System and Network Administration

Syslog
cron
Accounts
sudo ++
Shares

Things of Special interest to SysAdmins

- **Logs and Audit Trails**
  - A detailed list of actions recorded by OS
  - File system Logs used to reinstate data
  - Usage Logs used for billing
  - Auditing used for security
    - Trace source of activity
    - Provide non-repudiation
- **Review high priority messages:**
  - failed logins
  - full disks
  - repeated messages

My-tiny.net Orientation::Logfiles

System Logging

syslogd - system logging daemon

- System log messages are normally written to files in `/var/log`

- Rules for logging are specified in `/etc/syslog.conf` in the form of

```
facility.priority    action
mail.info            /var/log/maillog
*.emerg              @mothership.mydomain.org
```

syslogd

- Whenever syslogd receives a log message, it acts based on the message's type (or **facility**) and its **priority**.
- **Facilities** are simply categories. Linux facilities are auth, authpriv, cron, dæmon, kern, lpr, mail, news, syslog, user, UUCP and local0 through local7.
- **Priorities** are hierarchical. Linux priorities are debug, info, notice, warning, err, crit, alert, emerg (in increasing order of urgency)

- Note that facility and priority are set by the programs that generate messages or the system administrator

```
facility.priority    action
mail.info            /var/log/maillog
*.emerg              @mothership.mydomain.org
```

- In practice, most log messages are written to files. If you list the full path to a filename as a line's action in syslog.conf, messages that match that line will be appended to that file. (If the file doesn't exist, syslog will create it.)

- https://www.linuxjournal.com/article/5476
- syslog Configuration - by Mick Bauer Dec 1, 2001

Facilities	Facility Codes†
auth	4
auth-priv	10
cron	9
dæmon	3
kern	0
lpr	6
mail	2
mark	n/a
news	7
syslog	5
user	1
uucp	8
local{0–7}	16–23
* (*"any facility"*)	n/a

Priorities (in increasing order)	Priority Codes†
none	n/a
debug	7
info	6
notice	5
warning	4
err	3
crit	2
alert	1
emerg	0
* (*"any priority"*)	n/a

Actions	
/some/file	(*log to specified file*)
-/some/file	(*log to specified file but don't sync after*)
|/some/pipe	(*log to specified pipe*)
|/dev/some/tty_or_console	(*log to specified console*)
@remote.hostname.or.IP	(*log to specified remote host*)
username1, username2, etc.	(*log to all user's screens*)

Usage of "!" and "=" As Prefixes With Priorities

*.notice (no prefix) = "any event with priority of 'notice' or higher

*.!notice = "no event with priority of 'notice' or higher"

*.=notice = "only events with priority 'notice'"

*.!=notice = no events with priority of 'notice'"

†Numeric codes should *not* be used in *syslog.conf* on Linux systems. They are provided here strictly as a reference. Should you need to configure a non-Linux syslog dæmon which uses numeric codes only, e.g.: Cisco IOS, to send syslog messages to your log server.

Centralized syslog Server

- A centralized log server that receives and processes log messages from multiple sources can be of great value to a system administrator.
  - When a hacker breaks into a system, the first thing she will want to do is to hide and stay hidden. To do this, most hackers will modify system logs to remove evidence of their existence.

Centralized syslog Server

- You can have the syslog daemon accept remote messages with two steps:

```
/etc/rc.d/rc.syslog stop
```

```
/etc/rc.d/rc.syslog listen
```

- When you check the output of

```
netstat -tulpn
```

- you should see it listening on udp-port 514.

Centralized syslog Server

- To have a client system send log entries to the syslog server, you need to edit /etc/syslog.conf and change the log target from the name of a file to **@server.host.name**

(using the name of your syslog server, of course).

- The messages will be processed according to the facility + priority selectors on the server without regard to origin.
- The source domain name will be logged as part of each message though, which can be longer or shorter: the man pages for syslog and syslog.conf have all the details.

Logging Policies

- Logrotate is designed to ease administration of systems that generate large numbers of log files. It allows automatic rotation, compression, removal, and mailing of log files.
- Each log file may be handled daily, weekly, monthly, or when it grows too large. Normally, logrotate is run as a daily ***cron job***.
- Configuration File /etc/logrotate.conf


Q
&
A

Regular Expressions

- Regular expressions allow us to select the output we want to see and ignore the rest, which is super-useful when we work with logfiles.
- The downside is that regular expressions can be very cryptic when they use multiple wildcard symbols to select sets of words, characters, or numbers,
- and (as if that isn't enough) we need to be aware that there are a number of regular expression dialects developers can choose to use.

Regular Expressions

- The standard unix tool for applying regular expressions is **grep**, which is capable of using three dialects of Regular Expressions: Basic (BRE) which is the default, Extended (ERE) with -E, and Perl-Compatible (PCRE) with -P
- The developers of multitail decided to use the "Perl Compatible Regular Expressions" (PCRE) libraries, which have also been adopted by the developers of Python and JavaScript.

Regular Expressions: Logfile Analysis

- In /var/log/dnsmasq.log there are a lot of *dnsmasq-dhcp* messages that say "wrong network", "no address available", and "range" (at startup).
- we want to exclude these when we select lines that start with today's three-letter Month and day and have "-dhcp" in the line.
- grep searches stdin and writes all of the lines with the search string to stdout. We have to use -e to protect a pattern that begins with - because grep is aggressive about its command line arguments (it does not respect the quotes alone).

Regular Expressions: Logfile Analysis

- Using grep we cannot select and exclude lines at the same time, so we need to pipe several command together to filter the output of each one, like this:

```
# can use cat and a pipe or input redirection to get started
bd="$(date +%b' '%d)"; \
grep "^$bd" < dnsmasq.log |grep -e '-dhcp' |grep -v 'no addr' |grep -v 'wrong' |grep -v 'range'
```

- reduced with regular expression:

```
# can use cat and a pipe or input redirection to get started
bd="$(date +%b' '%d)"; \
grep "^$bd" < dnsmasq.log |grep -e '-dhcp' |grep -v "no addr\|wrong\|range"
```

Regular Expressions: Logfile Analysis

- Using multitail we can select and exclude lines at the same time, as long as we follow the rules: [1] You can have multiple -ev (exclude) but only one -e (select) [2] All -ev must come before -e

- multitail equivalents of the grep examples:

```
# series of exclude filters: (only one -e is allowed, so that MUST be a regex)
bd="$(date +%b' '%d)"; \
multitail -ev "no addr" -ev "wrong" -ev "range" -e ^$bd".*-dhcp" dnsmasq.log next window
```

- reduced with regular expression:

```
bd="$(date +%b' '%d)"; \
multitail -ev ".*no addr+|.*wrong+|.*range" -e ^$bd".*-dhcp" dnsmasq.log next window
```

Regular Expressions: Logfile Analysis

- Script on the remote server /usr/local/sbin/showlog

```
#!/bin/bash
 read FILE theRest
 /usr/bin/tail -f /var/log/$FILE
```

- Set up a listener on the remote host using xinetd or netcat
  - SSH is a bit more complicated, and not really necessary for logfiles
- View the filtered local file and the remote file in multitail:

```
#!/bin/bash
# /usr/local/sbin/dhcpActivity ETH HOST PORT

 bd="$(date +%b' '%d)"; multitail -s 2 \
 -ev ".*no addr+|.*wrong+|.*range" \
 -e "^$bd+.*-dhcp+.*eth$1" \
 /var/log/dnsmasq.log \
 -l "echo dhcpcd-log |nc -w 300 $2 $3"
```

- Call the script

```
/usr/local/sbin/dhcpActivity 3 mail.tinynet.edu 23435
```


Q
&
A

Configuration and Maintenance

- Configuration –
  How to initially setup system as required
- Maintenance –
  How to keep it that way!!
  - Systems tend towards disorder during use
  - There are many ways for disorder to occur

- Adding and Removing Users
- Adding and Removing Hardware
- Performing Backups
- Installing New Software

- Monitoring the System
- Troubleshooting
- Maintaining Local Documentation
- Auditing Security
- Helping Users

Things of Special interest to SysAdmins

Procedures

- Chores are always done the same way
- Checklists reduce errors and forgotten steps
- It's faster to work from a recipe
- Changes are self-documenting
- Measurable standard of correctness

Automation

- Configuring and maintaining any non-trivial network can be a heavy workload….
- Automation hides the effort required, increasing the "efficiency" of administrators
- But may increase reliance on net services - Therefore won't work well if the network is unreliable!!
  - Lots of commercial tools from major manufacturers (Oracle, IBM, HP, etc.)
  - CFengine is a popular open-source tool
  - rdist, rsync, rcp can be used in tightly controlled environments

Executing Scheduled Tasks

- Most host management systems require regular execution of housekeeping tasks
  - This is a key feature in most configuration management systems
- Unix cron service
  - crontab command
  - /etc/crontab file format
- Windows Schedule service
  - Corresponds to linux at command

Virtual servers, NFS, and cron

- 'Virtual Hosts' allow the webserver to serve multiple sites with different domain names
  - Separate DocumentRoot, ScriptAlias, IndexDefault
  - Requires a cname in the DNS
- Users upload their files to a separate server in the "DMZ"
  - No user accounts on the main webserver
  - Firewall rules: webserver only internal network except http, https
  - Users upload to their home directory: minimal privileges
- cron job moves files to exported directory
  - Webserver mounts exported directory as DocumentRoot for the VirtualHost websites
  - Short delay to see updates


Q
&
A

User Accounts

- Unique username, UID number, and password
  - Stored in system password database
    - /etc/passwd /etc/shadow
    - /etc/group /etc/gshadow
- Create login directory for user (home)
- Specify initial shell program
- Set up standard initialisation files and login environment
  - defaults are usually copied from /etc/skel
- Usually done using adduser / useradd command

```
useradd -c "Bill Gates" -u 1001 -g msoft
  -d /home/billg -m -k /etc/skel -s /bin/bash billg
```

/etc/passwd

World readable: lots of applications need to know file permissions (UID, GID) and Home

Format:

```
username:password:uid:gid:gecos:homedir:shell
```

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
billg:x:1001:501:Bill Gates:/home/billg:/bin/bash
```

/etc/shadow

Only root read/write: actual encrypted password is stored here

Format:

```
username:password:lastchange:min:max:warn:inact:expire:
```

```
root:j3dghRBqe$2fjvGJ8js:12650:0:99999:7:::
bin:*:12650:0:99999:7:::
nobody: ...
billg: ...
```

`*` does not match any password

`!!` account is locked

/etc/group

- Used to allocate process or file permissions to groups or Collections of existing users of users
- Individual users may be members of several groups
- **Seldom used now, common to see group same as user**

Format of a group entry in /etc/group

```
groupname:password:gid:user_list
```

`user_list` separated by commas, no spaces

- Group passwords can be stored in /etc/gshadow
- Change your group with `newgrp [ group ]`

“Non-system” accounts

- **/home/vmail/mail-pwd**
  - Username:Password
  - not a system login
    - maintained by sysadmin
    - verified by dovecot
- HTTP Basic Authentication (e.g., for membership)
  - Username:Password
  - not a system login
    - maintained by webmaster
    - verified by webserver

System Users

- postfix / dovecot / nobody
- vmail:vmail

- Creating a standard user account with a false shell stops that account from being able to log into the system, but still allows them to own files and processes

```
useradd -c “dovecot process" -s /sbin/nologin vmail
```

files and permissions

- UID Zero: root (superuser)
- UIDs mapped in /etc/passwd
- GIDs mapped in /etc/group

System Users: **No Password, Owner of files and processes**

nobody
postfix
dovecot
vmail

Read, Write, Execute

Owner, Group, World

Unix permission bits:

4 = read
2 = write
1 = execute

Permissions

4 + 2 + 1 = 7

644 means what?
755 means what?

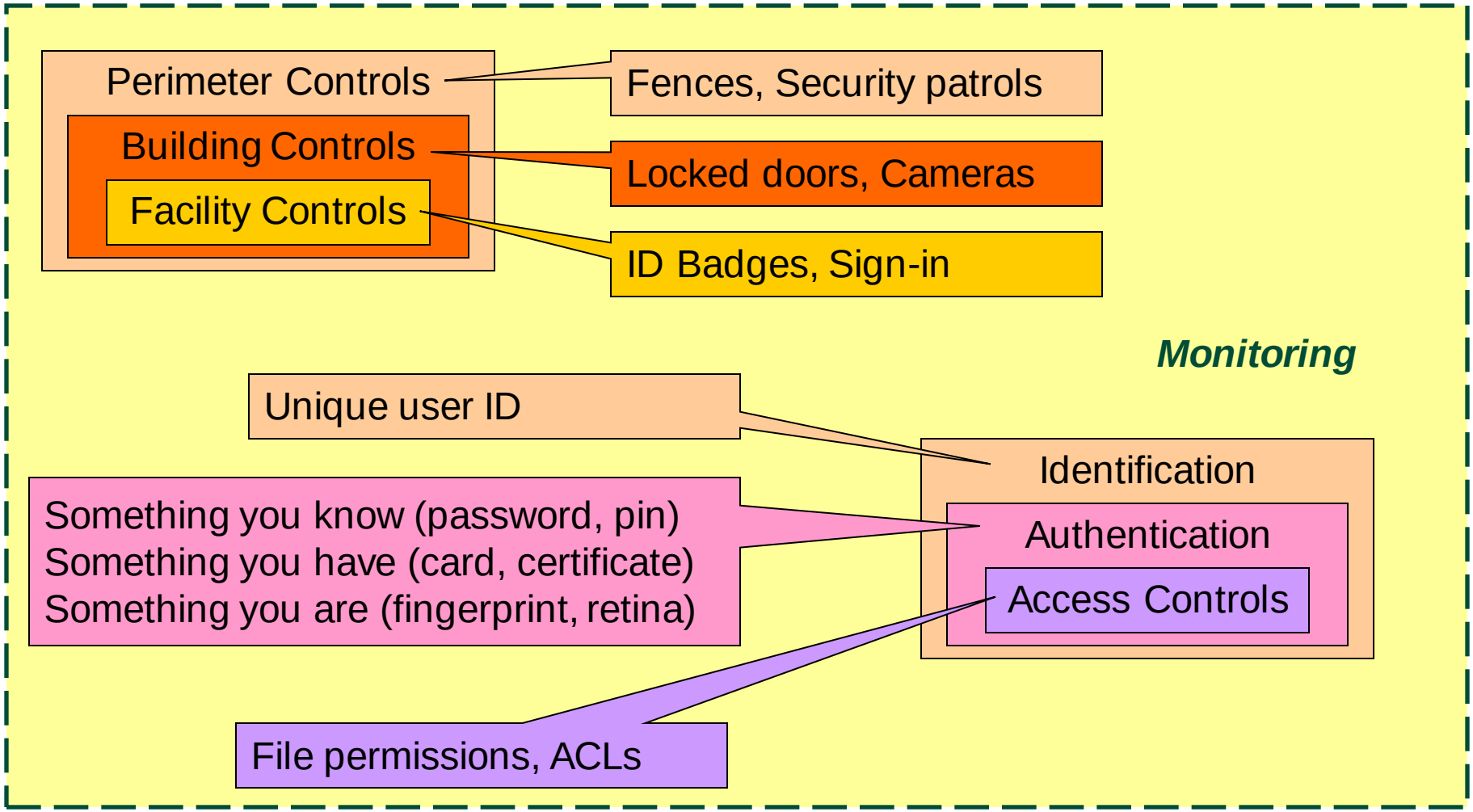
oddities

- [x] set for a file makes it executable
- [x] set for a directory allows seeing the files there

- drwxrwxrwt is a directory with the “sticky bit” set
- Anyone can create files, only the owner can delete


Q
&
A

Layers of Security (Defense in Depth)



Analysing Security Tradeoffs

- **Tradeoffs must be made between achieving security goals and goals for affordability, usability, performance, and availability.**
- ***More security = Less convenience***
- Security adds to the amount of management work (i.e. login IDs, passwords, and audit logs)
- Packet filters and data encryption consume CPU power and memory on hosts, routers, and servers
- Security can reduce network redundancy, and make it harder to offer load balancing

sudo

- Some distributions enable the root user (such as Fedora, Red Hat, openSuSE), while some do not (such as Ubuntu and Debian). There are pros and cons for each.
- Sudo stands for either "substitute user do" or "super user do" (depending upon how you want to look at it).
- Effectively, sudo allows a user to run a program as another user (most often the root user).
- There are many that think sudo is the best way to achieve "best practice security" on Linux. There are many others, however, that feel quite the opposite.

su and sudo

With classic distributions (like ours) you can log in as the root user, or you can issue the command

```
su -
```

as a normal user to effectively log in as root (root's home becomes your home).

To many sysadmins this is a bad idea. They say NEVER log in as the root user: log in as a standard user and su to become the root user.

With sudo-based distributions such as Ubuntu, the root user account has been "disabled." You cannot log in as root and you cannot su to become the root user. All you can do is issue commands with the help of sudo to gain administrative privileges.

/etc/sudoers

Technically, you should use a specific command - sudo visudo.

This locks the logins while /etc/sudoers is being changed.

In Reality you can use any text editor if your system is not busy.

When you open up this file you will notice that it is fairly small in size.

The basic entry for a user looks like this:

user hostlist = (userlist) commandlist

Typically you will find an entry like this:

```
root   ALL=(ALL) ALL
```

Which indicates that the user root on all hosts can run all commands as any user.

/etc/sudoers

- allow the user mary to issue a command without having to issue a password. Add

```
mary ALL = NOPASSWD: /usr/sbin/synaptic
```

- to the /etc/sudoers file. Now the user mary can run synaptic by entering `sudo synaptic` but will not be prompted for a password.

- To provide sudo access to a group, add the following line to the /etc/sudoers file.

```
%admin ALL=(ALL) ALL
```

sudo

http://www.linux.com/learn/tutorials/306766:linux-101-introduction-to-sudo

- comment out an entry by adding a "#" at the beginning of the line.

- keep track of this - lots of links at the bottom

http://www.cyberciti.biz/faq/linux-sudo-configuration-howto/

- youtube

http://www.youtube.com/watch?v=hv3QxFfkW-8
http://www.youtube.com/watch?v=BPMNofGgD4Y

Restricted Shell

If bash is started with the name rbash (via a symlink) the shell becomes restricted. It behaves identically to the standard shell with the exception that the following are disallowed or not performed:

- changing directories with *cd*
- specifying command names containing /
- specifying a file name with a ./ (dot slash)
- setting or unsetting the values of SHELL, PATH, ENV, or BASH_ENV
- redirecting output using the >, >|, <>, >&, &>, and >> redirection operators

However, *when a shell script is executed, rbash turns off any restrictions in the shell spawned to execute the script.* This includes the files bash reads when it starts up.

Chroot jail

- A **chroot** operation changes the apparent root directory for a running process and its children.
- To the process, it appears that the directory in which it is running is the root directory (/)
- The program cannot see or access files outside the designated directory tree.
- Such an artificial root directory is called a *chroot jail*, and its purpose is to limit the directory access of a potential attacker.
- The chroot jail locks down a given process and any user ID that it is using so that all they see is the directory in which the process is running.

Chroot jail

- For a chroot process to be able to start successfully, you must populate the chroot directory with **all** required program files, configuration files, device nodes, and shared libraries at their expected locations relative to the level of the chroot directory.

To set up a useful chroot jail, first determine which utilities the users of the chroot jail will need. Then copy the appropriate binaries and their libraries **(use ldd!)** into the jail directory tree (e.g., /jail/usr/lib, /jail/sbin)

- **Caution:** A program can break out of a chroot jail if it can gain root privilege and use chroot() to change its current working directory to the real root directory.
  - ensure that a chroot jail does not contain any setuid or setgid executables that are owned by root.

Chroot jail

- Running a shell inside a jail has limited usefulness – more likely to need to a specific service inside the jail.
- It bears repeating that to run a service inside a jail, you must make sure all files needed by that service are inside the jail directory tree.

- The format of a command to start a service in a chroot jail is

  `/usr/sbin/chroot` ***`jailpath`*** `/bin/su` ***`user daemonname`*** &

- where ***jailpath*** is the pathname of the jail directory, ***user*** is the username that runs the daemon, and ***daemonname*** is the path (inside the jail) of the daemon that provides the service.

- Some servers are already set up to take advantage of chroot jails. For example, you can set up DNS so that `named` runs in a jail, and the vsftpd FTP server can automatically start chroot jails for clients.

Chroot jail

Good Reference:

http://docs.oracle.com/cd/E37670_01/E36387/html/ol_cj_sec.html

Good exercise

http://how-to.linuxcareer.com/how-to-automatically-chroot-jail-selected-ssh-user-logins

Challenge:

Finding a way to justify the extra work and overhead

Note:
We see this in system descriptions, so we need to know what it is even if we don't use it


Q
&
A

Network File Systems

Mounting remote filesystems locally

- e.g., home directory, data files

- ✓ SSH (FUSE / FISH)
- ✓ Network File System (originally by SUN)
  - Uses Remote Procedure Calls (RPC)
- ○ Windows
  - NT LAN Manager, Workgroups (SMB) aka “samba”
  - Common Internet File System (CIFS)
- ○ Web-based Distributed Authoring and Versioning (WebDAV)

SSH

Remote access: start the ssh service on two VMs

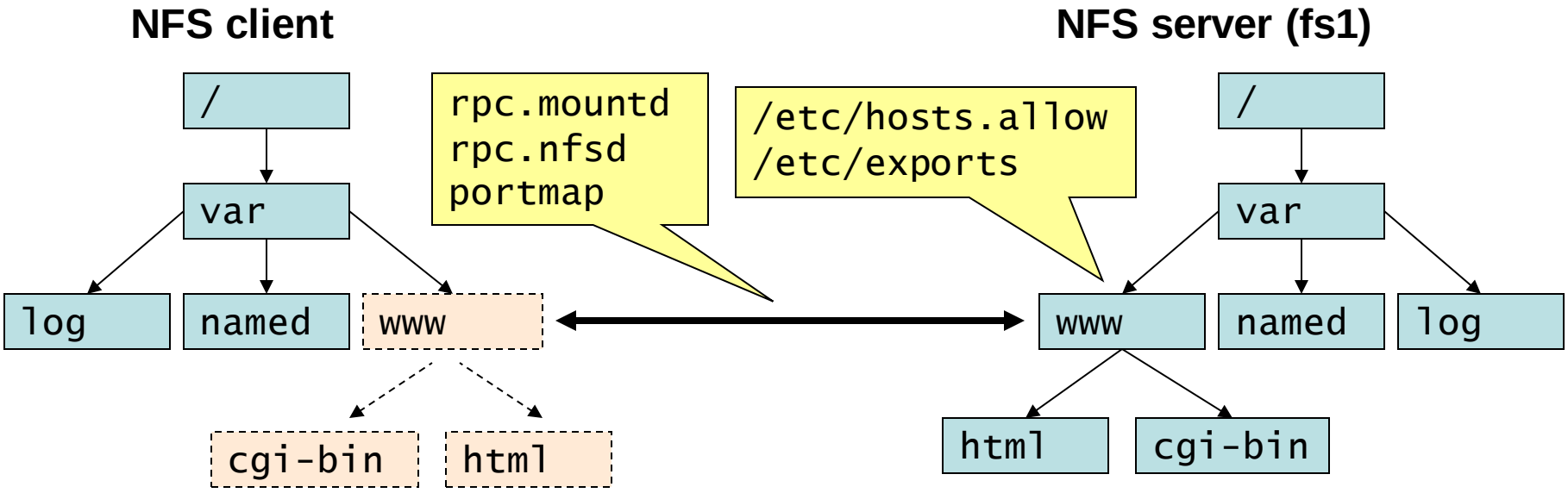
- Then `ssh ipaddress` or `ssh hostname`
  - To get a login shell

- Use mc and “Shell Link” { [F9] then Left or Right }
  - Note that this is a one-way connection:
  - can copy FROM the remote directory, but not TO it

- Use **winscp** on your windows host
  - Can contact VMs on the same subnet as the host-only adapter

NFS

```
mount -t nfs -o rw,soft,intr,suid fs1:/var/www /var/www
```



RPC processes notify portmap when they start, revealing the port number they are monitoring and the RPC program numbers they expect to serve. The client contacts portmap on the server with a particular RPC program number, and portmap redirects the client to the proper port to communicate with its desired service. ***(a bit like xinetd)***

NFS Client

sap p.100-107 rg p.121

/etc/fstab controls what file systems are mounted when the system boots, as well as supplying default values for others that may be mounted manually from time to time

Placing a properly formatted line in /etc/fstab has the same effect as manually mounting the file system.

```
# specifier   mountpoint    type      options          dump check
  /dev/hda1      /           ext2      defaults            0 1
  /dev/sda1      /mnt/usb    auto      defaults            0 0
  none           /proc       proc      defaults            0 0
  /dev/hda2      swap        swap      defaults            0 0
  /dev/cdrom     /mnt/cdrom  iso9660   noauto,owner,ro     0 0
192.168.66.78:/var/www  /var/www  nfs  rw,soft,intr,suid 0 0
```

hostname, IP address, or FQDN

local mount point must exist before /etc/fstab is read – if there are files here they will be invisible until unmounted

Server-Side NFS

`/etc/exports` – list of directories to export

- In its simplest form, only need to list the directory to be exported and the hosts permitted to use it:

```
/var/www          192.168.76.54
/var/spool/mail   workhorse
```

IP address,
NetAddress/mask
hostname, FQDN

exportfs – command to export directories

- `rpc.mountd` refers to `/var/lib/nfs/xtab` when deciding access privileges for a file system,
- **`exportfs -r`** causes all directories listed in `/etc/exports` to be exported by constructing a new export list in `/var/lib/nfs/xtab`

/etc/exports *Warning*

Be careful not to add extraneous spaces when editing /etc/exports

- For instance, the following line in the /etc/exports file shares the directory with read and write permissions to a single host

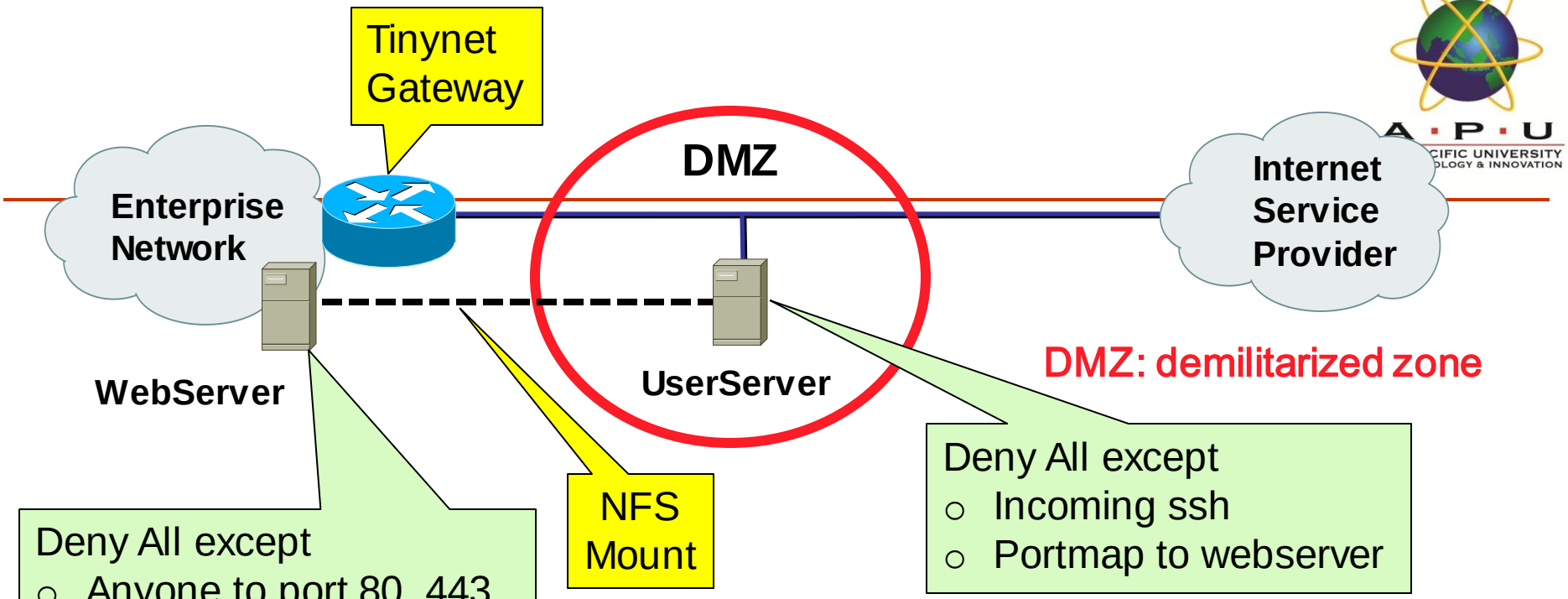
```
/tmp/nfs/ client.example.com(rw)
```

- on the other hand, the following line shares the directory to one host as read only and shares it to the world as read+write due to a single space character after the hostname.

```
/tmp/nfs/ client.example.com (rw)
```

check configured NFS shares: `showmount -e <hostname>`

check with `nfsstat -s` `<or>` `nfsstat -c`



The UserServer should really

- have a public IP address (Bridged Interface)
- and act as a http/https proxy (port or URL forwarding)

RHEL Guide: Securing Portmap

SMB/CIFS

- "samba"
  - Server Message Block protocol
  - Designed for Microsoft NetBIOS networking (before TCP/IP)
  - Still used by Linux clients to access shared printers controlled by Windows servers
- Common Internet File System (CIFS)
  - Can be used to share folders on Windows servers with Linux clients
  - Requires special support and configuration

WebDAV

- The WebDAV protocol is a popular option for accessing files remotely as it runs over the http/https protocols which are accessible from any location.
- Web Distributed Authoring and Versioning (WebDAV) is an extension of the HTTP protocol to allow end users and applications to view and edit files.
- WebDAV (RFC 4918) dates back to the late 90s. The versioning and configuration management piece of WebDAV (RFC 3253) and search capabilities (RFC 5323) were added as extensions.

WebDAV

- WebDAV implementations can be quirky. Many servers and clients implement subsets or extended subsets of the multiple standards involved.
- Thus interoperability can't be assumed; success depends on the platform, environment, and vendor-specific extensions.
  - The built-in Windows IIS WebDAV implementation does not support large file size transfers, enumeration of folders and group shares, Special File Name Characters, nor does it support several enhancements to the WebDAV standard.

https://www.comparitech.com/net-admin/webdav/https://en.wikipedia.org/wiki/Comparison_of_WebDAV_software

http://redmine.lighttpd.net/projects/1/wiki/Docs_ModWebDAV

https://fossies.org/linux/lighttpd/doc/config/conf.d/webdav.conf


Q
&
A