System Admin

Class 6 - Security and Web
by Ian Robert Blair, M.Sc.

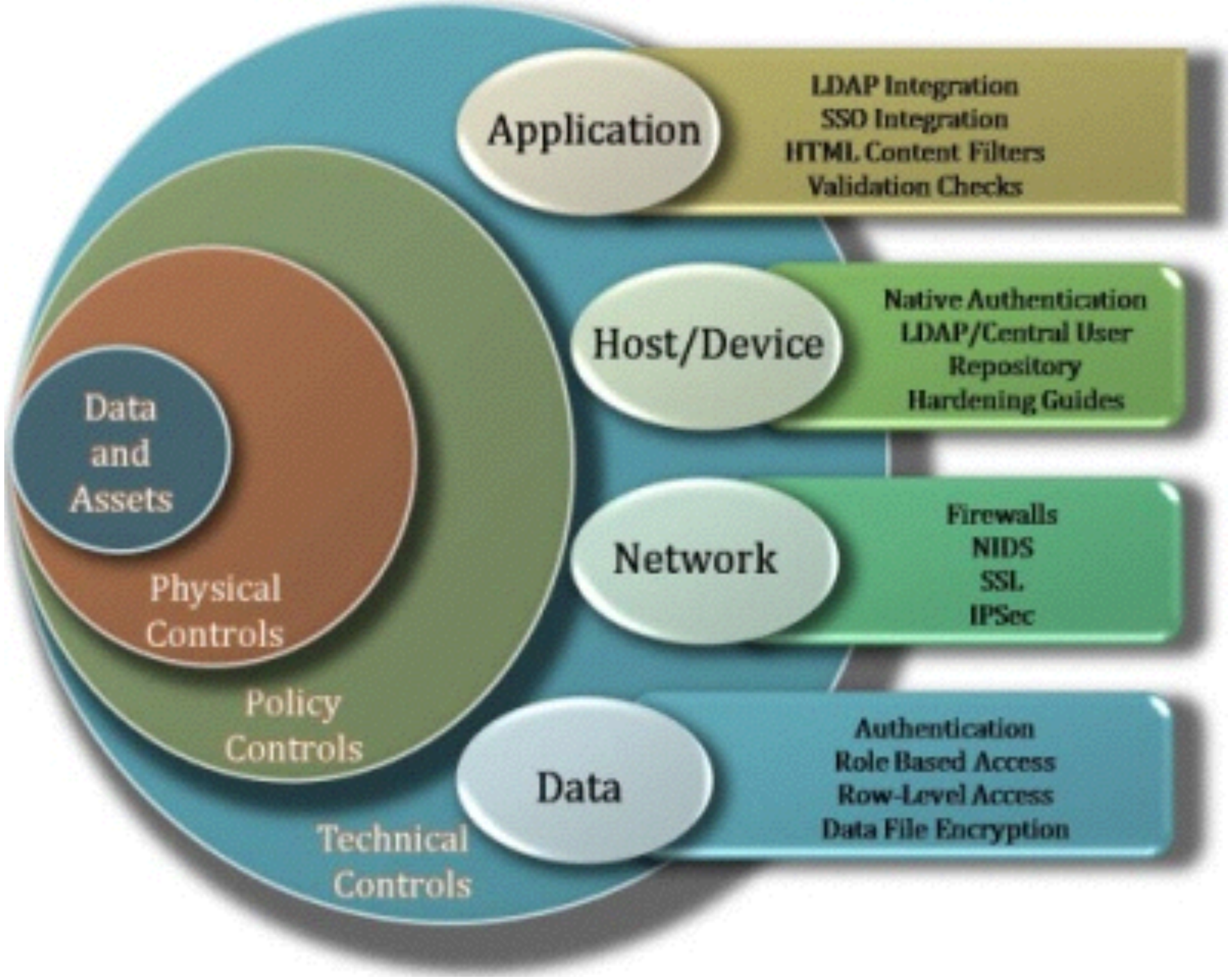
Agenda

- SSH
- IP Tables
- WEB/FTP

Security

Defense-in-Depth

The use of multiple computer security techniques to help mitigate the risk of one component of the defense being compromised or circumvented



XINETD

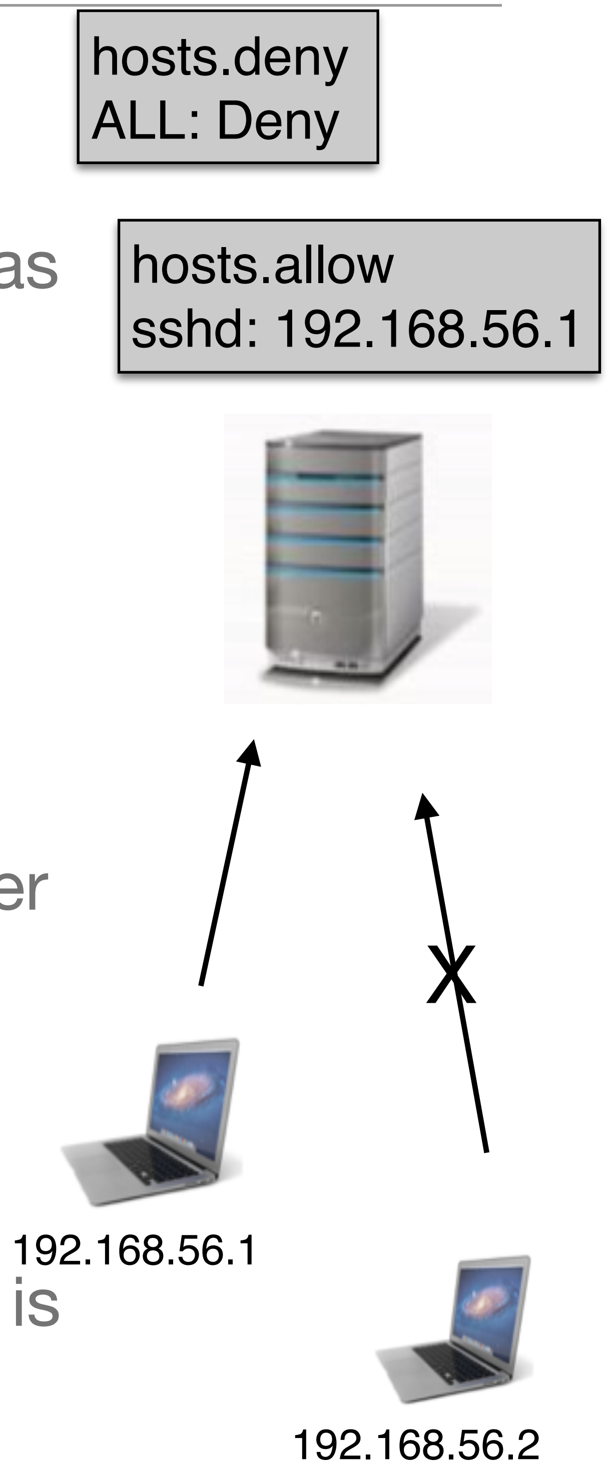
- When a network connection is made, **XINETD** identifies a server daemon based on the port the connection comes in on, sets the daemon's standard input and standard output file descriptors to the socket, and starts the **daemon**

- The **superserver** avoids the need for daemons to run when not in use

- The configuration for xinetd is stored in the **/etc/xinetd.conf** file and the files in /**etc/xinetd.d**

TCP Wrappers

- **TCP wrappers** some daemons
- It relies on the **/etc/hosts.allow** and **/etc/hosts.deny** files as the basis of a simple access control list (ACL)
- Each line in the hosts.allow and hosts.deny files has the following format:
  - sendmail : 192.168.1.0/255.255.255.0
  - sshd : 192.168.1.2 172.16.23.12
- daemon_list is a comma-separated list of one or more server daemons (e.g., rpcbind, vsftpd, sshd)
- client_list is a comma-separated list of one or more clients
- If the same client/user/ip is listed in both hosts.allow and hosts.deny, then hosts.allow takes precedence and access is permitted



hosts.deny
ALL: Deny

hosts.allow
sshd: 192.168.56.1

Chroot

- The **chroot** utility allows you to run a process with a root directory other than /
- For example, you run a program (process) and specify its root directory as /home/sam/jail
- (chroot) jail can prevent program from accessing, executing, or modifying files outside the directory hierarchy starting at its root
- Some servers are already set up to take advantage of chroot jails (ex. ftp)

SSH

OpenSSH is a suite of secure network connectivity tools that replaces telnet/telnetd, rcp, rsh/rshd, rlogin/rlogind, and ftp/ftpd (encrypts all traffic and passwords)

- scp—Copies files to and from a remote system
- sftp—Copies files to and from a remote system (a secure replacement for ftp)
- ssh—Runs a command on or logs in on a remote system
- sshd—The OpenSSH daemon (runs on the server)
- ssh-keygen—Creates, manages, and converts RSA or DSA host/user authentication keys

SSH Client

- The **ssh utility** allows you to log in on a remote system over a network
- It uses a host key pair and a session key to negotiate an encrypted session
- The host key pair is a set of public/private keys that is established the first time the server system runs sshd
- The session key is a symmetric key that the server and client share
- Global files are kept in **/etc/ssh** and user files in **~/.ssh, usage**:
  - ssh [options] [user@]host [command] (ex. ssh zach@plum)
- After connecting, the client appends the server's public host key to the user's **~/.ssh/known_hosts file**, to remove an entry:
  - ssh-keygen –R host (remove entry)

Open SSH Process

1. First, OpenSSH asks you to verify the client is connected to the correct server
2. After verification, the client makes a copy of the server's public host key (saved for future connections)
3. The client then generates a random key, which it encrypts with both the server's public host key and the session key
4. The client sends this encrypted key to the server
5. The server, in turn, uses its private keys to decrypt the encrypted key
6. This process creates the session key, a key that is known only to the client and the server and is used to encrypt the rest of the session

SCP and SFTP

- The **scp (secure copy)** utility copies an directory or file from one system to another over a network; both systems can be remote
  - scp [[user@]from-host:]source-file [[user@]to-host:][destination-file]
  - scp sls@plum:memo.txt allmemos
  - scp sls@plum:memo.txt speedy:old
- sftp is a secure alternative to ftp
  - sftp plum

Open SSH Server

- Uses TCP port 22
- Configured by the **/etc/ssh/sshd_config** configuration file
- Some important directives:
  - Deny/AllowUsers userlist - is a SPACE-separated list of usernames that specifies which users are allowed or not allowed to log in using sshd
  - PasswordAuthentication - permits a user to use a password for authentication
  - PermitRootLogin - Setting this declaration to no does not allow root to authenticate directly

SSH Tunneling

- The **ssh** utility can forward a port through the encrypted connection it establishes
- The **–L** option forwards a local port to a remote system, so a program that tries to connect to the forwarded port on the local system transparently connects to the remote system
- The **–N** option, which prevents ssh from executing remote commands
  - ssh -N -L 1550:pophost:110 pophost

Rsync

- The **rsync** utility is more configurable than **scp** and uses OpenSSH security by default
- The **–a** option preserves the ownership, group, permissions, and modification times associated with the files
    - rsync -av sls@plum:memo.txt allmemos

TCP Dump

- Network packet analysis tool
  - The -i [any] option allows you to select an interface
  - If you don’t need resolve names, use -n
  - Add -v for verbose, up to -vvv
  - Option -c is for the number of packets, and -e is to get the header as well
  - Examples:
    - tcpdump -nvS
    - tcpdump net 1.2.3.0/24
    - tcpdump icmp
    - tcpdump -s 1514 port 80 -w capture_file
    - tcpdump 'src 10.0.2.4 and (dst port 3389 or 22)'

NMap

- Security scanning tool
- map -O target.host.com
- nmap -O 10.0.0.1-42.
- nmap -vv host.target.com
- nmap -sV

Host-based IDS

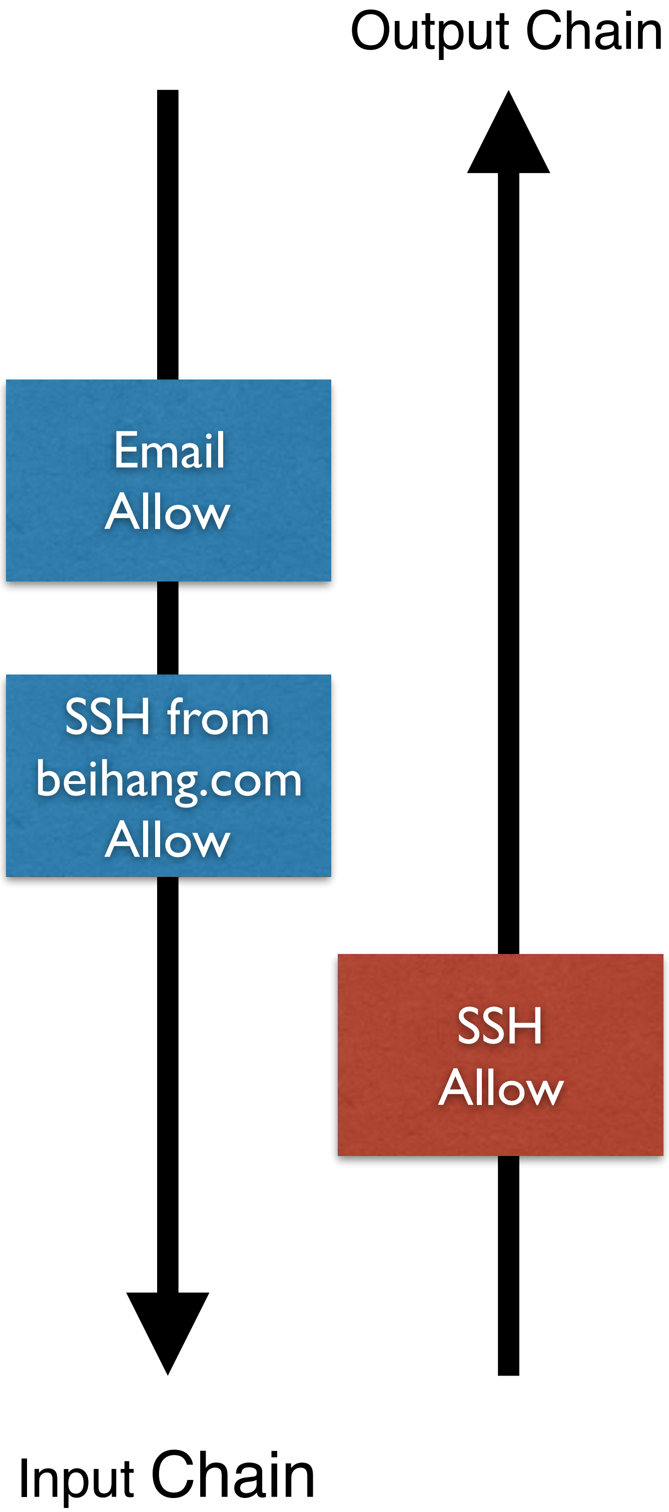
- A **rootkit** is a collection of computer software, typically malicious, designed to enable access to a computer
- Often it masks its existence or the existence of other software
- Tools to check for root kits:
  - rkhunter, chkrootkit
- AIDE (Advanced Intrusion Detection Environment), is a tool to prevent intrusion
  - File and directory integrity checker
  - Uses several message digest algorithms to check the integrity of files
  - Unix, Mac OSX, and Linux

Firewall Configuration (GUI)

- The **system-config-firewall** is a user-friendly, graphical front-end for iptables and ip6tables (text interface, system-config-firewall-tui)

IP Tables 1

- **iptables** and **ip6tables** is composed of two components: netfilter and iptables
- A **rule** comprises one or more criteria (matches or classifiers) and a single action (**a target**)
- Rules are stored in **chains**
- Each rule in a chain is applied, in order, to a packet until a match is found
- If there is no match, the chain's **policy,** or default action, is applied to the packet
- **firewalld** is Redhat's new firewall configuration utility, you must disable to it first (systemctl stop/disable firewalld)



IP Tables 2

- Chains are collected in three tables: **Filter, NAT,** and **Mangle**
- The Filter table
  - Used to DROP or ACCEPT packets based on their content; it does not alter packets
  - Builtin chains are **INPUT, FORWARD,** and **OUTPUT**
  - All user-defined chains go in this table
- The NAT table:
  - Used exclusively to translate the source or destination fields of packets
  - Builtin chains are **PREROUTING, OUTPUT,** and **POSTROUTING**
  - **DNAT (destination NAT)** alters the destination IP address of the first inbound packet in a connection so it is rerouted to another host

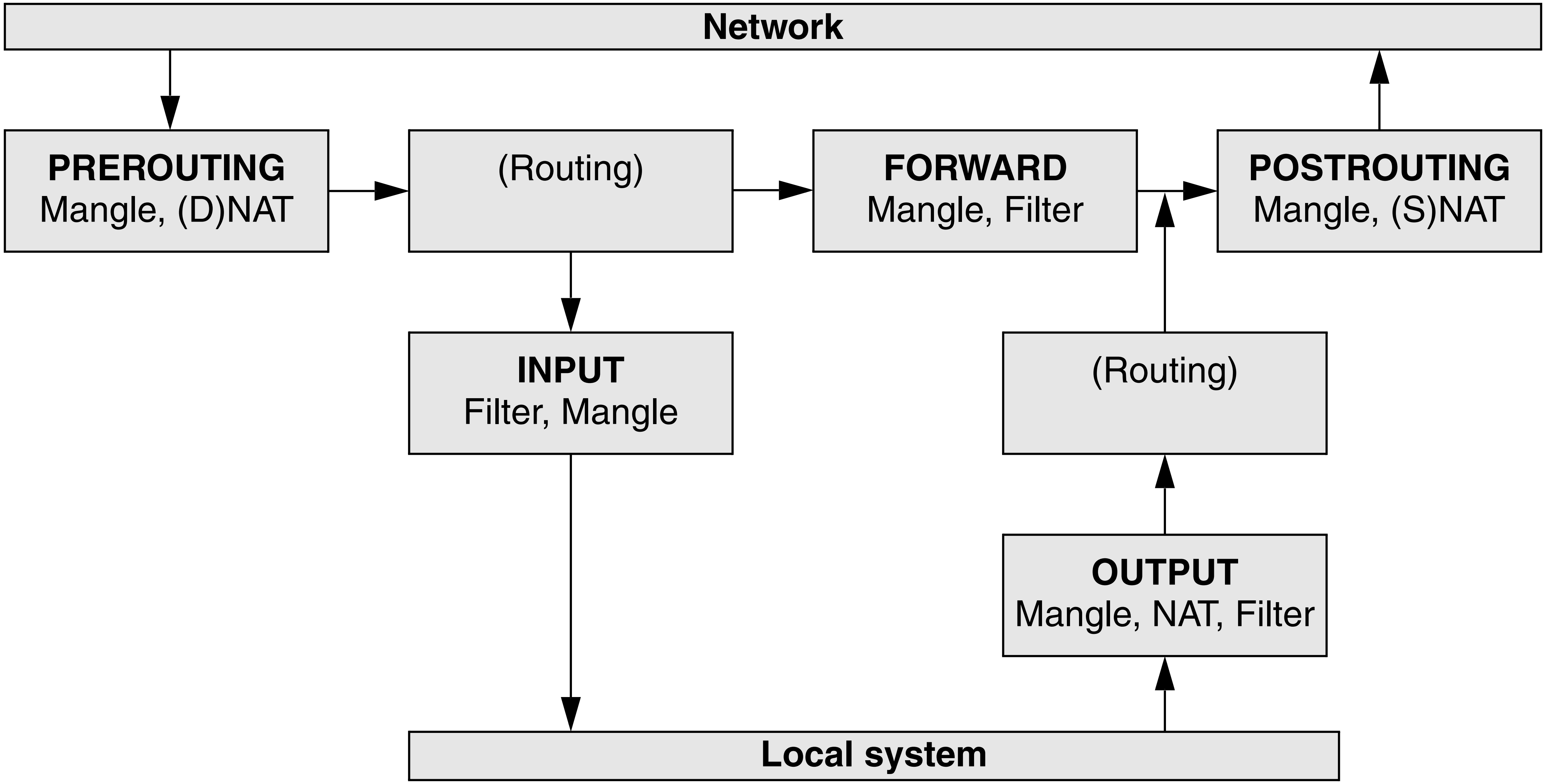
IP Tables 3

- The NAT Table, cont.:
  - **SNAT (source NAT)** alters the source IP address of the first outbound packet in a connection so it appears to come from a fixed IP address (i.e. a firewall address)
  - **MASQUERADE** checks for an IP address to apply to each outbound packet, making it suitable for use with dynamic IP addresses such as those provided by DHCP
- The Mangle Table:
  - Used exclusively to alter the TOS (type of service), TTL (time to live), and MARK fields in a packet. Builtin chains are PREROUTING and OUTPUT

IP Tables 4

- State
    - Connections classified as NEW, RELATED, INVALID or ESTABLISHED
- Jump or Target
    - Specifies the action the kernel takes if a network packet matches all the match criteria

Packet Filtering Diagram



Network

PREROUTING
Mangle, (D)NAT

(Routing)

FORWARD
Mangle, Filter

POSTROUTING
Mangle, (S)NAT

INPUT
Filter, Mangle

(Routing)

OUTPUT
Mangle, NAT, Filter

Local system

IP Tables Utility

- The **iptables** utility is a tool that manipulates rules in the kernel
- Iptables commands can be put in a script and run script each time the system boots (called from **/etc/rc.d/rc.local**) or you can put the arguments in **/etc/sysconfig/iptables**
- The **iptables-save** utility copies packet filtering rules from the kernel to standard output
- The **iptables-restore** utility copies rules from standard input, as written by iptables-save, to the kernel

IP Tables Commands

- --append or –A, Adds rule(s) specified by rule-specifications to the end of chain
- --delete or -D, Removes one or more rules from chain
- --insert or –I, Adds rule(s) specified by rule-specifications and target to the location in chain specified by rule-number
- --list or –L, Displays the rules in chain
- --flush or -F, Deletes all rules from chain
- --policy or –P, Sets the default target or policy builtin-target for the builtin chain
- --help or -h, Displays help

IP Tables Match Criteria

- --protocol or -P [!], Matches if the packet uses the protocol
- --source or -S, Matches if the packet came from address
- --destination or -D, Matches if the packet is going to address
- --in-interface or -i, Matches if iface is the name of the interface the packet was received from
- --out-interface or -o, Matches if iface is the interface the packet will be sent to
- --destination-port or --dport, Matches a destination port number, service name, or range of ports (1025:, 80:88)
- --source-port or --sport, Matches a source port number, service name, or range of ports
- --state, ESTABLISHED, INVALID, NEW, and RELATED

IP Tables Targets

- ACCEPT, Causes the packet to leave the chain
- DNAT, Rewrites the destination address of the packet
    - --to-destination ip[-ip][:port-port]
- DROP
- LOG
- MASQUERADE, Similar to SNAT (below) with —to-source, except that it grabs the IP information from the interface on the specified port
- REJECT, Similar to DROP, except it notifies the sending system that the packet was blocked
- SNAT, Rewrites the source address of the packet (--to-ports port[-port])

Internet Sharing

- Enable Routing:
    - sysctl -w net.ipv4.ip_forward=1
- Add Firewall Rules:

```
iptables -A FORWARD -i eth0 -o eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
iptables -A FORWARD -j LOG
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

IP Tables Examples 1

Flush all Rules

```
iptables -F
```

Set Default Policies

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

SSH Rules

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.10.0/24 --dport ssh -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp -d 192.168.10.0/24 --sport ssh -m state --state ESTABLISHED -j ACCEPT
```

SMTP Rules

```
iptables -A INPUT -i eth1 -p tcp --dport smtp -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth1 -p tcp --sport smtp -m state --state NEW,ESTABLISHED -j ACCEPT
```

IP Tables Examples 2

Internet DNS Rules

```
iptables -A INPUT -i eth1 -p udp --dport domain -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth1 -p tcp --dport domain -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth1 -p udp --sport domain -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth1 -p tcp --sport domain -m state --state NEW,ESTABLISHED –j ACCEPT
```

DNAT Rules

```
iptables -A PREROUTING -t NAT -p tcp --dport 25 -j DNAT --to-destination 192.168.0.33:25
iptables -A PREROUTING -t NAT -p tcp --dport 80 -j DNAT --to-destination 192.168.0.34:80
```

IP Tables Examples 3

Log and Drop Fragmented Traffic

```
iptables -A INPUT -f -j LOG --log-prefix "IPT: Frag "
iptables -A INPUT -f -j DROP
```

Prevent SYN Flooding

```
iptables -A INPUT -i $EXT_INTER -p tcp --syn -m limit --limit 5/second -j ACCEPT
# Log and Drop Traffic in the INVALID state
iptables -A INPUT -m state --state INVALID -j LOG --log-prefix "IPT: INV_STATE "
iptables -A INPUT -m state --state INVALID -j DROP
```

Prevent SYN Flooding

```
iptables -A INPUT -i $EXT_INTER -p tcp --syn -m limit --limit 5/second -j ACCEPT
```

Log and Drop Traffic in the INVALID state

```
iptables -A INPUT -m state --state INVALID -j LOG --log-prefix "IPT: INV_STATE "
iptables -A INPUT -m state --state INVALID -j DROP
```

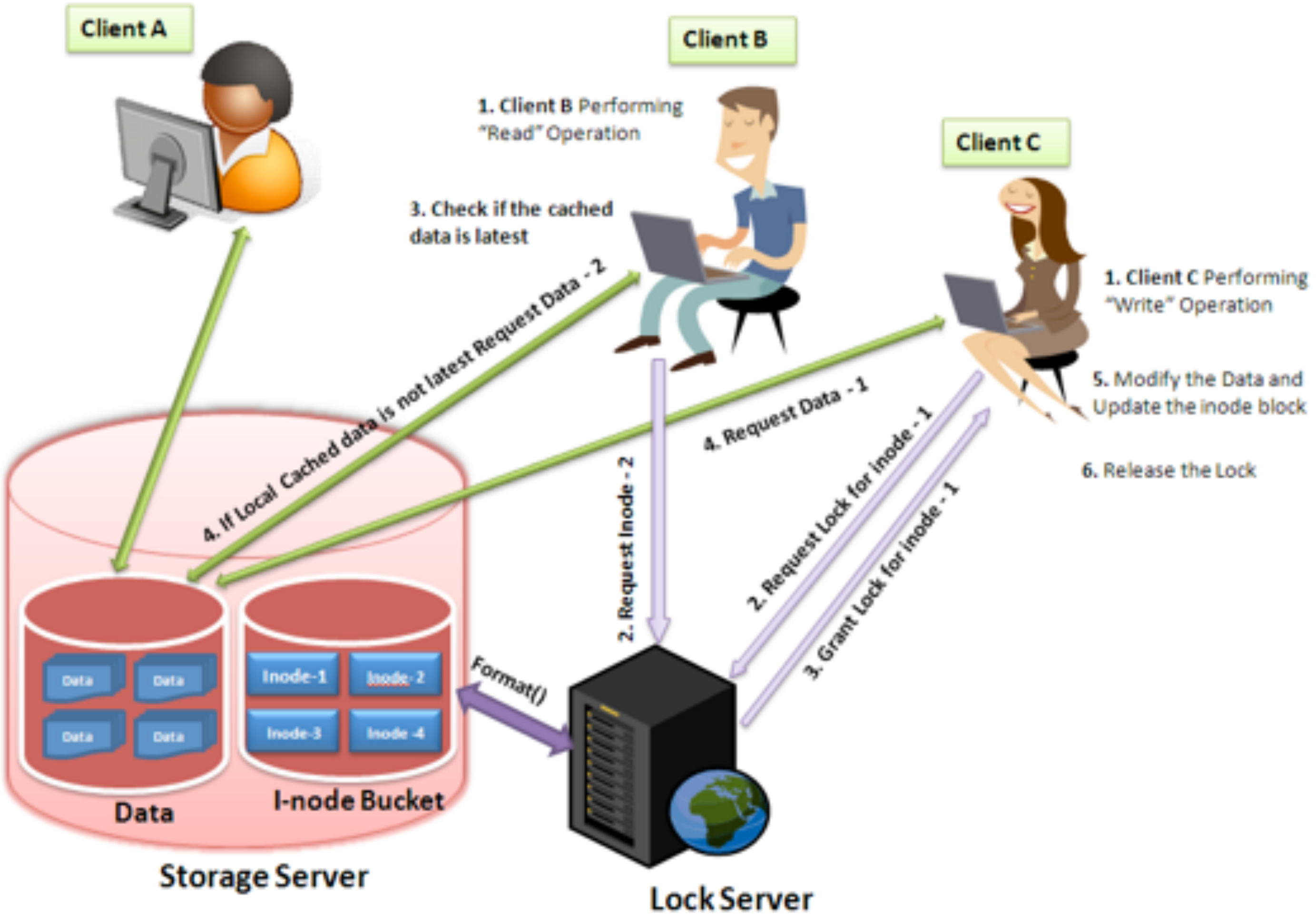
NFS

NFS, pt. 1

- The **NFS** (Network Filesystem) protocol allows a server to share selected local directory hierarchies with client systems

- Files the server appear as if they are present on the local system

- NFS allows you to store a copy of a program on a single system and give other users access to it over the network


Client A
Client B
1. Client B Performing "Read" Operation
3. Check if the cached data is latest
4. If Local Cached data is not latest Request Data - 2
Client C
1. Client C Performing "Write" Operation
4. Request Data - 1
5. Modify the Data and Update the inode block
6. Release the Lock
2. Request Inode - 2
2. Request Lock for inode - 1
3. Grant Lock for inode - 1
Format()
Data
Data
Data
Data
Inode-1
Inode- 2
Inode-3
Inode -4
Data
I-node Bucket
Storage Server
Lock Server

/etc/exports file

- Each line in the exports file has the following format:
    - export-point client1(option-list) [client2(option-list) ...]
- export-point is the absolute pathname of the directory
- client1-*n* are the names or IP addresses of one or more clients, separated by SPACEs
- option-list is a comma-separated list of options

```
/home/public 172.16.192.150(rw,sync)
/home/zach 172.16.192.150(rw,sync)
```

Exports Options, pt. 1

- auth_nlm (no_auth_nlm) or secure_locks (insecure_locks) - require authentication of lock requests

- ro (rw)

- secure (insecure) - requires NFS requests to originate on a privileged port so a program running without root privileges cannot mount a directory hierarchy, default secure

- no_subtree_check (subtree_check) - checks subtrees for valid files, default no_subtree_check

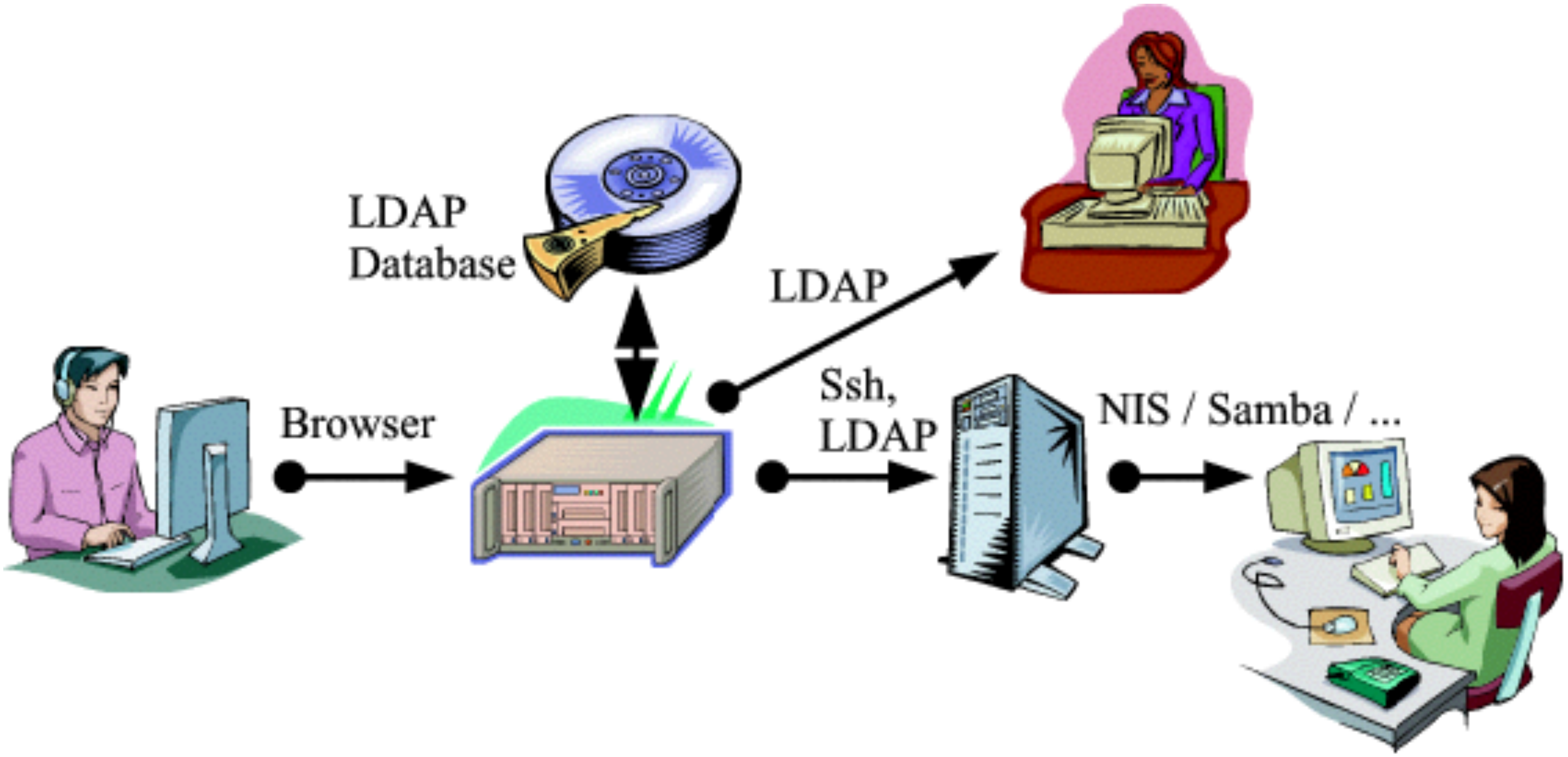
Password Sync

- Users might not have the same ID numbers on both systems allowing them to have access to others files
- The **root_squash** option to maps the ID number of the root account on a client to the nfsnobody user (UID 65534) on the server
- The **all_squash** option to maps all NFS users on the client to nfsnobody (UID 65534) on the server
- NIS Server (authentication) users automatically have the same UIDs on both systems
- Without NIS (or LDAP), you must synchronize the passwd files on all systems manually

Exportfs

- The **exportfs** utility maintains the /var/lib/nfs/etab file
- When mountd is called, it checks this file to see if it is allowed to mount the requested directory hierarchy
- Without any arguments, exportfs reports which directory hierarchies are exported to which systems:
    - exportfs
- To re-export the entries in /etc/exports and remove invalid entries from /var/lib/nfs/etab so etab is synchronized with /etc/exports:
    - exportfs -r
- Other options:
    - -a (all), -u (unexport), -f(flush)

LDAP



Lightweight Directory Access Protocol is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network

WEB and FTP

FTP 1

- FTP (File Transfer Protocol) is a method of downloading files from and uploading files to another system using TCP/IP over a network

- FTP sites can be public, allowing **anonymous** users to log in and **download/upload** software and documentation or private, requiring a username and password

- FTP is not a secure protocol: All usernames and passwords exchanged in are sent in cleartext, data exchanged over an FTP connection is not encrypted

- FTP is best used for downloading public files

FTP 2

- FTP uses two connections: one for control (port 21) and one for data transfer (port 20)
- A client can ask an FTP server two different connections:
  - a **PASV** (passive—the default, client initiated)
  - a **PORT** (active, server initiated)
- Passive connections are more common because a client behind a NAT can connect to a passive server and it is simpler to program a scalable passive server
- To connect to ftp server
  - ftp (or sftp) hostname
  - commands: put/mput, get/mget, status, help, !pwd, lcd, prompt

Vsftpd

- Install package **vsftpd**
- The configuration file for vsftpd, **/etc/vsftpd/vsftpd.conf**
- Other options:
    - listen, set the server to standalone mode (no xinetd)
    - listen_address and listen_port, IP address and Port
    - max_clients and max_per_ip, # of clients, 0 is unlimited
    - userlist_enable, userlist_deny, userlist_file
    - local_enable, Yes permits local users to login
    - anonymous_enable

FTP Security

- The anonymous account is mapped to the **ftp**
- The anonymous user works in the **/var/ftp** directory
- By default, anonymous users are placed in a chroot jail for security; local users are not
- chroot_local_user, puts each local user in a chroot jail whose root is the user's home directory (chroot_list_enable, selected local users in chroot jails)
- Change the following SELINUX Boolean to allow users to connect local directory:
  - setsebool -P ftp_home_dir 1

Uploading Files

- The write_enable parameter must be set to YES to permit local users to upload files
- The default local_umask is set to 022, giving uploaded files 644 permissions
- Other upload options:
  - anon_mkdir_write_enable
  - anon_upload_enable
  - anon_other_write_enable (delete and rename files)
- Logs written to **/var/log/xferlog** (xferlog format)

wget

- The **wget** utility (wget package) is a noninteractive, command-line utility that retrieves files from the Web using HTTP, HTTPS, or FTP
- With the —recursive (–r) option, wget downloads the directory hierarchy under the URI you specify
- The —background (–b) option runs wget in the background and redirects its standard error to a file named wget-l
- Examples:
    - wget http://fedoraproject.org/index.html
    - wget --recursive --background http://fedoraproject.org/index.html

Apache, pt. 1

- **Apache** is the most popular Web server on the Internet
- The Apache Software Foundation (1999) grew out of the Apache Group, which was established in 1995 to develop the Apache server
- Apache uses external **modules** to increase load-time flexibility and allow parts of its code to be recompiled without recompiling the whole program
- Modules can do a number of things including process scripts written in Perl, PHP, Python, and other languages, as well as use several different methods to authenticate users
- An Apache **child** process exists to handle incoming client requests

Apache, pt. 2

- An Apache server normally uses **TCP port 80**
- A secure server uses **TCP port 443**
- When SELinux is set to use a targeted policy, httpd is protected by SELinux
- Apache serves content on privileged ports, you must start it running with root privileges, but spawns processes that run as the user and group **apache**
- Configuration file for apache is /etc/httpd/conf/httpd.conf
- Configuration directives are lines in a configuration file that control some aspect of how Apache functions

Apache Directives, pt. 1

- Listen, Specifies the IP address and port that Apache listens for requests on
  - Listen 80
  - Listen 192.168.1.1:8080
  - Listen 192.168.1.2:443
- Redirect, tells the client to fetch a requested resource from a different, specified location.
  - Redirect /www.example.com/pictures http://pictures.example.com/
- ServerAdmin, Sets the email address used in mailto: links on error pages
  - Server Admin max@example.com
- ServerName, Specifies the server's name and the port it listens on
  - ServerName www.example.com:80

Apache Directives, pt. 2

- DocumentRoot, Points to the root of the directory hierarchy that holds the server's content
    - Can be anywhere, but the apache user must have read access on files and execute on directories
    - DocumentRoot /srv/www
- UserDir, Allows users to publish content from their home directories
- When you do not specify a dirname, Apache publishes content in ~/public_html
- http://www.example.com/~user1
    - UserDir disabled
    - UserDir enabled user1 user2 user3
    - UserDir web

Apache Directives, pt. 3

- DirectoryIndex, Specifies which file Apache serves when a user requests a directory
  - DirectoryIndex index.html
- MaxClients - maximum number of child processes (or threads) that will be launched to serve requests
- MaxRequestsPerChild - Process dies after a specified number of requests
- MaxSpareServers - Specifies the maximum number of idle processes
- StartServers - Specifies the number of child processes at the start
- ErrorLog, LogLevel
- Alias, Maps a URI to a directory or file
  - Alias /pix /usr/local/pix

Apache Directives, pt. 4

- ErrorDocument
  - ErrorDocument 403 "Sorry, access is forbidden."
  - ErrorDocument 403 /cgi-bin/uh-uh.pl
  - ErrorDocument 403 http://errors.example.com/not_allowed.html
- LoadModule,
  - LoadModule module filename
- Options, Controls server features by directory
  - None, All, ExecCGI, FollowSymLinks, Includes (SSI)
- ScriptAlias, Maps a URI to a directory or file and declares the target to be a server (CGI) script
  - ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

Contexts

- **server config**, in the httpd.conf file or included files only, but not inside <VirtualHost> or <Directory> containers unless so marked
- **virtual host**, inside <VirtualHost> containers in the httpd.conf file or included files only
- **directory,** inside <Directory>, <Location>, and <Files> containers in the httpd.conf file or included files only
- **.htaccess**, in .htaccess files only

Containers, pt. 1

- Containers, or special directives, are directives that group other directives

- <Directory>

```
<Directory /var/www/html/corp>
              Deny from all
              Allow from 192.168.10.
              AllowOverride All
           </Directory>
```

- <Files>

```
< Files ~ "^\.ht">
              Order allow,deny
              Deny from all
              Satisfy All
</Files>
```

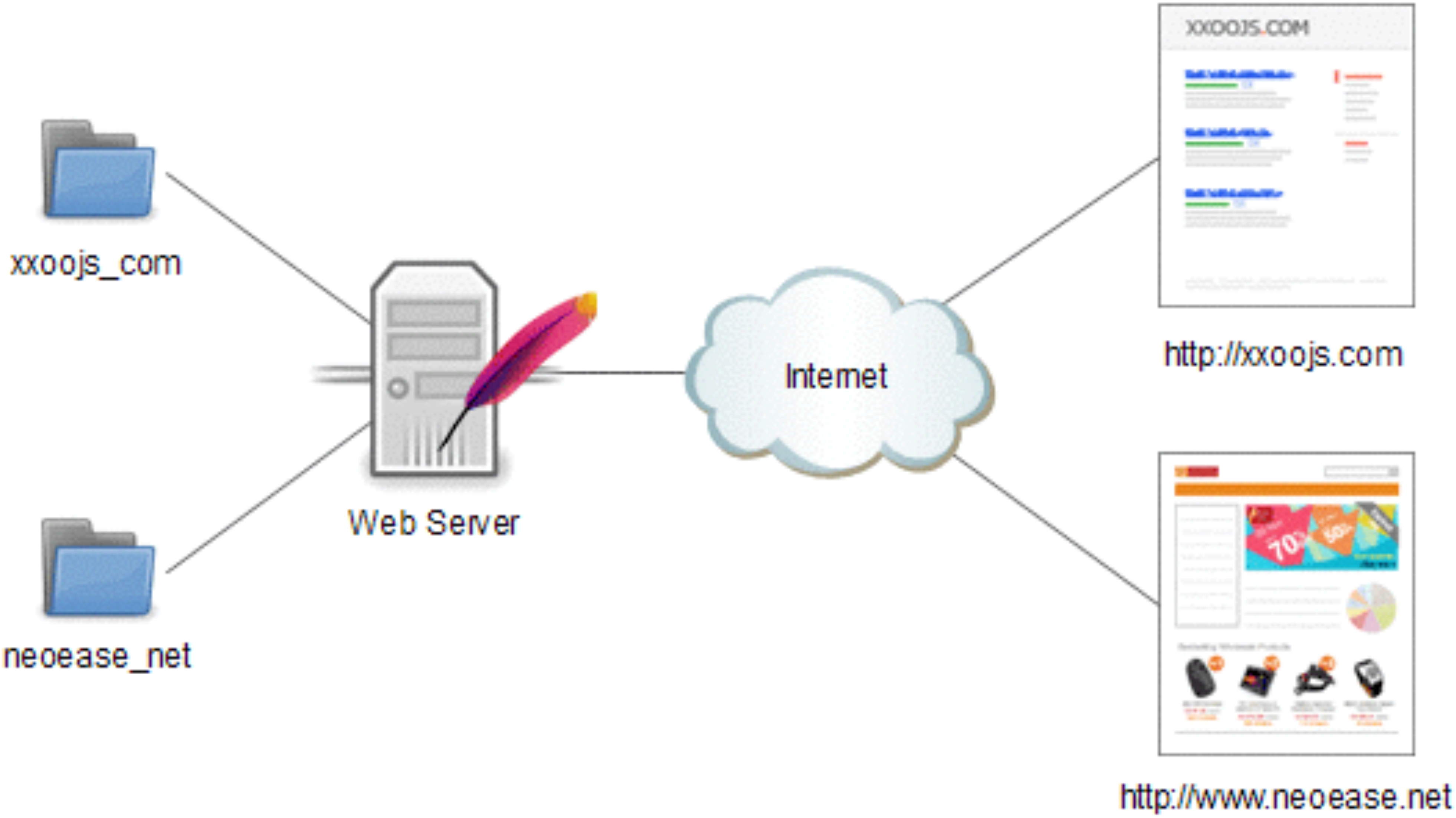
Virtual Hosts, pt. 1

- <VirtualHost>, Applies directives to a specified virtual host

-
  ```
  <VirtualHost *>
          ServerName example.com
          ServerAlias www.example.com
          ServerAdmin webmaster@example.com
          DocumentRoot /var/www/html/example.com
          CustomLog /var/log/httpd/example.com.log combined ErrorLog /var/log/
  httpd/example.com.err


   </VirtualHost>
  ```

Virtual Hosts, pt. 2



SSL

- SSL (Secure Sockets Layer), which is implemented by the mod_ssl module, has two functions:
    - It allows a client to verify the identity of a server
    - it enables secure two-way communication between a client and a server
- SSL is used on Web pages in conjunction with forms that require passwords, credit card numbers, or other sensitive data
- Apache uses the HTTPS (port 443) protocol—not HTTP—for SSL communication

Performance Testing

- Apache Bench
- http_load

Scalability and High Availability

- **High availability** really means “less downtime”
  - One common goal for systems and application is the “5 9’s” or 99.999% uptime (5 minutes per year)
- **Scalability** is the ability to add capacity by adding resources
  - A system or application may have to support 1,000 reads per minute during normal operation, but during **peak utilization** have to scale to 10,000

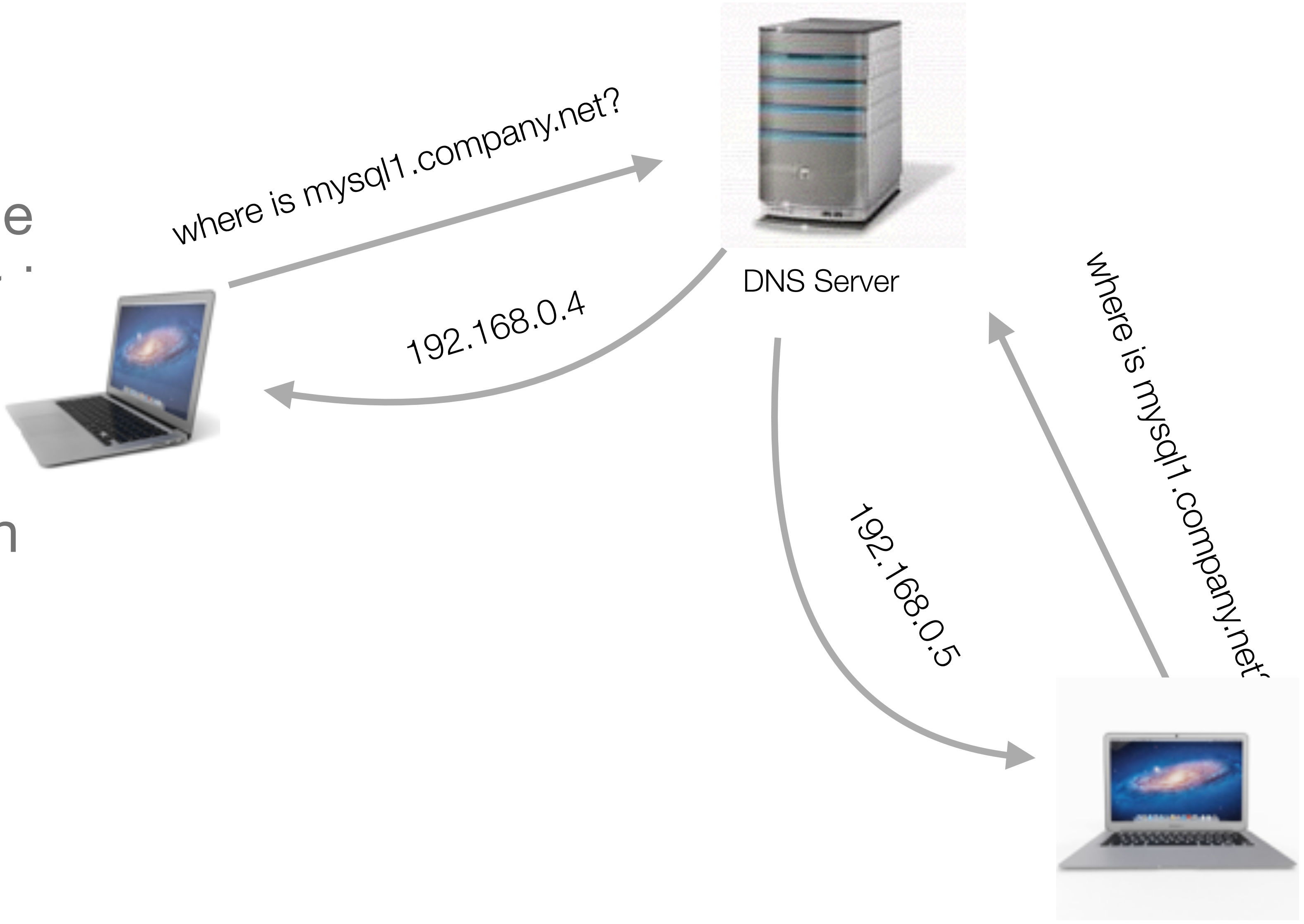
DNS Road Robin

- Round robin is a local balancing mechanism used by DNS servers to share and distribute network resource loads

- Multiple records with the same hostname but point to differer IP addresses

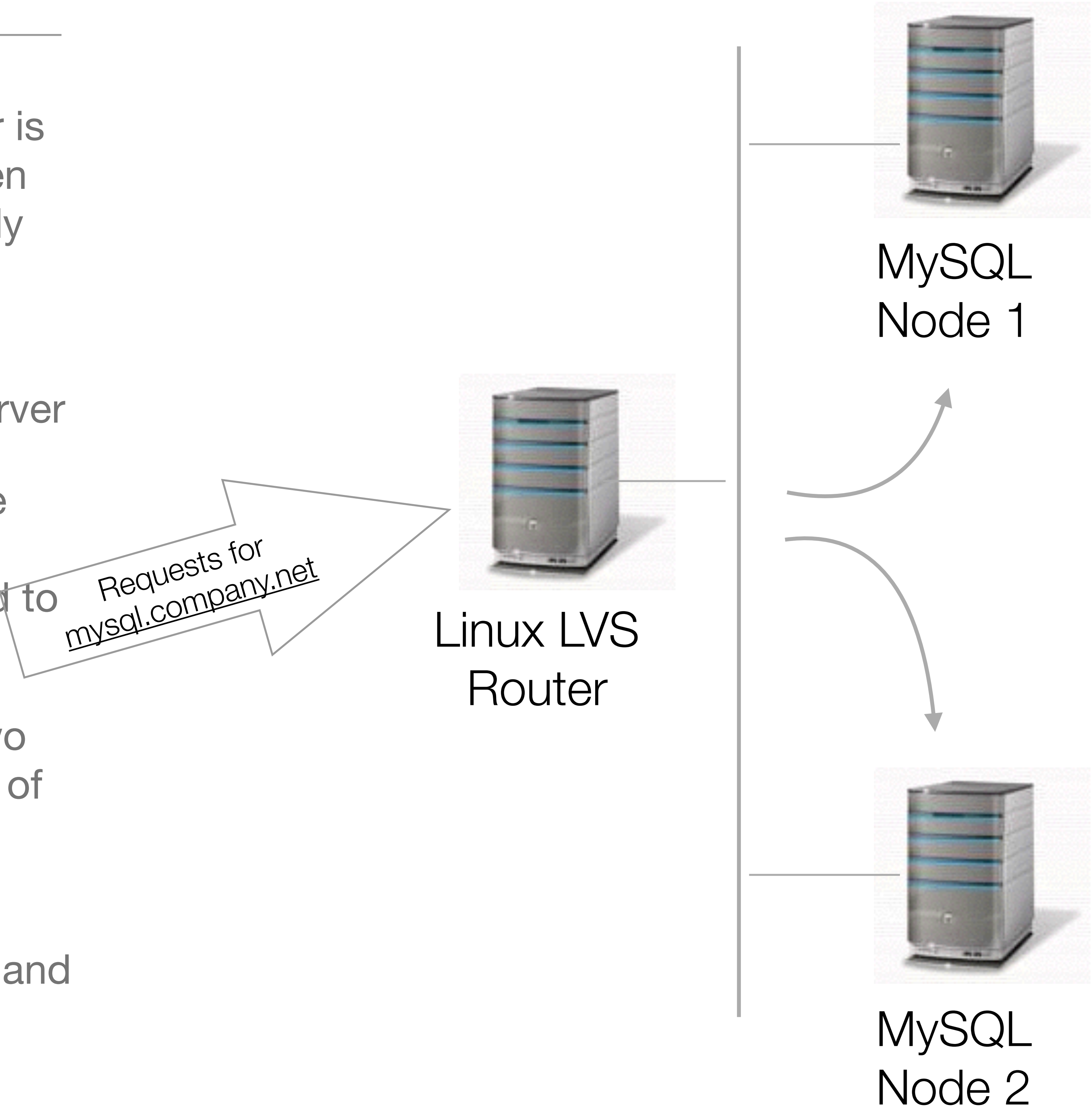
- Client requests are returned in a “round robin” style

```
; company.net zone file
;
mysql1    IN  A   192.168.0.4
          IN  A   192.168.0.5
          IN  A   192.168.0.6
```



LVS

- A **Linux Virtual Server (LVS)** cluster is a collection of servers that have been specially configured to provide highly available services
- Service requests arriving at an LVS cluster are addressed to a virtual server
- A **virtual server** has a domain name that is associated with a floating IP address, and which can be migrated to a different host
- An LVS cluster consists of one or two router nodes and a variable number of servers
- LVS supports round robin, least-connections, weighted round robin, and weighted least-connections

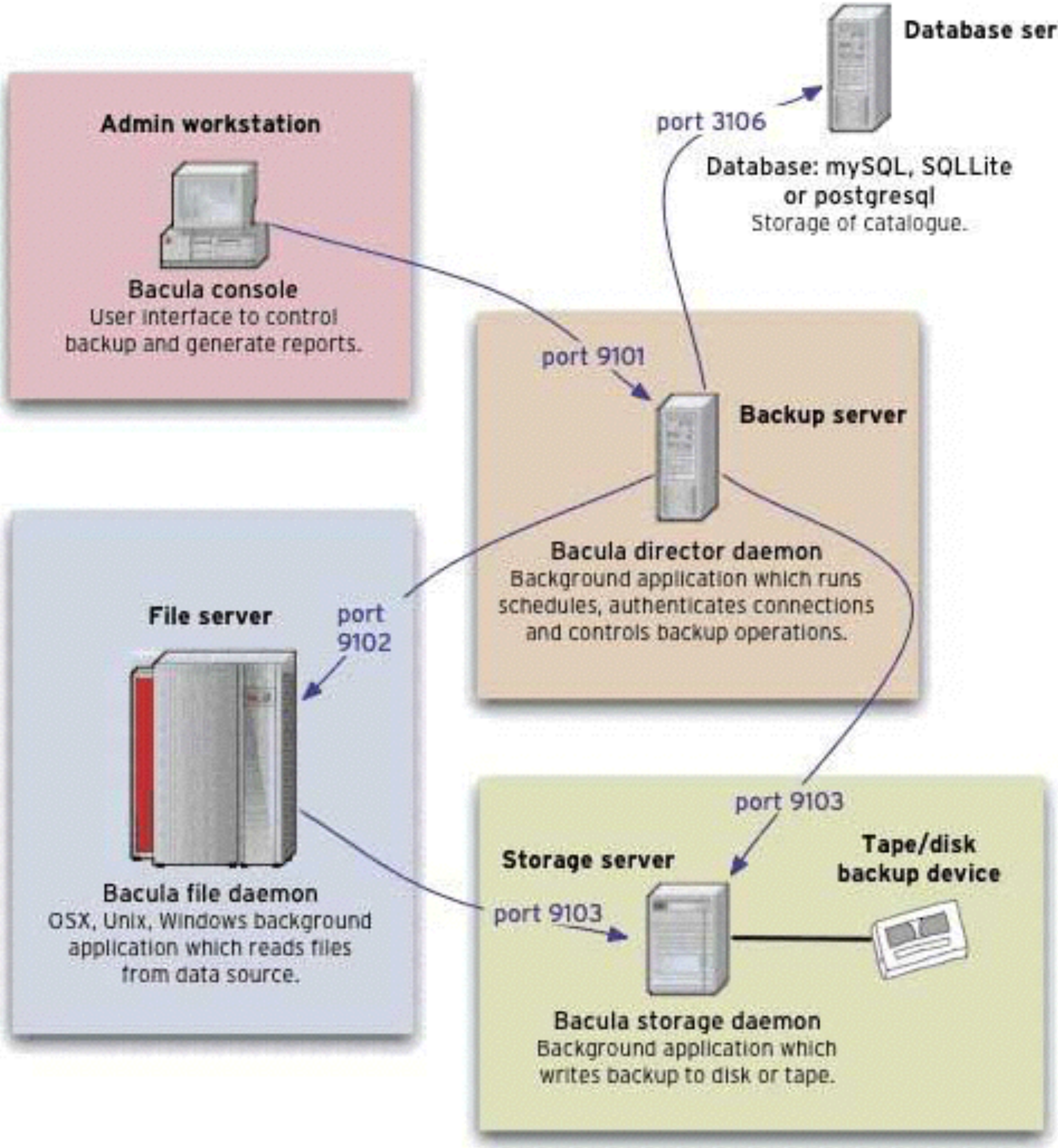


Incremental/Differential Backups

- A **differential backup** is a backup of everything that has changed since the last full backup
- An **incremental backup** contains everything that has changed since the last backup of any type
  - Incremental backups add complexity, risk, and time to recovery
- A **Full backup** is an entire backup of the database
- Very common run a full back on tape and diff/inc backups to disk
- Require the use of a Enterprise Backup solution
- Popular Linux backup software: cpio, tar, rsync, bacula

Enterprise Backup: Bacula

- Director — supervises all of Bacula
- Console — Used to communicate with the Director
- File — On the client
- Storage — reads and writes to the storage space
- Catalog — Database
- Monitor — Used to keep track of the status of the various Bacula tools


Database server
port 3106
Database: mySQL, SQLLite or postgresql
Storage of catalogue.
Admin workstation
Bacula console
User interface to control backup and generate reports.
port 9101
Backup server
Bacula director daemon
Background application which runs schedules, authenticates connections and controls backup operations.
File server
port 9102
Bacula file daemon
OSX, Unix, Windows background application which reads files from data source.
port 9103
Storage server
Tape/disk backup device
port 9103
Bacula storage daemon
Background application which writes backup to disk or tape.

Final Exam

- 25 multiple choice questions
- Study the following presentations:
    - Class 2
    - Class 5
    - Class 6
    - Class 7

Lab Assignments

- Configure SSH
  - Configure ssh so that you can authenticate automatically (without password) when logging in to each host
  - Use SCP and rsync to copy directories between hosts
- TCP Wrappers
  - Configure TCP Wrappers only allow SSH from your host to one of your VMs
  - Test by making a connection from your host and one of the other VMs
  - View the end of the /var/log/secure for messages
- IP Tables
  - Add IP tables rules to allow ssh and smtp incoming connections to one of your VMs only from your host computer, back them up to a file, and configure the machine to load the rules on boot
  - Test to verify the rules are working by viewing by trying to connect from another host

Homework

- Read Chapter 19, 24, 26
- Do page 403, Exercises 1, 3, 8, 9
- Do page 699, Exercises 6
- Do page 914, Exercises 1-4

Contact

Ian Robert Blair, MSc.

ian.robertblair@icloud.com

QQ: 2302412574