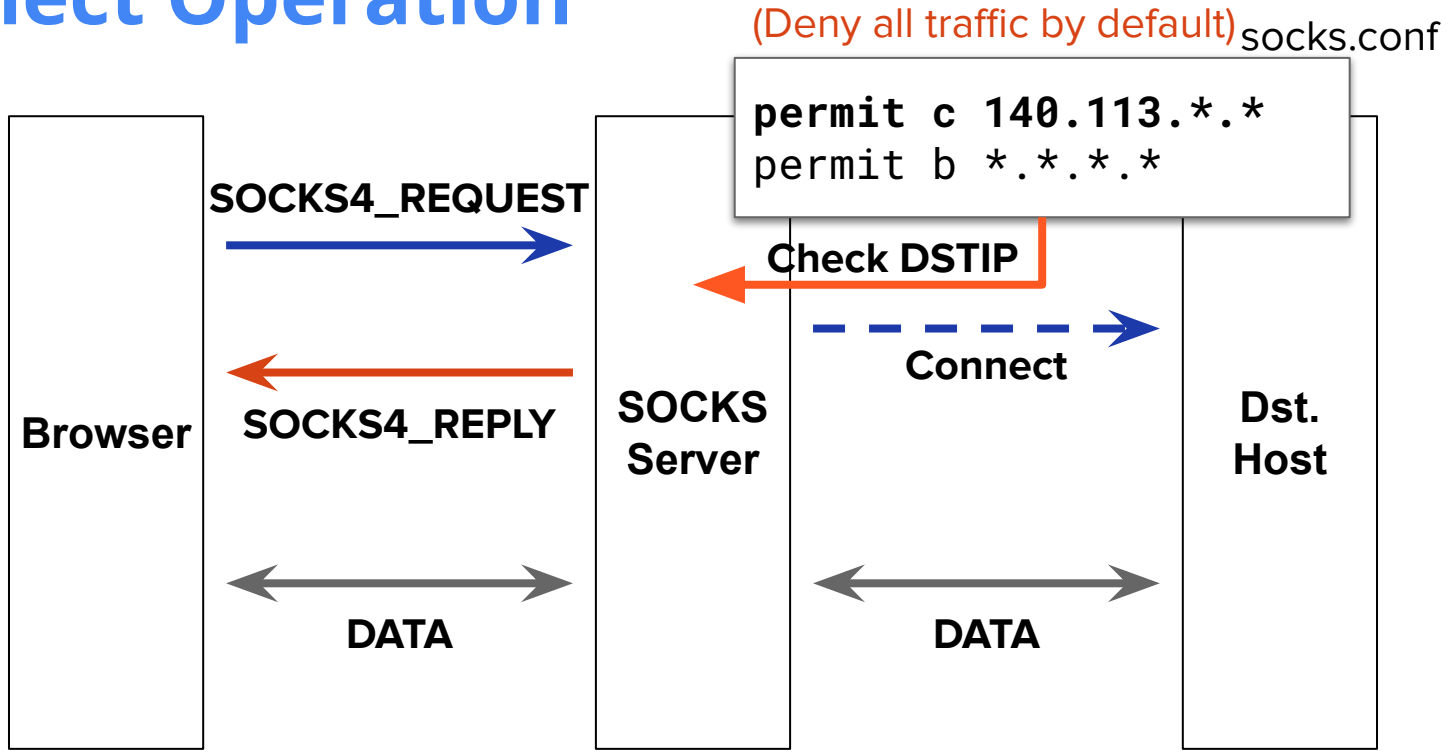


Details

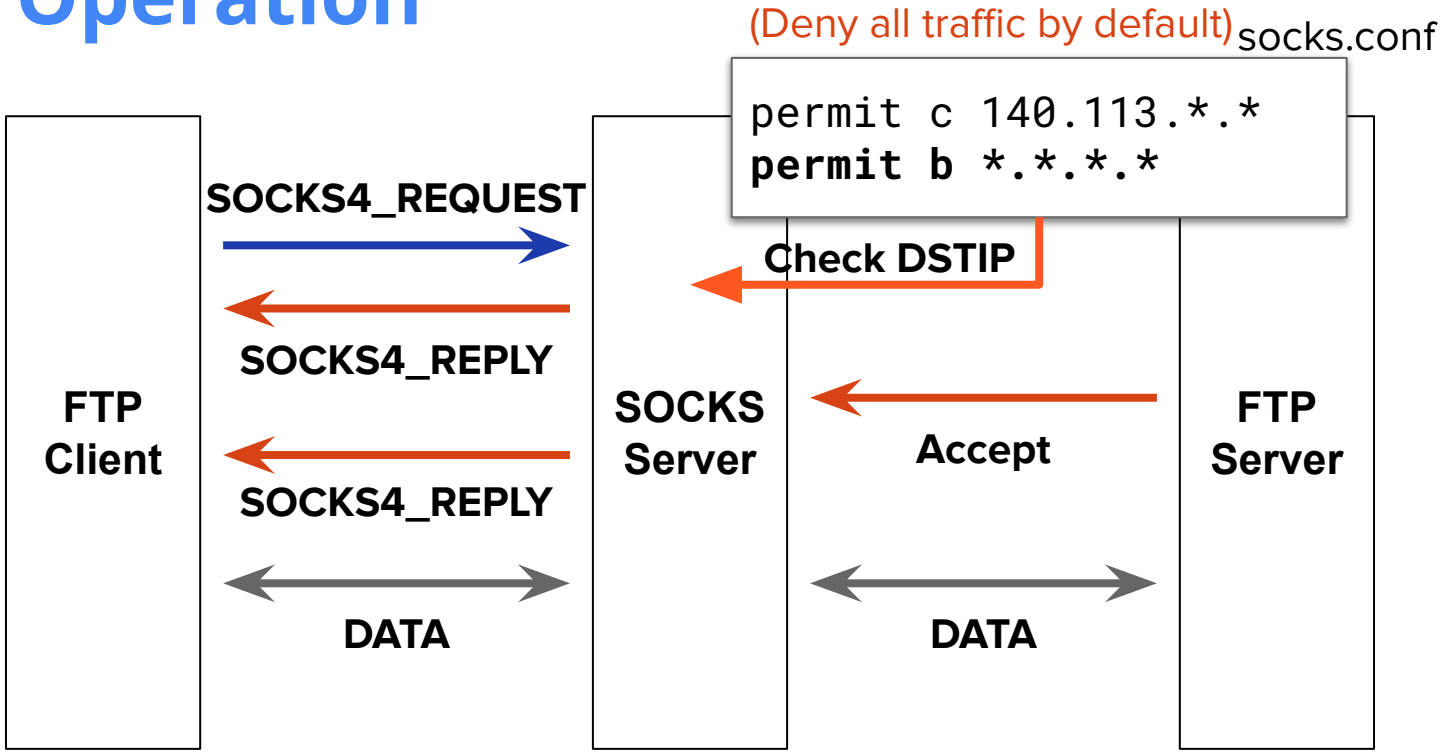
- Modify Project3 *console.cpp* to implement **SOCKS 4 client** (*hw4.cgi*)
  - In `QUERYSTRING`, there will be **`sh=<SocksHost>&sp=<SocksPort>`**
- *panel_socks.cgi* will be provided
- Testing steps
  - Close proxy setting of your browser
  - Put *test_case*, *panel_socks.cgi* and *hw4.cgi* in *~/public_html*
  - Run your **socks server** and ***np_single_golden*** on nplinux
  - Connect and run panel_socks.cgi
    - E.g., nplinux2.cs.nctu.edu.tw/~<yourname>/panel_socks.cgi

IV. Firewall

Connect Operation



Bind Operation



Reference

- SOCKS 4
- SOCKS 4A
- Regular expressions library - cppreference.com

Note

- You are **HIGHLY** encouraged to ask your questions on Teams
- For personal problems, you can mail to all the TAs:
  - lcd010308@gmail.com
  - kuo0404@gmail.com
  - hpc.cs08g@nctu.edu.tw
  - kyojeong.cs10@nycu.edu.tw