Homework 3
File server & Backup

zyyang, shfchen

國立陽明交通大學資工系資訊中心

Computer Center of Department of Computer Science, NYCU

Outline

- HW 3-1: File server
- HW 3-2: Pure-ftpd uploadscript with RC
- HW 3-3: ZFS & Backup

HW 3-1: File server (50%)

HW 3-1: Requirement (1/4)

Use **pure-ftpd** to build a file server; create 3 directories under */home/**ftp***

1. /home/ftp/***public***:
   - Everyone can download & upload file
   - Everyone can mkdir, rmdir, delete except anonymous
2. /home/ftp/***upload***:
   - Everyone can upload & download
   - Everyone can mkdir except anonymous
   - Everyone can only delete & rmdir their own file or directory except anonymous and sysadm
3. /home/ftp/***hidden***:
   - Create a directory called “treasure” inside hidden
   - Create a file called “secret” inside *hidden/**treasure***
   - Anonymous can’t list */home/ftp/hidden* but can enter *hidden/treasure* and show *hidden/treasure/**secret***

HW 3-1: Requirement (2/4)

Create users

1. Create a system user "sysadm"
   - Can login by SSH
   - Password is your IP without dots, e.g. password=1011301 when IP=10.113.0.1
   - Full access to */home/ftp* and subdirectories under "ftp"
2. Create two virtual users "ftp-vip1", "ftp-vip2"
   - Password is your IP without dots, e.g. password=1011301 when IP=10.113.0.1
   - Can only delete files in */home/ftp/**upload*** which are created by themselves
   - Other permissions are same as sysadm
3. Anonymous with no password
   - Can't create any directories, and can't delete any files & directories
   - Can't list */home/ftp/hidden* but can enter *hidden/treasure* and show *hidden/treasure/**secret***

HW 3-1: Requirement (3/4)

Other requirements

- Your ftp server should support Explicit FTP over TLS (FTPES)
  - Should refuse connections that don't use the TLS security mechanism.
- All accounts except sysadm are chrooted (***/home/ftp*** is the root directory)

HW 3-1: Requirement (4/4)

	sysadm		**ftp-vip**		**anonymous**	
	public/	*upload/*	*public/*	*upload/*	*public/*	*upload/*
upload	✔	✔	✔	✔	✔	✔
download	✔	✔	✔	✔	✔	✔
mkdir	✔	✔	✔	✔	✘	✘
rmdir	✔	✔	✔	▲	✘	✘
delete	✔	✔	✔	▲	✘	✘

✔: full access ▲: only the owner has permission ✘: permission denied

HW 3-1: Grading (35/50%)

- FTP over TLS (5%)
- sysadm
  - Login from ssh (4%)
  - Full access to “***public***” (3%), “***upload***” (4%), “***hidden***” (4%)
- ftp-vip1, ftp-vip2
  - Chrooted (***/home/ftp***) (4%)
  - Full access to “***public***” (3%), “***hidden***” (4%)
  - Full access to “***upload***”, but can only delete their own files and directories. (4%)

HW 3-1: Grading (15/50%)

- Anonymous login
  - Chrooted (***/home/ftp***) (4%)
  - Can only upload and download from “***public***” (3%)
  - Can only upload and download from “***upload***” (4%)
  - Hidden directory “***/home/ftp/hidden***” problem:

    can enter but can’t retrieve directory listing (4%)

HW 3-1: Hint

- README
  - */usr/local/share/doc/pure-ftpd/**
- Accounts related
  - Virtual user
  - pure-pw(8)
  - pure-pwconvert(8)
  - README.Virtual-Users
- If `pure-ftpd` is not working
  - Check your pure-ftpd.conf

HW 3-2:
Pure-ftpd uploadscript with RC (15%)

HW 3-2: Requirements (1/5)

- Create an “uploadscript.sh” for filtering every file uploaded.
  - File name with extension .exe is violated
  - Move these files to /home/ftp/hidden/.exe/

```
[zyyang@bsdsa ~]$ sudo ls /home/ftp/hidden/.exe/
test.exe
```

HW 3-2: Requirements (2/5)

- Uploadscript feature in pure-ftpd should be activated
- Log violation of uploadscript into /home/ftp/public/pureftpd.viofile
  - Format -

    timestamp **hostname** **programname**: **filename** violate file detected. Uploaded by **upload_user**.

```
[zyyang@bsdsa ~]$ cat /home/ftp/public/pureftpd.viofile
Nov 10 11:41:29 sa ftpuscr[33351]: /home/ftp/public/test.exe violate file detected. Uploaded by
sysadm.
```

HW 3-2: Requirements (3/5)

- You should write an rc script “ftp_watchd” as a daemon to start the pure-uploadscript program
    - pure-uploadscript should be run in the background when ftp_watchd is started
- Your service must support these operation:
    - $ service ftp_watchd start
    - $ service ftp_watchd stop
    - $ service ftp_watchd restart
    - $ service ftp_watchd status

HW 3-2: Requirements (4/5)

- Requires a pid file to indicate which process to stop

```
[zyyang@bsdsa:~]% cat /var/run/pure-uploadscript.pid
32803
```

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

```
[zyyang@bsdsa:~]% sudo service ftp_watchd start
Starting ftp_watchd.
```

  - Service stop

```
[zyyang@bsdsa:~]% sudo service ftp_watchd stop
Kill: 32803
```

HW 3-2: Requirements (5/5)

- Service restart

```
[zyyang@bsdsa:~]% sudo service ftp_watchd restart
Kill: 32803
Starting ftp_watchd.
```

- Service status

```
[zyyang@bsdsa:~]% sudo service ftp_watchd status
ftp_watchd is running as pid 32803.
```

HW 3-2: Grading (15/15%)

- pure-uploadscript
  - pure-uploadscript activated (2%)
  - File name with .exe should moved to /home/ftp/hidden/.exe/ (4%)
  - Record should be written in log file after violation file upload (3%)
- ftp_watchd
  - rc.d auto start on boot (2%)
  - Service operation work correctly
    - pure-uploadscript should be run in the background (2%)
    - start/status/stop/restart (2%)

HW 3-2: Hint

- Enable upload script under pure-ftpd.conf
    - CallUploadScript yes
- For pure-uploadscript, you can manually start the daemon by following command:
    - $ pure-uploadscript -B -r **/your/uploadscript/to/execute**
- pure-uploadscript(8)

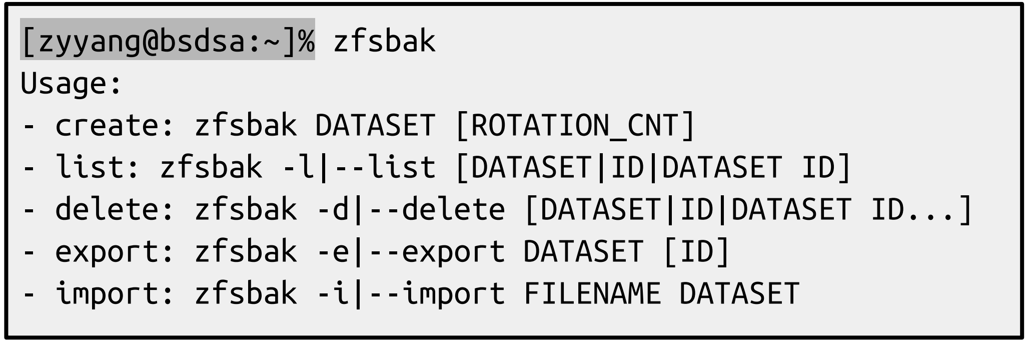
HW 3-3: ZFS & Backup (35%)

HW 3-3: Requirement (1/9)

- Enable ZFS service
  - Reboot and everything is fine (ZFS still mounted)
- Add two new hard disks and create a mirror pool called “mypool”
  - Mount **mypool** on */home/ftp*
- Create ZFS datasets
  - Set lz4 compression, atime=off to all datasets
  - Create **mypool/public**, **mypool/upload**, **mypool/hidden**

HW 3-3: Requirement (2/9)

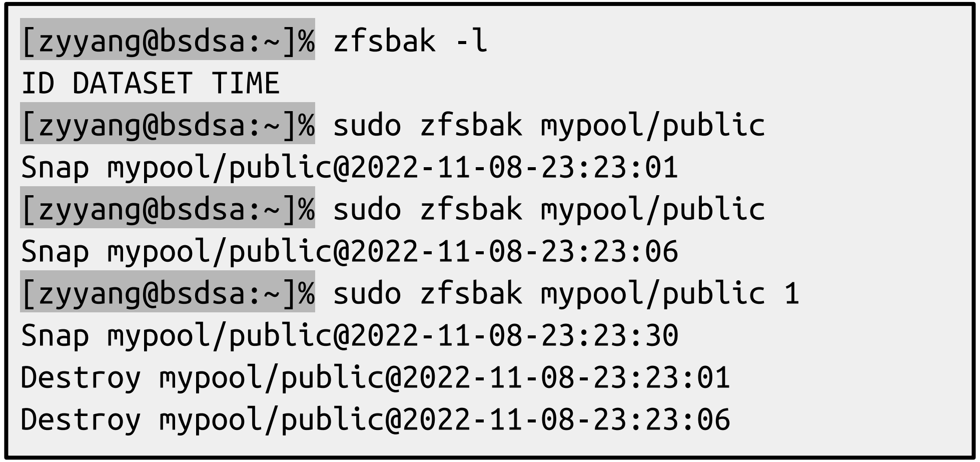
- Automatic Snapshot Script: **zfsbak**
  - Add your script to $PATH
    - Allow to execute zfsbak with command “zfsbak”, not “./zfsbak”
  - Usage:
    - Create: zfsbak DATASET [ROTATION_CNT]
    - List: zfsbak -l|--list [DATASET|ID|DATASET ID]
    - Delete: zfsbak -d|--delete [DATASET|ID|DATASET ID...]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



```
[zyyang@bsdsa:~]% 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 (3/9)

- 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



```
[zyyang@bsdsa:~]% zfsbak -l
ID DATASET TIME
[zyyang@bsdsa:~]% sudo zfsbak mypool/public
Snap mypool/public@2022-11-08-23:23:01
[zyyang@bsdsa:~]% sudo zfsbak mypool/public
Snap mypool/public@2022-11-08-23:23:06
[zyyang@bsdsa:~]% sudo zfsbak mypool/public 1
Snap mypool/public@2022-11-08-23:23:30
Destroy mypool/public@2022-11-08-23:23:01
Destroy mypool/public@2022-11-08-23:23:06
```

HW 3-3: Requirement (4/9)

- Specification - List
  - List snapshots created by zfs. **Sorted by time**.
  - 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 **dataset** and **id**
  - Otherwise, list all snapshots

```
[zyyang@bsdsa:~]% zfsbak -l
ID  DATASET          TIME
1   mypool/upload    2022-11-08-23:29:09
2   mypool/public    2022-11-08-23:29:43
3   mypool/hidden    2022-11-08-23:29:44
4   mypool/public    2022-11-08-23:29:46
```

```
[zyyang@bsdsa:~]% zfsbak -l 3
ID  DATASET          TIME
3   mypool/hidden    2022-11-08-23:29:44
[zyyang@bsdsa:~]% zfsbak -l mypool/public
ID  DATASET          TIME
1   mypool/public    2022-11-08-23:29:43
2   mypool/public    2022-11-08-23:29:46
[zyyang@bsdsa:~]% zfsbak -l mypool/public 2
ID  DATASET          TIME
2   mypool/public    2022-11-08-23:29:46
```

HW 3-3: Requirement (5/9)

- 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

```
[zyyang@bsdsa:~]% zfsbak -l
ID  DATASET              TIME
1   mypool/upload        2022-11-08-23:29:09
2   mypool/public        2022-11-08-23:29:43
3   mypool/hidden        2022-11-08-23:29:44
4   mypool/public        2022-11-08-23:29:46
5   mypool/upload        2022-11-08-23:29:46
6   mypool/hidden        2022-11-08-23:29:47
7   mypool/public        2022-11-08-23:30:03
```

```
[zyyang@bsdsa:~]% sudo zfsbak -d 1
Destroy mypool/upload@2022-11-08-23:29:09
[zyyang@bsdsa:~]% sudo zfsbak -d mypool/hidden 2
Destroy mypool/hidden@2022-11-08-23:29:47
[zyyang@bsdsa:~]% sudo zfsbak -d mypool/upload
Destroy mypool/upload@2022-11-08-23:29:46
[zyyang@bsdsa:~]% sudo zfsbak -d mypool/public 1 2
Destroy mypool/public@2022-11-08-23:29:43
Destroy mypool/public@2022-11-08-23:29:46
[zyyang@bsdsa:~]% sudo zfsbak -d
Destroy mypool/hidden@2022-11-08-23:29:44
Destroy mypool/public@2022-11-08-23:30:03
```

HW 3-3: Requirement (6/9)

- Log
  - Must contain the action (e.g. snap), dataset name and time
    - Print “**Snap `dataset@create_time`**” after creating the new snapshot, e.g.,
      - Snap mypool/public@2022-11-08-23:23:01
    - Print “**Destroy `dataset@create_time`**” after destroying the deleted snapshot, e.g.,
      - Destroy mypool/public@2022-11-08-23:29:43
  - For any undefined operation, just print the error message and exit

HW 3-3: Requirement(Bonus) (7/9)

- Specification - Export **(Bonus)**
  - Must specify **dataset**
  - **ID** defaults to 1
  - Compress with zstd
  - Encrypt with gpg
    - Import this public key into gpg keyring
    - Encrypt with this public key
    - Hint: Since you don’t have the private key to decrypt the exported file, you can use your key when testing
  - A filename example: `mypool_upload@2022-11-08-23:49:48.zst.gpg`
  - Put the export file at the user’s home directory.

```
[zyyang@bsdsa:~]% sudo zfsbak -e mypool/upload 1
Export mypool/upload@2022-11-08-23:49:48 to ~/mypool_upload@2022-11-08-23:49:48.zst.gpg
```

HW 3-3: Requirement(Bonus) (8/9)

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

```
[zyyang@bsdsa:~]% sudo zfsbak -i ~/mypool_upload@2022-11-08-23:49:48.zst mypool/upload2
Import ~/mypool_upload@2022-11-08-23:49:48.zst to mypool/upload2
[zyyang@bsdsa:~]% zfsbak -l
ID  DATASET            TIME
1   mypool/upload      2022-11-08-23:49:48
2   mypool/upload2     2022-11-08-23:53:13
[zyyang@bsdsa:~]% ls /home/ftp/
hidden  public  upload  upload2
```

HW 3-3: Requirement(Bonus) (9/9)

- **(Bonus)** Log
  - **(Bonus)** Print “**Export `dataset@create_time` to `file_location`**” after exporting the target snapshot, e.g.,
    - Export mypool/upload@2022-11-08-23:49:48 to ~/mypool_upload@2022-11-08-23:49:48.zst.gpg
  - **(Bonus)** Print “**Import `target_file` to `dataset`**” after importing the target file, e.g.,
    - Import ~/mypool_upload@2022-11-08-23:49:48.zst to mypool/upload2
  - For any undefined operation, just print the error message and exit

HW 3-3: Grading (35/35%, Bonus +10%)

- Create a mirror storage (3%)
- Create all dataset and set up correctly (3%)
- zfsbak
  - Usage (2%)
  - Create, List, Delete (9% / each)
  - Export, Import (include log) (Bonus +10%)

HW 3-3: Hint

- It will be much easier if you implement `Delete`, `Export`, `Import` with a well coding `List`
- 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

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.
- BACKUP your server before judge EVERY TIME
  - We may do something bad when judging.
- Make sure everything is fine after reboot.

Attention!

- TAs reserve the right of final explanations. Specs and the points of each sub-judges are subject to change in any time.
- We might randomly pick some student to demo after end of HW3.
- **Start from 2022/11/10 21:00**
- **Deadline 2022/11/30 23:59**

Help me!

Questions about this homework

- Ask them on https://groups.google.com/g/nctunasa
- We MIGHT give out hints on google group
  - 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

Good Luck!