

HOMEWORK 3

# File Server & Backup

wkhsiao, yaclu

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

# Outline

- HW 3-1: File server
- HW 3-2: SFTP auditing with RC
- HW 3-3: ZFS & Backup

# HW 3-1: File server (24%)

---

國立陽明交通大學資工系資訊中心

Information Technology Center, Department of Computer Science, NYCU

# HW 3-1: Requirement (1/4)

Use **SFTP** to build a file server; create 2 directories under /home/sftp

1. /home/sftp/public:

- Everyone can **download & upload** files except for **anonymous**
- Everyone can **mkdir** except **anonymous**
- Everyone can only **delete & rmdir** their **own** file or directory
- sysadm can **download, upload, delete, mkdir, rmdir** all content

2. /home/sftp/hidden:

- Create a directory called **“treasure”** inside hidden directory
- Create a file called **“secret”** inside hidden/treasure
- **Anonymous** can't list /home/sftp/hidden but **can enter hidden/treasure** and **show hidden/treasure/secret**
- sysadm can **download, upload, delete, mkdir, rmdir** all content

# HW 3-1: Requirement (2/4)

## Create users

1. Create a **system user** “sysadm”
  - Can log in by **SSH**
  - Full access to public and hidden
2. Create **two users** “sftp-u1”, “sftp-u2”
  - Can **not** log in by **SSH**
  - Can **only** delete files in /home/sftp/public **which are created by themselves**
  - Other permissions are the same as sysadm
3. Create a user “anonymous”
  - Can **not** log in by **SSH**
  - Read-Only permission (enter directory /home/sftp/{public,hidden} and read file)

# HW 3-1: Requirement (3/4)

## Other requirements

- All accounts except `sysadm` are `chrooted` to `/home/sftp`
- Everyone should support login to sftp with ssh key (same public key of judge)
- remaining users, “sftp-u1”, “sftp-u2”, “anonymous”
  - can only be used by `SFTP` (can't login by `SSH`)
  - every uploaded file should remove other's `read/write/execute` DAC permission

# HW 3-1: Requirement (4/4)

	<u>sysadm</u>		<u>sftp-u{1...2}</u>		<u>anonymous</u>	
	<i>public/</i>	<i>hidden/</i>	<i>public/</i>	<i>hidden/</i>	<i>public/</i>	<i>hidden/</i>
list dir	✓	✓	✓	✓	✓	✗
mkdir	✓	✓	✓	✓	✗	✗
rmdir	✓	✓	▲	✓	✗	✗
upload	✓	✓	✓	✓	✗	✗
download	✓	✓	✓	✓	✓	✓
delete	✓	✓	▲	✓	✗	✗

✓ : full access ▲ : only the owner has permission ✗ : permission denied

# HW 3-1: Grading (24%)

- sysadm
  - Login from ssh and sftp (2%)
  - Full access to “public” (2%), “hidden” (2%)
- sftp-u1, sftp-u2
  - disable SSH login, only accept SFTP, Chrooted (/home/sftp)(3%)
  - Full access to “public”, can only delete files and directories they owned. (2%)
  - Full access to “hidden” (2%)
  - adjust DAC (2%)
    - remove all permission (rwx) of others when uploading
- anonymous
  - disable SSH login, only accept SFTP, Chrooted (/home/sftp) (3%)
  - can enter “hidden” (2%) and “public” (2%)
  - operations are read-only(even the file is writable to anonymous) (2%)



# HW 3-1: Hint

- README (sftp config)
  - [sshd\\_config](#)
  - [sftp-server](#)
- If `ssh` or `sftp` run unexpectedly
  - Check your ssh log `/var/log/auth.log` first

# HW 3-2: SFTP auditing with RC (22%)

---

國立陽明交通大學資工系資訊中心

Information Technology Center, Department of Computer Science, NYCU

# HW 3-2: Requirements (1/6)

- Enable SFTP logging, aggregate all SFTP log to “`/var/log/sftp.log`”
  - SFTP log should only contain pure SFTP log, can't blend with other log (SSH, sudo...)

```
judge@freebsd-141:~ $ sudo cat /var/log/sftp.log
Oct 9 23:48:35 freebsd-141 internal-sftp[48981]: session opened for local user sftp-u1 from
[10.113.52.12]
Oct 9 23:48:35 freebsd-141 internal-sftp[48981]: open "/public/test.exe" flags
WRITE,CREATE,TRUNCATE mode 0666
Oct 9 23:48:35 freebsd-141 internal-sftp[48981]: set "/public/test.exe" size 0
Oct 9 23:48:35 freebsd-141 internal-sftp[48981]: set "/public/test.exe" modtime 20241008-15:11:01
Oct 9 23:48:35 freebsd-141 internal-sftp[48981]: close "/public/test.exe" bytes read 0 written 0
Oct 9 23:48:40 freebsd-141 internal-sftp[48981]: session closed for local user sftp-u1 from
[10.113.52.12]
```

# HW 3-2: Requirements (2/6)

- Create an executable stand-alone program (called “`sftp_watchd`”) that would filter every file uploaded.
  - “`sftp_watchd`” should reside in your system’s PATH
  - “`sftp_watchd`” can be written at **any** language (Python, Lua, Rust...)
  - The **executable files** are **violated**
  - **Executable files** might **not** end with a `.exe` extension. You need to try other ways to check the file type.
  - Move these files to `/home/sftp/hidden/.violated/`

```
judge@freebsd-141:~ $ sudo ls /home/sftp/hidden/.violated/  
test1.exe, program
```

# HW 3-2: Requirements (3/6)

- Log violation of our sftp\_watchd program policy into `/var/log/sftp_watchd.log`

- Format -

**timestamp hostname program\_name: filename** violate file detected.

Uploaded by **upload\_user**.

```
judge@freebsd-141:~ $ sudo cat /var/log/sftp_watchd.log
Oct 9 17:47:25 freebsd-141 sftp_watchd[3256]: /usr/home/sftp/public/test.exe violate file detected. Uploaded by sysadm.
Oct 9 17:47:25 freebsd-141 sftp_watchd[3256]: /usr/home/sftp/public/test.exe violate file detected. Uploaded by sysadm.
```

# HW 3-2: Requirements (4/6)

- You should write an **rc script** “**sftp\_watchd**” as a daemon to start the sftp\_watchd program
- Your service must support these operation:
  - `$ service sftp_watchd start`
  - `$ service sftp_watchd stop`
  - `$ service sftp_watchd restart`
  - `$ service sftp_watchd status`

# HW 3-2: Requirements (5/6)

- Requires a **pid file** to indicate which process to stop

```
judge@freebsd-141:~ $ cat /var/run/sftp_watchd.pid  
3209
```

- You should display as following format while using each command
  - Service start

```
judge@freebsd-141:~ $ sudo service sftp_watchd start  
Starting sftp_watchd.
```

## Service stop

```
judge@freebsd-141:~ $ sudo service sftp_watchd stop  
Kill: 3209
```

# HW 3-2: Requirements (6/6)

- Service restart

```
judge@freebsd-141:~ $ sudo service sftp_watchd restart  
Kill: 3204  
Starting sftp_watchd.
```

- Service status

```
judge@freebsd-141:~ $ sudo service sftp_watchd status  
sftp_watchd is running as pid 3204.
```



# HW 3-2: Grading (22%)

- sftp\_watchd
  - SFTP logging (3%)
  - aggregate only SFTP log to “/var/log/sftp.log” (3%)
  - violation file should moved to /home/sftp/hidden/.exe/ (4%)
  - logging after the violation file upload (4%)
- Service operation works correctly
  - sftp\_watchd should be auto-start (2%)
  - start/status/stop/restart (6%)
    - sftp\_watchd should be run in the background, and pid file is **not required** when using Linux

# HW 3-2: Hint

- [sftp-server\(8\)](#)
  - On some systems, sftp-server **must be able to access /dev/log** for logging to work, and use of sftp-server in a chroot configuration therefore requires that syslogd(8) establish a **logging socket inside the chroot directory** .
- [syslogd\(8\)](#)
  - if log files didn't get the logs, try to restart syslogd
  - [logger\(1\)](#)
- [daemon\(8\)](#)
- [nohup\(1\)](#)
- [file\(1\)](#)

# HW 3-3: ZFS & Backup (54%)

---

國立陽明交通大學資工系資訊中心

Information Technology Center, Department of Computer Science, NYCU

# HW 3-3: Requirement (1/8)

- Add four new hard disks and create a **raid10** pool called “**mypool**”
  - You should partition each disk with GPT partition scheme, and label
  - it as “mypool-1”, “mypool-2”, “mypool-3”, “mypool-4”
  - initialize ZFS pool using vdev with **GPT label** (under “/dev/gpt”)
  - Mount mypool on /home/sftp
- Enable ZFS service
  - Reboot and everything is fine (ZFS still mounted)
- Create ZFS datasets
  - Set **lz4 compression, atime=off** to all datasets
  - Create **mypool/public, mypool/hidden dataset**

# HW 3-3: Requirement (2/8)

- Automatic Snapshot Script: **zfsbak**
  - Add your script to \$PATH
    - Allow to execute zfsbak with command “zfsbak”, not “./zfsbak”
  - Usage:
    - Create: `zfsbak DATASET [ROTATION_CNT]`
    - List: `zfsbak -l|--list [DATASET|ID|DATASET ID...]`
    - Delete: `zfsbak -d|--delete [DATASET|ID|DATASET ID...]`
    - Export: `zfsbak -e|--export DATASET [ID]`
    - Import: `zfsbak -i|--import FILENAME DATASET`

```
judge@freebsd-141:~$ zfsbak
Usage:
- create: zfsbak DATASET [ROTATION_CNT]
- list: zfsbak -l|--list [DATASET|ID|DATASET ID...]
- delete: zfsbak -d|--delete [DATASET|ID|DATASET ID...]
- export: zfsbak -e|--export DATASET [ID]
- import: zfsbak -i|--import FILENAME DATASET
```

# HW 3-3: Requirement (3/8)

- Specification - Create (Default)
  - Must specify **dataset**
  - If no rotation count is specified, use 12 as default
  - No more than rotation count snapshots per dataset
  - If rotation count is reached, delete the oldest one
  - Your snapshot should include the dataset name and date
  - Every snapshot should **prefix with “zfsbak\_”** to avoid collision with other on-demand snapshot

```
judge@freebsd-141:~$ sudo zfsbak -l
ID      DATASET      TIME
judge@freebsd-141:~$ sudo zfsbak mypool/public
Snap mypool/public@zfsbak_2024-10-09-16:22:25
judge@freebsd-141:~$ sudo zfsbak mypool/public
Snap mypool/public@zfsbak_2024-10-09-16:22:32
judge@freebsd-141:~$ sudo zfsbak mypool/public 1
Snap mypool/public@zfsbak_2024-10-09-16:22:38
Destroy mypool/public@zfsbak_2024-10-09-16:22:25
Destroy mypool/public@zfsbak_2024-10-09-16:22:32
```

# HW 3-3: Requirement (4/8)

- Specification - List

- List snapshots created by zfs. **Sorted by time.**
- Ignored the snapshot that doesn't have the prefix **“zfsbak\_”**
- If only **ID** is specified, list only the snapshot with that **id**
- If only **DATASET** is specified, list all snapshots of that dataset
- If **DATASET** and **ID** are specified, list only the snapshot with that **id** of the **dataset**
- Otherwise, list all snapshots

```
judge@freebsd-141:~ $ sudo zfs create
mypool/public@not_zfsbak_target
judge@freebsd-141:~ $ sudo zfsbak -l
ID      DATASET      TIME
1       mypool/public 2024-10-09-16:22:38
2       mypool/public 2024-10-09-16:24:22
3       mypool/hidden 2024-10-09-16:24:28
4       mypool/hidden 2024-10-09-16:24:30
```

```
judge@freebsd-141:~ $ sudo zfsbak -l 3
ID      DATASET      TIME
3       mypool/hidden 2024-10-09-16:24:28
judge@freebsd-141:~ $ sudo zfsbak -l mypool/public
ID      DATASET      TIME
1       mypool/public 2024-10-09-16:22:38
2       mypool/public 2024-10-09-16:24:22
judge@freebsd-141:~ $ sudo zfsbak -l mypool/public 2
ID      DATASET      TIME
2       mypool/public 2024-10-09-16:24:22
```

# HW 3-3: Requirement (5/8)

- Specification - Delete
  - Delete snapshots created by zfs
  - If only **ID** is specified, delete the snapshot with that **id**
  - If only **DATASET** is specified, delete all snapshots of that dataset
  - If **DATASET** and **ID...** are specified, delete snapshots with those **id** of the **dataset**
  - Otherwise, delete all snapshots

```
judge@freebsd-141:~ $ sudo zfsbak -l
ID    DATASET    TIME
1     mypool/public  2024-10-09-16:32:30
2     mypool/hidden 2024-10-09-16:32:34
3     mypool/public  2024-10-09-16:32:36
4     mypool/hidden 2024-10-09-16:32:37
5     mypool/public  2024-10-09-16:32:38
6     mypool/public  2024-10-09-16:32:40
7     mypool/hidden 2024-10-09-16:32:41
```

```
judge@freebsd-141:~ $ sudo zfsbak -d 1
Destroy mypool/public@zfsbak_2024-10-09-16:32:30
judge@freebsd-141:~ $ sudo zfsbak -d mypool/hidden 2
Destroy mypool/hidden@zfsbak_2024-10-09-16:32:37
judge@freebsd-141:~ $ sudo zfsbak -d mypool/hidden
Destroy mypool/hidden@zfsbak_2024-10-09-16:32:34
Destroy mypool/hidden@zfsbak_2024-10-09-16:32:41
judge@freebsd-141:~ $ sudo zfsbak -d mypool/public 1 2 3
Destroy mypool/public@zfsbak_2024-10-09-16:32:36
Destroy mypool/public@zfsbak_2024-10-09-16:32:38
Destroy mypool/public@zfsbak_2024-10-09-16:32:4
```



# HW 3-3: Requirement (6/8)

- Log
  - Must contain the action (e.g. snap), dataset name and time
    - Print “**Snap `dataset@zfsbak\_create\_time`**” after creating the new snapshot, e.g.,
      - Snap mypool/public@zfsbak\_2024-10-09-16:32:30
    - Print “**Destroy `dataset@zfsbak\_create\_time`**” after destroying the deleted snapshot, e.g.,
      - mypool/public@zfsbak\_2024-10-09-16:32:30
  - For any undefined operation, just print the error message and exit

# HW 3-3: Requirement (7/8)

- Specification - Export
  - Must specify **dataset**
  - **ID** defaults to **1**
  - Compress with **zstd**
  - Encrypt with **aes-256-cbc (with password-based key derivation function 2)**
    - Encrypt with the environment we specified (EXPORT\_PASS)
  - A filename example: ``mypool_public@zfsbak_2024-10-09-17:29:56.zst.aes``
  - Put the export file at the user's home directory

```
judge@freebsd-141:~$ export ZFSBAK_PASS=secure_password
judge@freebsd-141:~$ sudo -E zfsbak -e mypool/public 1
Export mypool/public@zfsbak_2024-10-09-17:29:56 to ~/mypool_public@zfsbak_2024-10-09-17:29:56.zst.aes
```

# HW 3-3: Requirement (8/8)

- Specification - Import
  - Must specify **filename** and **dataset**
  - **filename** is the decrypted file exported by zfsbak
  - Load the snapshot to the dataset

```
judge@freebsd-141:~ $ sudo zfsbak -i "~/mypool_public@zfsbak_2024-10-09-17:29:56.zst"
mypool/public2
Import /home/judge/mypool_public@zfsbak_2024-10-09-17:29:56.zst to mypool/public2
judge@freebsd-141:~ $ zfsbak -l
ID      DATASET      TIME
1       mypool/public 2024-10-09-17:29:56
2       mypool/public2 2024-10-09-17:29:56
judge@freebsd-141:~ $ ls /home/sftp/
dev/ hidden/ public/ public2/
```

# HW 3-3: Grading (54%)

- Disk Setup (Add 4 new disks)
  - Enable kernel to show gpt label in /dev/gpt/ (FreeBSD), /dev/disk/by-partlabel (Linux) (3%)
  - partition with GPT scheme with correct label (2%)
- ZFS
  - Create a raid10 pool using block device at /dev/gpt as vdev (3%)
  - Create all datasets and set up correctly mountpoint, atime, compression (3%)
- zfsbak
  - Usage (2%)
  - Create, List, Delete (9% / each)
  - Export, Import (include log) (7% / each)

# HW 3-3: Hint

- It will be much easier if you implement `Delete`, `Export`, `Import` with a well coding `List`
- If you thinks shell script is hard to implement the function we wants, try [awk\(1\)](#)
- Check handbook first
  - <https://www.freebsd.org/doc/en/books/handbook/zfs-zfs.html>
  - <https://www.freebsd.org/doc/en/books/handbook/zfs-term.html>

# HW 3: Grading

- You can choose whatever OS you want to use
  - OJ does **NOT** guarantee **Linux** can pass all test cases, so we only make it work with our **best effort**

# Attention!

- Your work will be tested by Online Judge system.
  - You can submit multiple judge requests. However, OJ will cool down for several minutes after each judge.
  - **We will take the last submitted score instead of the highest score.**
  - Late submissions will not be accepted.
  - Plagiarism is prohibited. We will conduct random checks to compare your scripts with others.
- **BACKUP your server before judge EVERY TIME**
  - **We may do something bad when judging. (e.g. `rm -rf --no-preserve-root /`)**
- Make sure everything is fine after reboot.

# Help me!

## Questions about this homework

- Ask them on <https://groups.google.com/g/nctunasa>
- We MIGHT give out hints on Google Groups
  - Be sure to join the group :D
  - When posting a question, be sure to include all information you think others would need
    - including but not limiting to your ID, setups, configurations and / or what you have done to trace the error / problem
- Do not email us
- Do not use E3 to email us



Enjoy!