



# Computer System Administration

## Homework 3 – File Server

---

fchsieh / zswu

# Overview

---

- ❑ Image that you are a TA of course, the professor want you to build a file server that students can submit their homework
- ❑ To prevent your stupid colleagues accidentally deleting files on the server, the snapshot and rollback features is needed
  
- ❑ File Server (100% + 40%)
  - FTP Server (60%)
  - ZFS on /home/ftp (20% + 10% Bonus)
  - RC (20% + 30% Bonus)
    - Upload script for pure-ftpd (20%)
    - ZFS auto backup service, zbackupd (30% Bonus)



FTP

---

fchsieh

# Installation

- ❑ Install a FTP server for FreeBSD
  - Pure-FTPd: */usr/ports/ftp/pureftpd/*
- ❑ Compile it with “upload script” support from ports

```
pure-ftpd-1.0.47
+ [ ] ANONDELETE      Allow anonymous user to delete files
+ [ ] ANONRENAME     Allow anonymous user to rename files
+ [ ] ANONRESUME     Allow anonymous user to resume file upload
+[x] DOCS            Build and/or install documentation
+[x] EXAMPLES       Build and/or install examples
+ [ ] LARGEFILE     2GB+ file download
+ [ ] LDAP          User accounts in LDAP directories
+ [ ] MYSQL         User accounts in MySQL database
+[x] PAM            PAM authentication
+ [ ] PERUSERLIMITS Per-user concurrency limits
+ [ ] PGSQL         User accounts in PostgreSQL database
+[x] PRIVSEP       Privilege separation (recommended)
+[x] SCRYPT        Hash passwords in PureDB with scrypt
+[x] SENDFILE      Sendfile syscall
+ [ ] THROTTLING    Bandwidth throttling
+[x] TLS           Secure network connection support via TLS
+[x] UPLOADSCRIPT  Uploadscript daemon support
+[x] UTF8          UTF-8 filenames
+[x] VIRTUALCHROOT Follow symlinks outside a chroot jail

< OK >          <Cancel>
```

# Requirements (1/2)

---

## 1. Anonymous Login

- Can **download** from */home/ftp/public*
- Can **upload & mkdir** from */home/ftp/upload*
- But **no download or delete** from */home/ftp/upload*
- Hidden directory problem */home/ftp/hidden*
  - There is a directory called “treasure” inside */home/ftp/hidden/*
  - Client can't list */home/ftp/hidden/* but can enter *hidden/treasure*

# Requirements (2/2)

---

2. Create a **system user** “sysadm”
  - Could login by **SSH**
  - Password is your student id
  - Full access to */home/ftp* and subdirectories under “*ftp*”
  - Private directory is visible to sysadm
3. Create a **virtual user** “ftp-vip”
  - Password is your student id
  - Permission setting is same as “sysadm”
4. Your ftp server should support **Explicit FTP over TLS** (FTPES)
5. All accounts are **chrooted** (*/home/ftp* is the root directory)

# Grading (1/2)

---

## ❑ Anonymous login (30%)

- Chrooted (*/home/ftp*) (5%)
- Download from “*/home/ftp/public*” (5%)
- Upload to “*/home/ftp/upload*” (5%)
- Can't download or delete form “*/home/ftp/upload*” (5%)
- Hidden directory “*/home/ftp/hidden*” problem: can enter but can't retrieve directory listing (5%)
- FTP over TLS (5%)

## ❑ sysadm (10%)

- Login from SSH (2%)
- Full access to “*/home/ftp*”, “*upload*”, “*public*” (3%)
- Full access to “*hidden*” (list, mkdir, upload, download...) (3%)
- FTP over TLS (2%)

# Grading (2/2)

---

## ❑ ftp-vip (20%)

- Chrooted (*/home/ftp*) (5%)
- Full access to “*/home/ftp*”, “*upload*”, “*public*” (5%)
- Full access to “*hidden*” (list, mkdir, upload, download, ...) (5%)
- FTP over TLS (5%)



# Hint (1/2)

---

## ❑ Lots of README

- */usr/local/share/doc/pure-ftpd/\**

## ❑ Accounts related

- Virtual user
- pure-pw(8)
- pure-pwconvert(8)
- README.Virtual-Users

## ❑ If `pure-pw mkdb` is not working

- Check your pure-ftpd.conf
- PureDB `/usr/local/etc/pureftpd.pdb`

# Hint (2/2): Demo Environment Setting

---

- ❑ Within VM, using notebook or connect to your machine using remote desktop
  - Don't need to open GUI for VM
  - You only need SSH and FileZilla
  - Use host-only network interface, connect from host OS
    - Host-only network interface: create connection between host and VM
- ❑ Within VM, no remote desktop
  - Using the public IP
  - Port forwarding
- ❑ Real Machine
  - Using the public IP
  - Firewall settings
- ❑ If you have a public IP, using bridge mode is better for VM

# ZFS on /home/ftp

---

fchsieh / zswu

# Requirements (1/6)

---

## Enable ZFS service

- Reboot and everything is fine (ZFS still mount)

## Create a mirror (RAID 1) storage called “mypool”

- Create a mirror storage pool using the zpool command
- Usually it is required to create a mirror storage pool with two different physical disk, you can just use two partition or even use “file as device” for this demo project

# Requirements (2/6)

---

## ❑ Create ZFS datasets

- mypool/upload on /home/ftp/upload
- mypool/public on /home/ftp/public
- mypool/hidden on /home/ftp/hidden
- Set gzip compression to all datasets

# Requirement (3/6): Zbackup

---

## ❑ Automatic Snapshot Script

## ❑ Specification

- Usage: `zbackup [--list | --delete | --export] target-dataset [ID] | [--import] target-dataset filename | target dataset [rotation count]`
- Example

```
$ sudo ./zbackup data/to/backup 5
```

```
$ sudo ./zbackup data/to/backup 5
```

```
$ sudo ./zbackup --list data/to/backup
```

ID	Dataset	Time
1	data/to/backup	2018-10-05 10:12:23
2	data/to/backup	2018-10-05 10:14:25

# Requirement (4/6): Zbackup

---

## ❑ Specification

- Create (Default)
  - No more than **rotation count** snapshots per dataset
  - If no **rotation count** specified, max 20 snapshots are allowed
  - If **rotation count** has reached, delete the oldest one
- List
  - List the snapshot created by zbackup
  - If **dataset** is specified, list only the snapshot of that dataset
  - Otherwise, list all of the snapshot
- Delete
  - Delete snapshots created by zbackup
  - Must specify **dataset**
  - If **ID** is specified, delete that one
  - Otherwise delete all snapshot of the dataset

# Requirement (5/6): Zbackup

---

## ❑ Specification

- Export
  - Must specify **dataset**
  - **ID** defaults to 1
  - Must compress with xz
  - Must encrypt with aes256 (Hint: Use openssl; Ask user to input password)
  - The filename should be **dataset/to/backup@2018-10-12.xz.enc**
- Import
  - Must specify **dataset**
  - **Filename** is the file exported by zbackup
  - Ask user to input password
  - Load the snapshot to the dataset



# Requirement (6/6): Zbackup

---

- ❑ For others non-define operation, just print an error message and exit
- ❑ For create, print log message to stdout
  - Snap dataset@create time of the new snap, e.g.,
    - Snap mypool/mydataset@Sun Oct 21 16:32:09 CST 2018
  - Rotate dataset@create time of the deleted snap, e.g.,
    - Rotate mypool/mydataset@Sun Oct 21 16:32:09 CST 2018
  - Log must contain the action (snap/rotate), time and dataset name, but the format has no requirement

# Grading

---

- ❑ ZFS on /home/ftp (20% + 10% Bonus)
  - Create a mirror storage (5%)
  - Create all dataset and set up correctly (5%)
  - Zbackup (20%)
    - Create (10%)
    - List, Delete (+5%)
    - Export, Import (+5%)

# Hint

---

## ❑ 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>

# RC: Upload script for Pure-FTPd

---

fchsieh

# Requirements (1/5)

---

- ❑ Create a RC service which can run a command after a successful upload, e.g., do zfs snapshot
  - Auto start on boot
  - Execute a shell command when a file is successfully uploaded to the FTP Server
  - Passing arguments described in rc.conf
    - Don't hardcore the command, the command can be specified in rc.conf

```
ftp_watchd_enable="YES"  
ftp_watchd_command="zbackup mypool/upload 10"
```

## Requirements (2/5)

---

- ❑ Execute a command defined in rc.conf whenever a file is uploaded
- ❑ For example, echo “HI” and write to a file /tmp/hi
  - a) Set the command in rc.conf

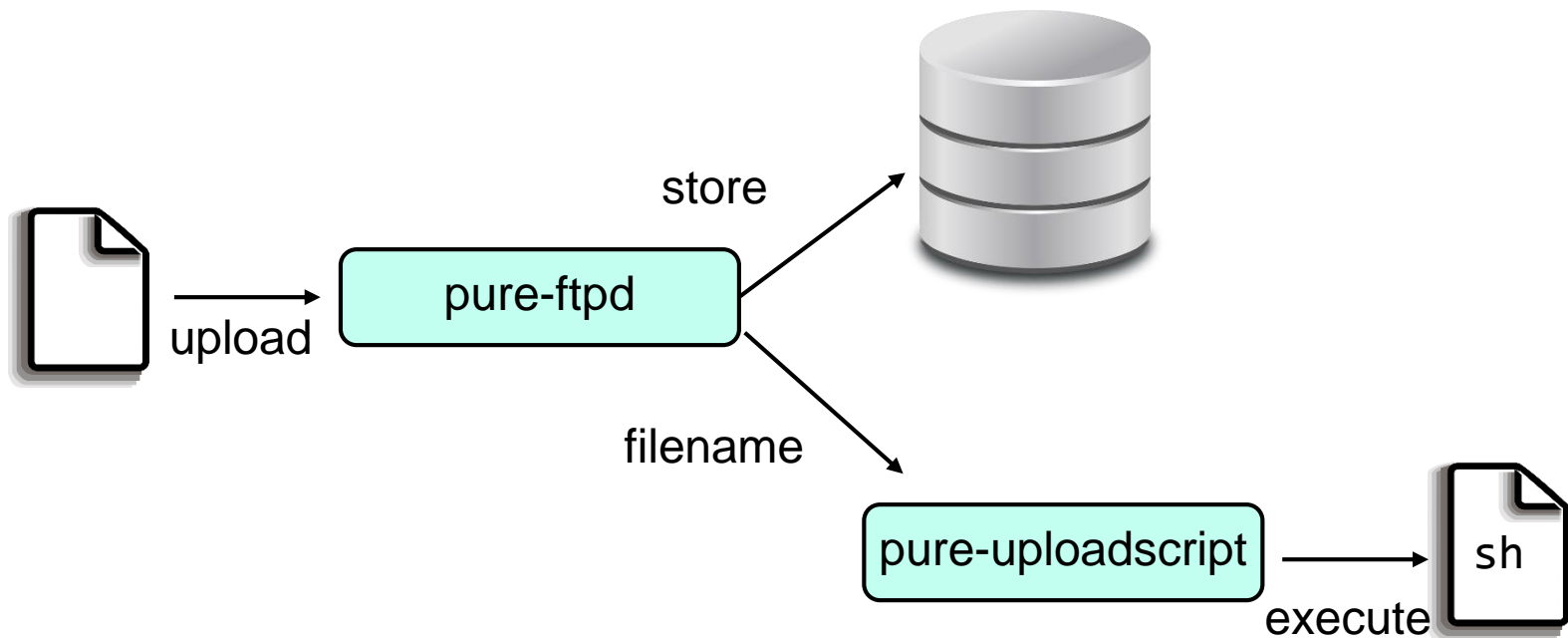
```
ftp_watchd_enable="YES"  
ftp_watchd_command="echo 'HI' >> /tmp/hi"
```

- a) The command should be invoked as expected after a successful upload

```
fchsieh@sabsd > /root > tail -f /tmp/hi  
HI  
HI  
HI
```

# Requirements (3/5)

- ❑ You can use pure-uploadscript to help you to do that
  - Automatically run an external program after a successful upload
  - Write a RC script, and turn pure-uploadscript into a daemon
  - You may also need to create a script for uploadscript
  - pure-uploadscript(8)



# Requirements (4/5)

- ❑ RC script can be used to start the pure-uploadsript program
  - Daemonize the uploadsript by ftp\_watchd
  - pure-uploadsript should be run in the background when ftp\_watchd is running

```
x fchsieh@sabsd ~ ➤ ps auxww | grep pure-uploadsript
root      45844  0.0  0.6 10332 5620  1  S+   03:11   0:00.00 pure-upload
fchsieh@sabsd ~ ➤
```

- ❑ Your service must support these operation
  - \$ service start ftp-watchd
  - \$ service stop ftp-watchd
  - \$ service restart ftp-watchd
  - \$ service status ftp-watchd
  - \$ service poll ftp-watchd



# Requirements (5/5)

- ❑ To stop ftp\_watchd service, you need to kill pure-uploadscript service

```
fchsieh@sabsd /u/l/e/rc.d sudo service ftp_watchd stop
Kill: 1758
fchsieh@sabsd /u/l/e/rc.d
```

- ❑ This requires a pid file to indicate which process to stop

```
fchsieh@sabsd /u/l/e/rc.d cat /var/run/pure-uploadscript.pid
1758
fchsieh@sabsd /u/l/e/rc.d
```

- You may need this to write the stopping function

# Hint

---

- ❑ Enable upload script under pure-ftpd.conf
  - CallUploadScript yes
- ❑ For pure-uploadsript, you can manually start the daemon by following command

```
pure-uploadsript -B -r /your/script/to/execute
```

# Grading

---

## ❑ ftp\_watchd (20%)

- All features work correctly (10%)
  - Command will be execute after a successful file upload
- rc.conf (5%)
  - Auto start on boot
  - User can specify command in rc.conf
- Service operation work correctly (5%)
  - start/stop/restart
  - status/poll



# RC: ZFS auto backup service Zbackupd

---

ZSWU

# Requirement (1/4): Zbackupd

---

## ❑ Automatic Backup Service

## ❑ Specification

- Usage: `zbackupd [-d] [-c /path/to/config/file.yaml] [-p /path/to/pid/file]`
  - -d, Background execution (Main process fork and exit)
  - -c, Path to config file, defaults to `/usr/local/etc/zbackupd.yaml`
  - -p, Path to pid file (A file contains the daemon process pid), defaults to `/var/run/zbackup.pid`
- If the daemon is running (i.e., pid file exists), show error and exit.
- Once you do a snapshot, print a log message to stdout
  - Just output the message output by `zbackup`

# Requirement (2/4): Zbackupd

---

## ❑ Configuration file syntax

```
1 backup:
2 - dataset: 'mypool/hidden'
3   enabled: false      # Default to true
4   rotation: 7
5   period: '1d'
6
7 - dataset: 'mypool/upload'
8   rotation: 10
9   period: '15m'
10
11 - dataset: 'mypool/public'
12   rotation: 4
13   period: '1w'
```

# Requirement (3/4): Zbackupd

---

## ❑ Configuration file syntax

- YAML format
- backup
  - enabled: defaults to true
  - rotation: rotation count for zbackup
  - period: [number][s|m|h|d|w] represent seconds, minutes, hours, days, weeks. The period of backup

## ❑ Do the first snapshot when the program started

# Requirement (4/4): Zbackupd

---

- ❑ You need to provide a RC script that can dynamically start/stop service (e.g., `$ service zbackupd start`)
- ❑ In `/etc/rc.conf`
  - `zbackupd_enable="YES"`
  - `zbackupd_config="/path/to/the/config/file"` (Default: `/usr/local/etc/zbackupd.yaml`)
- ❑ Actions that need to be supported
  - `start/stop/restart/reload`
  - `status/poll`
  - `list`
- ❑ If configuration file doesn't exist, you need to show error message.
- ❑ Redirect stdout to `/var/log/zbackup.log`



## Hint (1/2)

---

- ❑ How to write a rc start-up script:
  - <https://www.freebsd.org/doc/en/articles/rc-scripting/>
- ❑ You may need some shell script technique
  - <https://www.twbsd.org/cht/book/ch24.htm>
- ❑ /etc/rc.subr can help you to build rc service
  - <https://www.freebsd.org/doc/en/articles/rc-scripting/>

## Hint (2/2)

---

- ❑ You can use any language you want to implement this homework, however, we recommend you to use sh/bash to implement zbackup, use sh/bash/c/python etc... to implement zbackupd
- ❑ When exec “\$ service zbackupd reload”, the zbackupd process must have the same pid before and after reload. i.e. You need to send signal to the process to ask it to reload the config file instead of just restart the program

# Grading

---

## ❑ Zbackupd (30% Bonus)

- All features work correctly (+10%)
  - Follow the config file and do snapshot
- Support start/stop/restart/status/poll/list (+10%)
- Support reload (+10%)
  - Load new config file
  - Pid must not be changed

# Reminder

---

- ❑ Demo with root is **not allowed**
  - Please use **sudo**
- ❑ File/directory permissions are important
  - Owner, group, other
  - Read, write, execute
  - Set UID, set GID, sticky bit
  - Executable must be placed under `/usr/local/bin/`
- ❑ Setting **port forwarding** to allow your services to be accessible over the Internet
  - Host → NAT Port Forwarding → VM → FTP Server, etc.

# Deadline

---

- You do not need to submit anything
- Due(Demo): 2018/11/21

# Help!

---

- ❑ Email to [ta@nasa.cs.nctu.edu.tw](mailto:ta@nasa.cs.nctu.edu.tw)
- ❑ New E3 <https://e3new.nctu.edu.tw/>
- ❑ Office hour: 3GH at EC318