Homework 3
File server & Backup

changhoy, kuochw

交大資工系資訊中心

Computer Center of Department of Computer Science, NCTU

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)
- All accounts 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
  - login from ssh (4%)
  - Full access to "***public***" (3%), "***upload***" (4%), "***hidden***" (4%)
- ftp-vip1, ftp-vip2 login
  - 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) /usr/local/bin/pure-pw
  - 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 (25%)

HW 3-2: Requirements (1/5)

- Create an "uploadscript.sh" for recording every uploading into */var/log/uploadscript.log*
- The log message must include **upload time**, **upload user**, **upload filename**, and **file size**
  - Format -

    upload_time: **upload_user** has uploaded file **upload_filename** with size **file_size**

```
[changhoy@bsdsa:~]% cat /var/log/uploadscript.log
Sun Oct 25 16:45:37 CST 2020: Anonymous has uploaded file /usr/home/ftp/public/anon.txt with
size 0
Wed Oct 28 14:37:45 CST 2020: ftp-vip1 has uploaded file /usr/home/ftp/public/150771764.doc with
size 44544
```

HW 3-2: Requirements (2/5)

- Create a service "ftp-watchd" which enables running a command after a successful upload
  - The name of the service should be exactly the same as "ftp-watchd"
  - Execute uploadscript.sh when a file is successfully uploaded to the FTP Server
  - Passing arguments described in rc.conf
    - Don't hardcore the command; let the command can be specified in rc.conf
- Execute a command defined in rc.conf whenever a file is uploaded

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
  - $ service ftp-watchd poll

HW 3-2: Requirements (4/5)

- Requires a pid file to indicate which process to stop

```
[changhoy@bsdsa:~]% cat /var/run/pure-uploadscript.pid
20878
```

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

```
[changhoy@bsdsa:~]% sudo service ftp-watchd start
Starting ftp-watchd.
```

  - Service stop

```
[changhoy@bsdsa:~]% sudo service ftp-watchd stop
Kill: 20878
```

HW 3-2: Requirements (5/5)

- Service restart

```
[changhoy@bsdsa:~]% sudo service ftp-watchd restart
Kill: 3458
Starting ftp-watchd.
```

- Service status

```
[changhoy@bsdsa:~]% sudo service ftp-watchd status
ftp-watchd is running as pid 3477.
```

- Service poll **(Bonus)**

```
[changhoy@bsdsa:~]% sudo service ftp-watchd poll
Waiting for PIDS: 3477
```

HW 3-2: Grading (25/25%, Bonus +5%)

- pure-uploadscript
  - pure-uploadscript should be activated (5%)
  - Record should be written in log file after any successful upload (5%)
- ftp-watchd
  - rc.d auto start on boot (5%)
  - Service operation work correctly
    - User can specify command in rc.conf (5%)
    - start/status/stop/restart (5%)
    - poll (Bonus +5%)

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 (25%)

HW 3-3: Requirement (1/8)

- 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 (2/8)

- 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

```
[changhoy@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 (3/8)

- Specification - Create (Default)
  - Must specify **dataset**
  - If no rotation count is specified, use 20 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

```
[changhoy@bsdsa:~]% zfsbak -l
ID DATASET TIME
[changhoy@bsdsa:~]% sudo zfsbak mypool/public
Snap mypool/public@2020-11-05-17:44:21
[changhoy@bsdsa:~]% sudo zfsbak mypool/public
Snap mypool/public@2020-11-05-17:44:28
[changhoy@bsdsa:~]% sudo zfsbak mypool/public 1
Snap mypool/public@2020-11-05-17:44:34
Destroy mypool/public@2020-11-05-17:44:21
Destroy mypool/public@2020-11-05-17:44:28
```

HW 3-3: Requirement (4/8)

- 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

```
[changhoy@bsdsa:~]% zfsbak -l
ID  DATASET           TIME
1   mypool/public     2020-11-05-17:46:55
2   mypool            2020-11-05-17:46:58
3   mypool/public     2020-11-05-17:58:01
```

```
[changhoy@bsdsa:~]% zfsbak -l 3
ID  DATASET           TIME
3   mypool/public     2020-11-05-17:58:01
[changhoy@bsdsa:~]% zfsbak -l mypool/public
ID  DATASET           TIME
1   mypool/public     2020-11-05-17:46:55
2   mypool/public     2020-11-05-17:58:01
[changhoy@bsdsa:~]% zfsbak -l mypool/public 2
ID  DATASET           TIME
2   mypool/public     2020-11-05-17:58:01
```

HW 3-3: Requirement (5/8)

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

```
[changhoy@bsdsa:~]% zfsbak -l
ID  DATASET            TIME
1   mypool/public      2020-11-05-17:49:16
2   mypool/public      2020-11-05-17:49:17
3   mypool             2020-11-05-17:49:22
4   mypool             2020-11-05-17:49:23
5   mypool             2020-11-05-17:49:24
```

```
[changhoy@bsdsa:~]% sudo zfsbak -d 1
Destroy mypool/public@2020-11-05-17:49:16
[changhoy@bsdsa:~]% sudo zfsbak -d mypool 2
Destroy mypool@2020-11-05-17:49:23
[changhoy@bsdsa:~]% sudo zfsbak -d mypool/public
Destroy mypool/public@2020-11-05-17:49:17
[changhoy@bsdsa:~]% sudo zfsbak -d
Destroy mypool@2020-11-05-17:49:22
Destroy mypool@2020-11-05-17:49:24
```

HW 3-3: Requirement (6/8)

- Specification - Export **(Bonus)**
  - Must specify **dataset**
  - **ID** defaults to 1
  - Compress with gzip
  - Encrypt with aes256 (Hint: Use openssl; Ask user to input password)
  - A filename example: `dataset@2020-11-05-17:53:07.gz.enc`
  - Put the export file at the user's home directory.

```
[changhoy@bsdsa:~]% sudo zfsbak -e mypool/public 1
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
Export mypool/public@2020-11-05-17:53:07 to ~/mypool/public@2020-11-05-17:53:07.gz.enc
```

HW 3-3: Requirement (7/8)

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

```
[changhoy@bsdsa:~]% sudo zfsbak -i ~/mypool/public@2020-11-05-17:53:07.gz.enc mypool/public2
enter aes-256-cbc decryption password:
Import ~/mypool/public@2020-11-05-17:53:07.gz.enc to mypool/public2
[changhoy@bsdsa:~]% zfsbak -l
ID  DATASET           TIME
1   mypool/public     2020-11-05-17:53:06
2   mypool/public     2020-11-05-17:53:07
3   mypool/public2    2020-11-05-18:00:45
[changhoy@bsdsa:~]% ls /home/ftp/
hidden  public  public2 upload
```

HW 3-3: Requirement (8/8)

- 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@2020-11-05-17:44:21
    - Print "**Destroy `dataset@create_time`**" after destroying the deleted snapshot, e.g.,
      - Destroy mypool/public@2020-11-05-17:49:16
    - **(Bonus)** Print "**Export `dataset@create_time` to `file_location`**" after exporting the target snapshot, e.g.,
      - Export mypool/public@2020-11-05-17:53:07 to ~/mypool/public@2020-11-05-17:53:07.gz.enc
    - **(Bonus)** Print "**Import `target_file` to `dataset`**" after importing the target file, e.g.,
      - Import ~/mypool/public@2020-11-05-17:53:07.gz.enc to mypool/public2
  - For any undefined operation, just print the error message and exit

HW 3-3: Grading (25/25%, Bonus +10%)

- Create a mirror storage (2%)
- Create all dataset and set up correctly (2%)
- zfsbak
  - Usage (1%)
  - Create, List, Delete (5% / each)
  - Log (5%)
  - 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!

- Due date: 23:59 December 2nd (Wed.)
- Email us if you finish bonus, we will judge manually
  - ta@nasa.cs.nctu.edu.tw

Help me!

- TA time: 3 GH at EC 324 (PC Lab)
- 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
  - Do not email us
  - Do not use e3 to email us

Good Luck!