



# Host Security

**Gaurab Raj Upadhaya**

Packet Clearing House

[gaurab@pch.net](mailto:gaurab@pch.net)

**Vicky Shrestha**

WorldLink Communications

[vickysh@wlink.com.np](mailto:vickysh@wlink.com.np)

---



- Introduction
  - Administration
  - Securing Services
  - Securing FreeBSD
  - File systems
  - System Logging
  - Backup
  - Server Monitoring
  - Network Scanning
  - Firewall
  - Intrusion Detection System
-



"A computer is secure if you can depend on it  
and its software behave as you expect"

-- Practical Unix and Internet Security

---



- **Confidentiality**
    - The information Stored must be protected from those who are not authorized to view them
  - **Data Integrity**
    - The information must be protected from being altered or deleted without the permission or knowledge of the owner
  - **Availability**
    - The information must be available when the authorized user needs them. Failure to deliver information when needed is equivalent to having no information at all
-



- Consistency
    - The system must behave as expected. A radical change in the behavior of software due to upgrades, bug fixes can be disastrous
  - Control
    - A system administrator must make sure that there are no unauthorized users in the system. He should regulate access to the system
  - Monitoring
    - Constant Monitoring of system is necessary to detect intrusions and other security issues
  - Audit
    - Proper audit must be conducted for a damage made to determine what was done, by whom and what was affected
-



- **Network Connectivity**
    - Your host is connected to the network, which makes it more accessible to network based attacks
  - **Denial of Service Attacks**
    - Attacks on network infrastructure or Legitimate services, so that service is denied or unavailable
  - **Unauthorized Intrusions**
    - Unauthorized Intrusions are break-in performed by people who are not authorized to access the network resources
  - **Buffer Overflow**
    - Resulting from coding errors and lack of proper validation of input, an attacker can craft special data that will overwrite the buffer allocated by a program and run arbitrary codes
-



- Brute Force/Dictionary Attacks
    - Brute Force/Dictionary attacks are performed to find out the credentials necessary for logging in or to gain unauthorized access
  - Packet Sniffer
    - Packet sniffer such as tcpdump, ethereal are used to eavesdrop on an existing connection. Telnet and ftp sends clear text passwords that can be sniffed by these packet sniffers
  - Spoofing Attacks
    - Spoofing Attacks means forging source address of trusted hosts to access network services
  - Web Based attacks
    - Web based attacks include SQL injection, code injection (cross-site scripting), result of improper validation of input in dynamic websites
-



- **BackDoor**
    - Intruders generally leaves some sort of backdoor on compromised machines. If not detected it gives an easy way for the intruders to gain privileges in future
  - **Social Engineering**
    - Social Engineering means trying to gain information/credentials, that can compromise network security, from people who are authorized to access the network resources.
    - Social Engineering heavily depends upon Human interaction
-





- Insecure services running on a host
  - Services with security holes
  - Insecure configuration of Services
  - Presence of Malicious codes in the system
  - Use of Weak or Default passwords
  - Running vulnerable services such as telnet, rsh, r\*, etc
  - Running Services where authentication information is send in unencrypted format
  - Poorly protected hosts/networks without firewall, IDS
  - Insufficient validation of input in applications
  - Lack of importance given to security
-



## Common Motivations for Security Breaches

---

- A security breach can be performed by hired professionals to steal confidential data
  - A curious and intelligent computer savvy person might like to get into your system just for fun
  - A very secure system can serve as a challenge for crackers and can earn him respect after a successful break-in
  - Ex-Employees after being fired can also breach security for revenge
  - A computer user might breach security due to total ignorance
-



---

# Securing Services

---



- Install only the required services
  - Turn off everything else and enable only required services
  - Restrict access to running services to only those who should have access
  - Use TcpWrappers to restrict access to services
  - Update or Patch the programs regularly to fix potential or known security holes
  - Verify what services are actually running using ps and netstat/sockstat commands
  - Use Firewall to protect the services
  - Use chroot/jail where possible
  - Limit the number of processes a service can fork
  - Configure limits such as cpu, mem, file descriptors etc from “/etc/login.conf”
-



- It is advisable not to use telnet server and disable it all together
  - There are better alternatives to telnet, the popular one being ssh
  - SSH or Secure Shell provides a secure alternative to telnet
  - During telnet, the transmitted data are unencrypted meaning your passwords and session data are sent in plain text
  - Anyone running a sniffer on the network can find out what was transmitted and received during a telnet session
-



- User names and passwords are send in clear text
  - Anonymous logins should be disabled unless required
  - chroot should be used with ftp server to lock down users to their home directory only
  - ftp access should be denied to system users such as root,bin,mail and users having shell access
  - Ftp users' shell should be modified to '/bin/nologin' or '/bin/false'
  - Use of sftp should be promoted
-



- Portmap is not needed unless you are running nfs server or NIS
  - Proper protection from tcpwrapper and firewall should be provided if portmap must be on
  - R services such as rlogin, rsh, rcp should be disabled
  - R services can be replaced by SSH ssh, sftp, scp
-



- NFS (Network File Service) is the most frequently used method of sharing access to file system between Unix systems
  - System administrators need to be careful about how they implement NFS and be aware of the vulnerabilities associated with various daemons which collectively make up the NFS service including nfsd, mountd, statd, lockd
  - Regular updates/patches should be done for these daemons
  - NFS Services must be protected via Tcpwrapper and Firewall
  - Use `nfs_reserved_port_only="Yes"` to provide nfs services on a secure port
-





- Tcpwrapper is an utility that intercepts packets and authorizes it before delivering it to the final application
- It can be configured by editing the file:
  - /etc/hosts.allow
- The format of the file is
  - Service : List of Ips/Hosts : allow/deny
- It is recommended to allow only selected services and deny the rest as show in the following example
  - ALL : localhost : allow
  - sshd: 192.168.0. : allow
  - ALL : ALL : deny
- However all services that can run on Unix cannot be protected via tcpwrapper, some programs needs to be compiled with tcpwrapper support while others do not use it at all



- chroot is a way of limiting a service (or user) environment to a particular directory or file system
  - Services such as pure-ftp, proftpd and bind have features for chrooting
  - Services not having such support can also be run in a chroot environment but with some extra work
  - Services running inside a chroot environment cannot access files outside of the chrooted file system
  - FreeBSD also offers a jail command that is similar to chroot but much more advanced and accurate
  - Jail can be used to build virtual systems inside the real Operating system
-



- You should also make sure that the programs installed on your system are up to date
- You need to watch out for new bugs that are discovered and apply security patches as soon as possible
- A good practice is to subscribe to vendor's security mailing lists and other security mailing lists
- In FreeBSD you can use `cvsup` to sync your `src`, `ports` and `docs` tree
- You can also use `portupgrade` to upgrade packages built from ports
- You should use `portaudit` for auditing the installed packages from ports for security issues
- ~~`freebsd-update` can be used for binary~~



- When installing softwares, you should verify the integrity /authenticity of the software package
  - There are different tools available to verify that :
    - md5
    - gpg
  - The FreeBSD ports system verifies the md5 and size of the packages that you download
  - If portaudit is installed it can also verify the ports for security issues
  - Use portaudit -Fd to download the latest audit database and install it
  - Use portaudit -a to see the list of vulnerable installed packages
-



- cvsup is a software package for updating the source, ports and docs tree in FreeBSD
  - cvsup-without-gui is recommended for servers
  - example configurations for cvsup can be found in “/usr/share/examples/cvsup”
  - you can start with stable-supfile file
  - cvsup can be run as follows
    - cvsup -g -L 2 supfile
      - -g Disables GUI
      - -L 2 Logging level 2
-



# Securing FreeBSD





---

`/etc/rc.conf`

---



- `nfs_server_enable="NO"`
    - Disable NFS Server
  - `nfs_client_enable="NO"`
    - Disable NFS Client
  - `rpcbind_enable="NO"`
    - Disable portmap
  - `syslogd_enable="YES"`
    - Enable Syslog Daemon
  - `syslogd_flags="-s"`
    - Disable logging from remote hosts
    - "-ss" will disable opening udp 514 port
    - It will also disable logging to central logging server
  - `sshd_enable="YES"`
    - Enable sshd server
  - `inetd_enable="NO"`
    - Disable inetd daemon
-





- `accounting_enable="YES"`
    - Enable Process Accounting
  - `clear_tmp_enable="YES"`
    - Enable cleaning up /tmp directory on startup
  - `enable_quotas="YES"`
    - Turn on quotas on startup
  - `check_quotas="YES"`
    - Check quotas on startup
  - `kern_securelevel_enable="YES"`
    - Enable Kernel Secure Level
  - `kern_securelevel="3"`
    - The kernel secure level is set to 3 during boot
  - `log_in_vain="YES"`
    - Enable logging of connection attempts to closed ports
  - `sendmail_enable="NO"`
    - Only listen on loopback interface
-



- `tcp_drop_synfin="YES"`
    - Drops packets with SYN and FIN flags set
    - Can defeat OS detection by scanners
    - Also breaks RFC Compliance
    - Not recommended on web servers
    - kernel option `TCP_DROP_SYNFIN` must be enabled
  - `tcp_extensions="NO"`
    - Disable RFC1323 TCP extensions
  - `accept_sourceroute="NO"`
  - `forward_sourceroute="NO"`
    - Disable Source Routes
-



- `icmp_drop_redirect="YES"`
    - Drops/Ignore icmp redirect messages
  - `icmp_log_redirect="YES"`
    - Logs icmp redirect messages
  - `icmp_bmcastecho="NO"`
    - Don't reply to Broadcast ping requests
-



- `firewall_enable="YES"`
    - Enable Firewall (IPFW)
  - `firewall_script="/etc/rc.firewall"`
    - Script containing the firewall rules
  - `firewall_quiet="YES"`
    - Don't echo the rules as you insert
    - "ipfw -q" is necessary when you rerun firewall script remotely
  - `firewall_logging="YES"`
    - Enable logging from firewall
-



---

`/etc/sysctl.conf`

---



- `net.inet.tcp.blackhole=2`
    - Don't send RST packets for tcp connection on closed port
  - `net.inet.udp.blackhole=1`
    - Ignore packets send to closed udp port
  - `security.bsd.see_other_uids=0`
    - Don't let users see each other's processes
    - root can however see all the processes
  - `net.inet.ip.check_interface=1`
    - Check that the incoming packets arrive on the correct interface
  - `net.inet.tcp.syncookies=1`
    - Enable SYN Cookies
  - `net.inet.ip.fw.verbose=1`
    - Enable logging from firewall with log rules
-



- `net.inet.ip.fw.verbose_limit=200`
    - Limit the logs messages from firewall log rules to 200
  - `net.inet.icmp.icmplim=200`
    - Limit icmp messages from the host to 200
    - Helps against DDoS
  - `net.inet.icmp.bmcastecho=0`
    - Don't reply to broadcast ping requests
  - `net.inet.ip.rtxpire=2`
    - Set the route cache entry timeout to 2 seconds
    - Protects from Dos Attack
  - `net.inet.ip.rtminexpire=2`
    - The kernel will never lower the route cache entry timeout below this value
-



---

`/etc/ttys`

---





- You can disallow direct root logins in the terminals by modifying this “/etc/ttys”
  - Just change the word “secure” to “insecure” for all the lines starting with “ttyvX”
  - You can also enable password prompt in single mode by changing the line that begins with console
-



---

`/etc/inetd.conf`

---



- 
- FreeBSD provides you the option of Rate limiting via inetd
  - Syntax of inetd.conf
    - service name
    - socket type
    - protocol
    - {wait | nowait } [/ max-child [/ max-connections-per-ip-per minute[/max-child-per-ip]]
    - user
    - server program
    - server program arguments
  - `echo stream tcp nowait/100/1 root internal`
    - In the above example service echo is allowed to have maximum 100 childs and only 1 connections per IP address per minute is allowed
-



---

# Kernel Secure Level

---



- FreeBSD provides additional security by defining kernel secure levels
  - The kernel can run in Five Different levels of Security
  - The security level can be increased by root
  - The security level can NOT be decreased under any condition
  - If the security level is initially nonzero, then init leaves it unchanged
  - Init raised the level to 1 before going to multi-user mode if the initial level is 0
  - Can be enabled in rc.conf
    - kern\_securelevel\_enable="YES"
    - kern\_securelevel="3"
-



- -1
    - Permanently insecure mode
    - Always run the system in level 0
  - 0
    - Insecure mode
    - Immutable and append-only flags may be turned off
    - All devices may be read or written subject to their permissions
  - 1
    - Secure mode
    - The system immutable and system append-only flags may not be changed
    - disks for mounted file system, /dev/mem, /dev/kmem and /dev/io may not be opened for writing
    - kernel modules may not be loaded or unloaded
-



- 2
    - Highly Secure mode
    - Everything in Secure level 1
    - Disks may not be opened for writing except by mount
    - Will not allow newfs to be run
    - Kernel time changes are restricted to less than or equal to 1 second
  - 3
    - Network Secure mode
    - Everything in Secure level 2
    - Packet filter rules cannot be changed(IPFW,IPF,PF)
    - dummynet or pf configuration cannot be adjusted
-



---

# File System

---





- File and Directory permissions can be basically used to control who gets access to resources
  - Since everything in Unix is a file, you can control virtually everything by means of file permissions
  - Sample File permissions
    - ```
$ ls -l /home/  
total 4  
drwx----- 29 vicky vicky 4096 Feb 9 21:08 vicky
```
  - The First character 'd' denotes that it's a directory. The following 3 characters means read,write and execute access to owner. The last 6 characters are blank(-) meaning access is not allowed to group and other users. It secures my home directory, as only I have access to it
-



- Here is a tabulated form of the permissions

|    |   |                              |           |     |
|----|---|------------------------------|-----------|-----|
| 1  | - | File type                    | Directory | d   |
| 2  | r | read access to owner (vicky) | r         | yes |
| 3  | w | write access to owner        | w         | yes |
| 4  | x | execute access to owner      | x         | yes |
| 5  | r | read access to group         | -         | no  |
| 6  | w | write access to group        | -         | no  |
| 7  | x | execute access to group      | -         | no  |
| 8  | r | read access to others        | -         | no  |
| 9  | w | write access to others       | -         | no  |
| 10 | x | execute access to others     | -         | no  |

---



- Lets take a look at the following permissions

```
- $ ls -l /tmp/  
total 4  
-rwxrwxrwx 4 vicky vicky 4096 Feb 9 22:01 public.txt
```
  - The above file 'public.txt' has been created inside '/tmp' directory with read,write and execute permissions to owner,group and all others
  - Notice the blank '-' as the first character  
This means it is a regular file
  - The user and group is shown on the 3<sup>rd</sup> and 4<sup>th</sup> column when we do 'ls -l'
-



- 
- You can change file permissions using chmod command
    - \$ chmod u+rwx filename
      - Gives the owner full rights, ie., read,write and execute
    - \$ chmod u-x filename
      - Remove execute rights from owner
    - \$ chmod g+rx filename
      - Gives the users of the group read and execute permissions
    - \$ chmod g-x filename
      - Remove execute rights from the users of the group
    - \$ chmod o+r filename
      - Gives read access to all (others)
    - \$ chmod o+x filename
      - Gives execute access to all (others)
    - \$ chmod 755 filename
      - Gives read,write,execute to owner and read and execute to group and others
-



- You should generally be careful about these files
  - SUID (Set UserID) and SGID (Set GroupID) has special file permissions attributes that enables a executable to be executed with the privileges of the owner or group
  - You should keep a record of the number of SUID and SGID files owned by user root and/or group root
  - SUID Executable
    - ```
$ ls -l /usr/bin/passwd
```

```
-r-s--x--x 1 root wheel 6052 Jun 6 2003 /usr/bin/passwd
```
  - The executable `'/usr/bin/passwd'` will run with the privileges of user root
  - This is necessary as it needs to update `'/etc/master.passwd'` which is readable and writable only by user root
  - SUID is needed for this command as users cannot run around system administrators to change their passwords
-



- **SGID Executable**

- `$ ls -l /usr/bin/write`  
`-rwxr-sr-x 1 root tty 8744 Aug 27 2001 /usr/bin/write`

- The executable '/usr/bin/write' will run with the privilege of group tty as it needs access to '/dev/ttyX' and '/dev/console' normally owned by group tty
  - Finding SUID and SGID files in the system
    - SUID
      - `$ /usr/bin/find / -type f -perm +4000`
    - SGID
      - `$ /usr/bin/find / -type f -perm +2000`
-



- It is a misconception that once you delete a file no one can possibly view them
  - However there are softwares available that can retrieve data from deleted file
  - This is possible because when we delete a file, it only removes the entry from inode table and the actual data is not erased
  - There are some tools available in the Internet that can ensure proper/complete file deletion such as:
    - Wipe
      - <http://wipe.sourceforge.net>
-



- FreeBSD now supports ACL or Access Control Lists
  - Previously the file permissions were modified on the basis of user, group and other
  - ACLs allow you to use more flexible/fine grained rules
  - Using ACLs, you can grant full access to user ram, read to user laxman, execute to group chemistry and full to group physics
  - ACL can be enabled by using “options UFS\_ACL” in kernel config file when compiling
  - ACL can be viewed by getfacl and modified by setfacl
-





- 
- FreeBSD offers special flags to be set in Files using the utility chflags
  - sappnd
    - Append Only flag
    - Can only be set by user root
    - Cannot be removed when kernel Secure level  $\geq 1$
  - schg
    - Immutable flag
    - Cannot be edited, moved or replaced
    - Can only be set by user root
    - Cannot be removed when kernel Secure Level  $\geq 1$
  - uappnd
    - Append only flag for users
    - Owner and root can modify
  - uchg
    - Immutable flag for user
    - Owner and root can modify
-



- When Possible, make separate partitions
    - /boot
    - /
    - /usr
    - /var
    - /tmp
    - /home
    - swap
  - The benefit of separate partitions is we can control what partitions are mounted with what options
  - We can use mount options such as noexec, nosuid, nodev, ro in '/etc/fstab' to protect the file system
-



## Sample /etc/fstab

---

/dev/ad0s1a	/	ufs	rw	1	1
/dev/ad0s1b	swap	swap	sw	0	0
/dev/ad0s1d	/usr	ufs	rw	2	2
/dev/ad0s1e	/var	ufs	noexec,nosuid,nodev	2	2
/dev/ad0s1f	/tmp	ufs	noexec,nosuid,nodev	2	2
/dev/ad0s1g	/home	ufs	noexec,nosuid,nodev	2	2

---



---

# System Logging

---



- You should use ntp for synchronizing time which is necessary for effective analysis of system logs
  - You can either use 'ntpd' or 'ntpdate' command periodically from cron to sync to a time source
  - Install primary and secondary ntp servers in your network that will sync to external ntp servers
  - Other servers and network devices can then sync to these local servers
  - To enable ntpdate on boot edit /etc/rc.conf
    - ntpdate\_enable=YES
    - ntpdate\_flag="-b 192.168.0.1"
-



- 
- Unix Systems uses syslogd for system logging
  - Some of the important files that needs to be monitored are
    - /var/log/messages
    - /var/log/auth.log
    - /var/log/security
  - You will find critical information about network login failures, failed su attempts, and various other useful informations
-



- You should also run a central network logging server
- Central logging server is important from security perspective because logs residing on a compromised machine can be tampered
- You can send logs to remote syslog server by putting the following in `/etc/syslog.conf`
  - `*.* @ remote-syslog-server`
- Individual services also generate their own log files
  - Apache
    - `/var/log/httpd/access.log`
    - `/var/log/httpd/secure.log`
  - Proftpd
    - `/var/log/proftpd/auth.log`
    - `/var/log/proftpd/access.log`
  - Squid
    - `/var/log/squid/access.log`



- As the number of servers grow, it can be quite cumbersome to audit logs of all the servers
  - There are numerous programs available on the Internet that can do the job
    - swatch
    - fwlogwatch
    - webalizer
-





---

# Backup

---



- 
- Backups are very crucial
  - In cases of problems that leads to loss of data, you can recover them from backups
  - Backups should be done regularly
  - Full and Reliable backup system should be maintained
  - Off-line Archives of softwares, upgrades, patches should be kept
  - Configuration files should be backed up regularly
  - Media, storage capacity, rotation methods should be considered
  - A full backup should be kept off-site for disaster recovery
  - Backups should be tested for reliability
-



- There are different kinds of backups you can perform in Unix Systems
    - Configuration files backup
    - tar
    - cpio
    - dump/restore
    - Rsync
    - Amanda (used tar and dump)
-



- When editing system configuration files, you should always make a backup before editing
  - For eg.
    - # cd /etc/
    - # cp syslog.conf syslog.conf.bck-2005-02-07



- 
- Tar is one of the most popular backup tool available in Unix Systems
  - You can create compressed tar archive using
    - # mkdir -p /backups/2005-02-07
    - # tar czvf usr.tar.gz /usr
    - # tar cjvf etc.tar.bz2 /etc
    - # tar cjvf home.tar.bz2 /home
  - You can restore from tar archive using
    - # tar xzvf home.tar.gz
-



- 
- cpio ( copy in copy out) is another popular backup tool that creates archives of your files and directories
  - Create cpio archive
    - # find / | cpio -o > full.cpio
  - Restore from cpio archive
    - # cd /
    - # cpio -idmv < full.cpio
-



- Dump can also be used as a backup tool
  - Creating a Dump
    - # /sbin/dump 0f /dev/tape /dev/hda2
  - Restoring from dump
    - # /sbin/restore rf /dev/tape
-



- 
- rsync is primarily used as a remote synchronizing tool
  - It can also be used to backup data to remote servers
  - The remote server needs to run rsync in daemon mode
  - Rsync server can also be run from inetd
  - Backup using rsync
    - `$ export RSYNC_PASSWORD=xxxxxxxxxxxx`
    - `$ rsync -a --delete /home user@rsync_server::backup/home`
  - Rsync Security
    - You can use SSH protocol to secure data in transit
    - `$ rsync -a --delete -e 'ssh' /etc user@rsync_server::backup/etc`
-





---

# Server Monitoring

---



- 
- Uptime command will show you details such as current time, time since last reboot, number online users and system load
    - `$ uptime`  
5:51pm up 186 days, 2:06, 6 users, load average: 0.06, 0.06, 0.05
    - The above output tells us that current time is 5:51pm
    - system is up for 186 days, 2 hr 6 min
    - 6 users are online on the system
    - Load average for 1 minute = 0.06
    - Load average for 5 minute = 0.06
    - Load average for 15 minute = 0.05
-



- ps is one of the most important tools for system resource monitoring
  - ps reports the running processes in the system
  - ps can be used with different options
  - 'ps -aux' will print all the processes with/without controlling terminal and will also display the user running the process
  - top is just like ps, however it updates its stat in real time
  - Additional stats on virtual memory can be seen using vmstat
-



- netstat/sockstat will show you the network socket information
  - For eg.
    - \$ netstat -an
      - will show you all sockets
    - \$ netstat -n -p tcp
      - will show you the currently open tcp sockets
    - \$ netstat -s
      - will show you various networking related informations
  - sockstat will you show you additional details such as command opening the port and it's PID
    - \$ sockstat -l -4
      - Show all IPV4 Listening sockets
-



- Process accounting is security system whereby you can keep track of allocation of system resources to users and processes
  - It also provides command auditing features
  - You can enable process accounting by adding the following to “/etc/rc.conf”
    - accounting\_enable=YES
  - To see the resource utilization/allocations
    - % sa
  - lastcomm can also be used to see the history about command executed on the system or commands run by a particular user
    - % lastcomm ls
    - % lastcomm user
-



## Other Monitoring Software

---

- Nagios
  - Big Brother
  - Big Sister
  - MRTG
  - RRD
  - Ntop
  - Cacti
-



---

# File System Integrity

---



- A security administrator must ensure that the configuration files and binaries crucial to system security and operation are not tampered with
  - It is generally seen that crackers after breaking in a system, most often make changes to system configuration files and leave Trojans
  - Trojans are programs that mimic the real program however also conducts certain other tasks as assigned by the cracker
-





- There are various tools that can report changes to File System
  - AIDE
  - Osiris
  - Tripwire



- Advance Intrusion Detection Environment
    - <http://www.cs.tut.fi/~rammer/aide.html>
  - AIDE is a Host-based Intrusion detection system
  - AIDE is a free alternative to Tripwire
  - AIDE is used to detect changes to important system configuration files and binaries
  - It makes a unique cryptographic hash of the files and stores it in a database
  - On a regular basis it compares the current hash generated with the stored “known-good” hash to determine file changes
  - AIDE is a great way to detect disallowed changes to a system
  - The configuration file for AIDE is based on regular expressions, macros and rules for files and directories
-



- 
- Initialize Database
    - # cp /usr/local/etc/aide.conf.sample /var/db/aide/
    - # aide -i
  - Storing the Database
    - # cd /var/db/aide/databases
    - # mv aide.db.new aide.db
    - # cp aide.db Some\_other\_secure\_server
  - Scan the system for file changes
    - # aide -C
  - Add it to '/etc/crontab' for regular checks
    - 0 3 \* \* \* root /usr/bin/aide -u
      - Make sure you have alias for root, as AIDE reports via email to root from crontab
-



- If changes are found it will create a new database `'/var/db/aide/databaes/aide.db.new'`
  - It will also email to root if run from cron
  - If changes are expected you can commit by replacing the old database with the new one
-



---

# Network Scanning

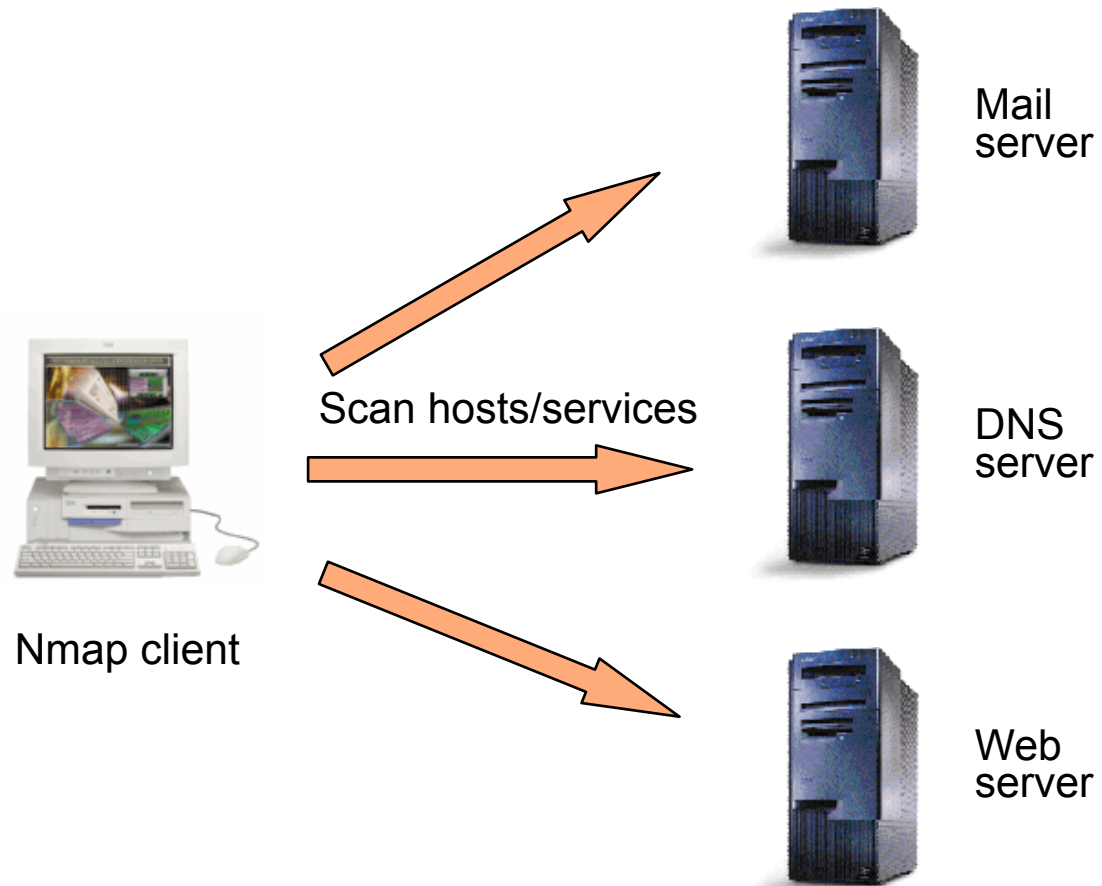
---



- Network Mapper
  - Powerful utility for network exploration and security auditing
  - Can scan a large network rapidly
  - Can determine hosts information available in the network
  - Can list ports available in hosts
  - Can Reports Operating System versions
  - Can report existence of packet filters/Firewalls
  - Runs on most Operating systems
  - Both Console and Graphical versions available
  - Free Software
-



# Nmap





- Target Selection
  - Specify targets on the command line or in a filename with the '-i' option
    - \$ nmap 192.168.0.1
    - \$ nmap 192.168.0.0/24
    - \$ nmap www.nosuchdomain.com
    - \$ nmap -i iplist.txt





- Ping Scan
    - Sends ICMP echo request
    - Also sends TCP ACK to port 80 and expects RST
    - Third option is to sent TCP SYN to port 80 and expect RST or SYN/ACK
    - Pinging is done by default
    - Ping can be disabled by option '-P0'
-



- The vanilla scan is a tcp connect() scan '-sT'
  - SYN scan ( -sS) also called “half-open” scans sends a SYN packet and looks for a SYN/ACK (open) or RST (closed).
  - Syn scan tears down the connection before sending the ACK that would normally complete the TCP 3way Handshake
  - FIN (-sF), XMAS (-sX) and NULL (-sN) scans sends following tcp flags to probe networks
    - FIN FIN
    - XMAS FIN,URG,PUSH
    - NULL NULL Flags
  - UDP Scanning (-sU) sends 0 sized udp packets to scan udp ports on target machine
-



- tcpdump is a traditional utility available in Unix to sniff packets
  - Based on libpcap
  - Can sniff packets destined for local hosts
  - Can sniff packets for the whole network in promiscuous mode
  - Various options are available for sniffing packets
-



- 
- Source network 192.168.0.0/24
    - # tcpdump -i em0 src net 192.168.0.0/24
  - Without Hostname Translation
    - # tcpdump -i em0 -n src net 192.168.0.0/24
  - Without Port name translation
    - # tcpdump -i em0 -nn src net 192.168.0.0/24
  - Src 192.168.0.0/24 and Destination 192.168.0.1
    - # tcpdump -i em0 src net 192.168.0.0/24 and dst 192.168.0.1
-



- Protocol Specific
  - # tcpdump -i em0 tcp
  - # tcpdump -i em0 ! arp
- Print packets in Hex and ASCII
  - # tcpdump -X -i em0



- 
- ntop is a network traffic monitoring program
  - It is based on libpcap
  - It can also act as a netflow collector
  - Some of the features of ntop are
    - Traffic Statistics
    - Sort network traffic according to protocol,source/destination address,ports etc
    - Identify Host OS on the network
    - Report IP traffic Usage
    - Monitor suspicious traffic
    - Web page to monitor and administer ntop
-



Welcome to ntop!

# ntop

About Summary IP Media Admin Utils

- Traffic
- Hosts**
- Network Load
- ASN Info
- VLAN Info
- Network Flows

## Host Information

	Domain	IP Address	MAC Address	Other Name(s)	Bandwidth	Nw Board Vendor	Hops	Dis
host254		83.149.146.254						
host078-144		83.149.144.78						
host005-160		83.149.160.5						
host019-154		83.149.154.19						
host017-148		83.149.148.17						
host081-144		83.149.144.81						
host016-148		83.149.148.16						
host067-144		83.149.144.67						
host153-147		83.149.147.153						
host095-144		83.149.144.95						
host019-146		83.149.146.19						
host014-148		83.149.148.14						
freebsd.computerhouseprato.com		83.149.154.10						
freebsd.giovannelli.com		83.149.149.149						
host012-144		83.149.144.12						
host023-146		83.149.146.23						



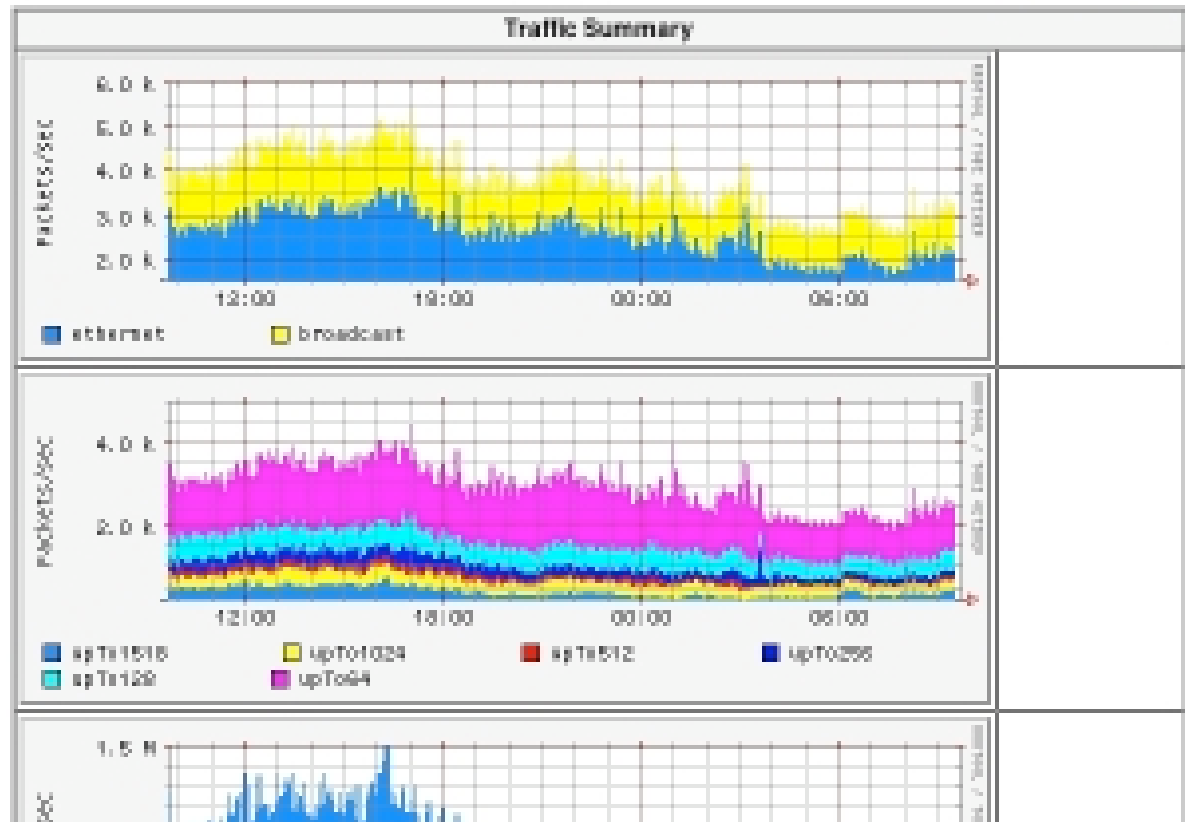
Welcome to ntop!

**ntop**

About Summary IP Media Admin Utils

### Info about interface Consiag

View: [year](#) | [month](#) | [week](#)







---

# Cryptography

---



- Cryptography comes from a Greek word for “Secret writing”
  - A cipher is a character–for–character or bit–for–bit transformation without regard to the linguistic structure of the message
  - A code replaces one word with another word or symbol, not in used nowadays
  - In cryptography the message to be encrypted, known as plaintext are transformed by a function that is parameterized by a key. The output of the encryption process is know as the ciphertext
-



- Traditional Ciphers
    - Substitution ciphers
      - each letter of group of letters is replaced by another letter of group of letters
    - Transposition Ciphers
      - Reorder the the letters
    - One time pad
      - Here the plaintext is converted into a bit string which is Xor(exclusive OR) with another one time bit string
-

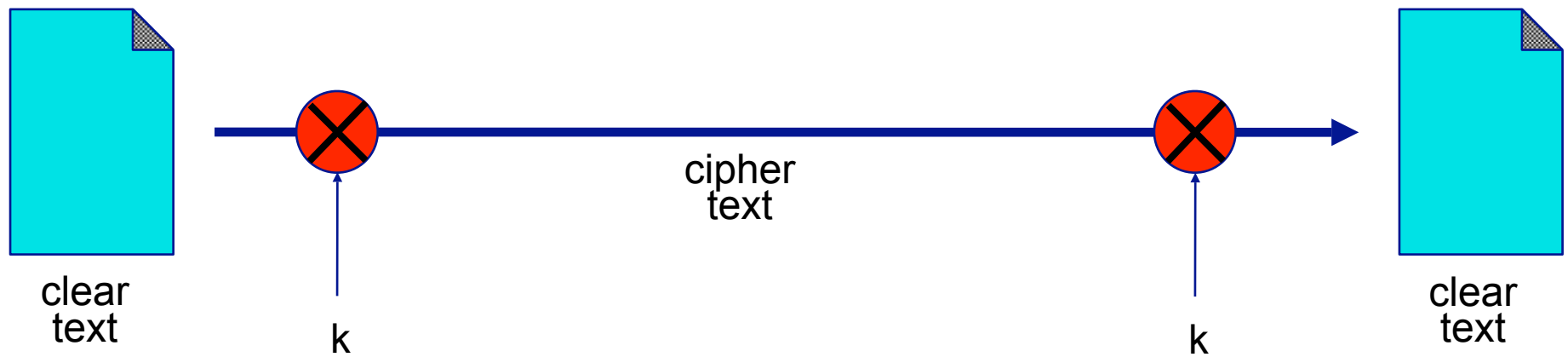


- Same key is used for encryption and decryption process
  - The strength of the Encryption is related to the key length of the cipher
  - Examples
    - DES (Data Encryption Standard)
    - Triple DES
    - AES (Advance Encryption Standard)
    - Blowfish
    - IDEA (patented not free )
-



# Symmetric Key Cryptography

---





- Two keys are used for encryption and Decryption process
    - Private Key
    - Public Key
  - The public key and private key are mathematically related (generated as pair)
  - We can generate public key from private key but not private key from public key
  - Public key cryptography can be used to encrypt or digitally sign messages
  - Example : RSA (Rivest Shamit Adleman)
-

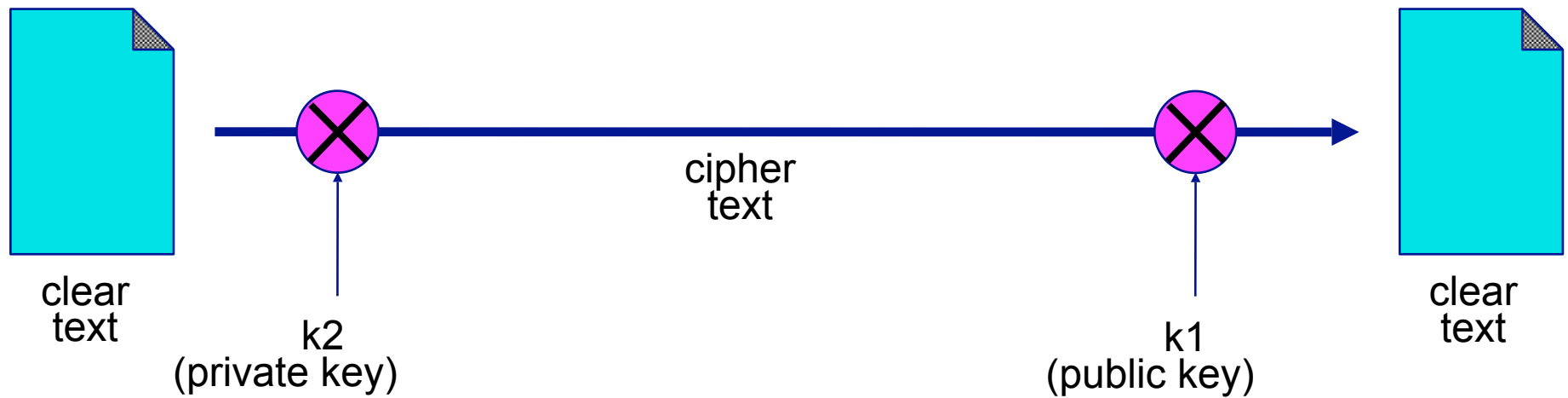


- The security of Private key is of paramount importance. Should be kept private as the name implies
  - You must not loose it
  - Prefer to have a backup on encrypted File systems at multiple places
  - Protect the private key with a pass phrase
  - Public key is meant to be distributed to others so that you can have secure communications with them
-



# Public Key Cryptography

---







- 
- Secure Socket Layer
  - Encryption method developed by Netscape to provide security over the Internet
  - Supports different encryption protocols and provides client-server authentication
  - SSL operates in Transport Layer
  - Create a secure Encrypted tunnel for data
  - Commonly used in secure sites
  - Uses https instead of http
-



- GNU Privacy Guard
  - Uses Public Key Cryptography
  - Used for Secure email communications
  - Can be used to Digitally sign emails for authenticity
  - Can be used to Encrypt emails for security
  - You can also encrypt files using GnuPG (GNU Privacy Guard)
    - \$ `gpg -e --default-recipient-self filename.txt`
      - A file named 'filename.txt.gpg' will be created
    - \$ `gpg --decrypt filename.txt.gpg`
      - gpg will decrypt the file and send it to standard output
  - You can also use “vim” Editor to edit and save encrypted gpg files
-



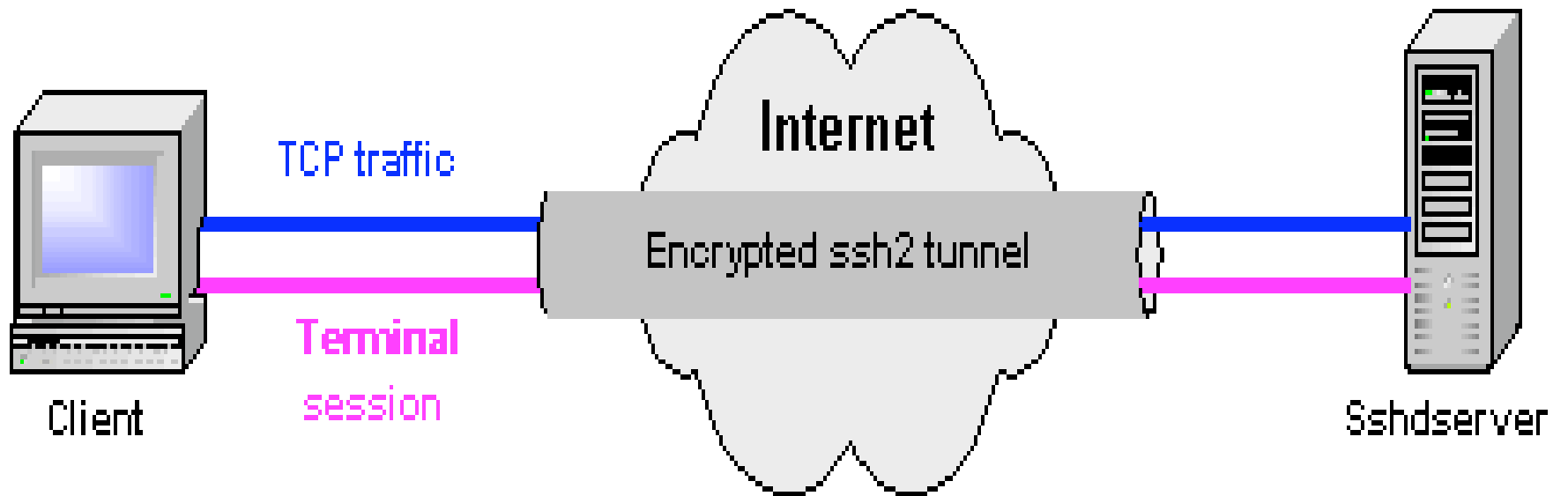
- Sign Email Messages
    - Create a Hash of the message
    - If the message is altered then the Hash differs and the receiver can notice the changes
    - Private Key is used to Sign the Messages
    - The person receiving the email should have your Public key to verify the integrity of the email
  - Encrypt Email Messages
    - You encrypt the email messages using the Public key of the person you are sending the email
    - The person receiving the email should have his private key to open the encrypted email
  - Encrypt Files
    - You can use gpg to encrypt text files
    - You have to use your own public key to encrypt the file
    - You have to use your private key to decrypt the file
-

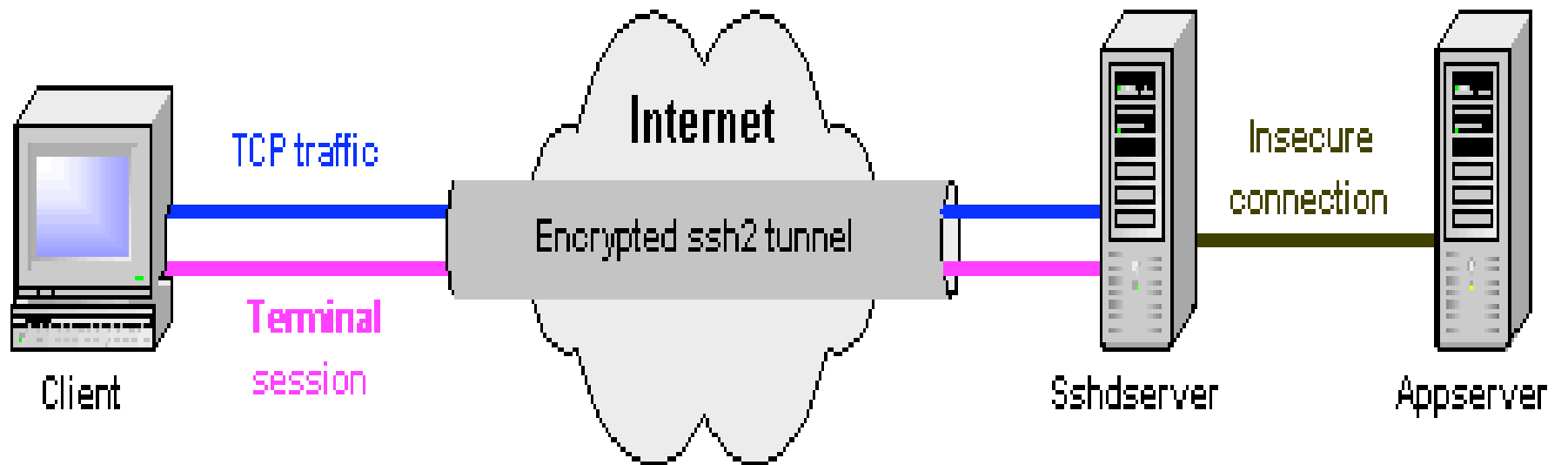


- Secure Shell
  - SSH is a suite of protocols used as a secure replacement for rlogin, rsh and rcp
  - It can use various ciphers to encrypt data between two hosts
  - It supports Public key based authentication
  - It allows secure login and secure copy of files
  - It can provide data compression
  - It can provide secure X11 communications
-



- 
- It can provide secure forwarding of arbitrary tcp connections
  - It prevents eavesdropping , session hijacking , dns spoofing etc
  - Transparent to the user, similar to a telnet session
  - SSH is strongly recommended against telnet, ftp, r-services
  - SSH is installed by default in FreeBSD
  - Disable SSH version 1 as it contains various exploits
  - Disable direct root login from ssh
-







---

# Firewall

---





- People often think that a firewall provides the ultimate security, but they are wrong
  - In most cases a mis configured firewall gives less or no security
  - A firewall is a piece of software and should be treated the same way as any other softwares, because it is just as likely to contain bugs
  - Firewall are used to enforce Access Control Policy between two networks such as the Internet and the Local Network
-



- Firewall are used to filter out unauthorized / Malicious traffic in or out of the network
  - Firewall are used to protect the Internal Network from External network
  - Firewall are used to protect services running in the network
  - Firewall can not protect traffic that does not go through it
-



- Basically there are two Modes of Firewall
  - Packet Filtering Firewall
  - Application Gateway Firewall



- All network traffic are sent in the form of packets
  - Large amount of data is split up into small packets for easy handling/routing and then reassembled when it arrives at destination
  - In packet header of every packets contains information on how and where it should be delivered
  - This very information is used by Packet Filtering Firewalls to implement Access control
-



- Packet Filtering is done on the basis of
    - Source/Destination IP address
    - Source/Destination Port
    - Protocol
    - Flags within a specified protocol
    - Combination of Above
  - Examples
    - ipfw
    - pf
    - ipf
-



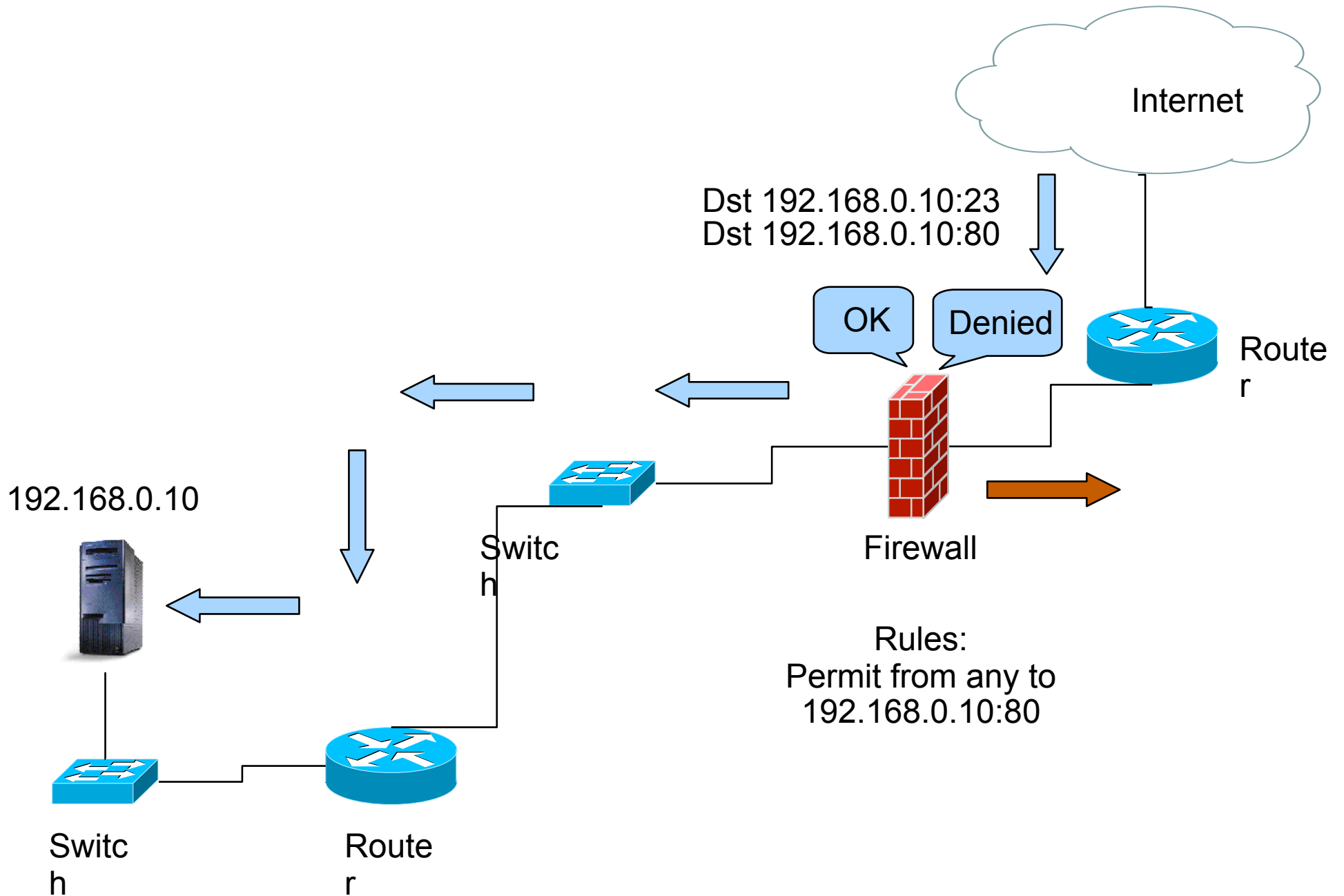
- Stateful Packet Filtering
    - Examines the state and the context of the packets ( connection tracking)
    - An entry is made for each outgoing packet (request) and only replies (response) to these packets are allowed, and vice versa
    - Don't need to explicitly allow higher ports (1024:) as the stateful firewall will allow all required packets
-



- Advantages
    - Simple and Easy to Implement
    - Can give warnings of a possible attack
    - Good for stopping SYN Attacks and PING flood
  - Disadvantages
    - Address information can be spoofed
    - Data within allowed packets can contain malicious codes that can exploit bugs
    - Usually a single point of failure
-



# Packet Filtering Firewall







- The application gateway is a proxy for applications
  - Exchange data with remote systems on behalf of the clients
  - The Actual client is not visible to the Internet
  - Protects the clients from the outside network by proxying request for them
-



- Filtering is done on the basis of
    - Source/Destination IP address
    - Source/Destination Port
    - Packet Content
    - File Type and Extensions
    - Time of Day
    - User Authentication
  - Examples
    - squid
-



- Advantages
    - Can cache files for increased network performance
    - Detailed logging of all connections
    - Scalability, cache sharing , redundancy
    - No direct access to clients from Outside networks, not reachable from the Internet
    - Can alter packet contents on the fly
    - Considered to be the most secure since these services don't need to be run as root
  - Disadvantages
    - Configuration and Implementation is complex
-



- Allow only needed services
  - The default policy should be to Reject
  - Firewall is an addition to the level of security, take other measures as well
  - Regularly review/update firewall rules
  - Regularly audit firewall logs for potential attacks
  - Apart from the Main firewall, install firewall on individual hosts too
  - Make sure traffic can't bypass the Firewall
  - A central firewall should be running only the firewall software and sshd, and nothing else
-



---

ipfw

---



- FreeBSD has three different Firewall Packages
    - ipfw                   - IPFWALL
    - ipfilter   - IPFILTER
    - pf                   - OpenBSD Packet Filter
  - ipfw can be enabled at boot time by adding the following to rc.conf
    - firewall\_enable="YES"
    - firewall\_script="/etc/rc.firewall"
    - firewall\_logging="YES"
  - The amount of logs can be controlled by adding the following to sysctl.conf
    - net.inet.ip.fw.verbose\_limit=200
  - ipfw can also be used for bandwidth limiting using Dummynet
-



- The following options can be used in the kernel config to build ipfw directly in the kernel
    - options IPFIREWALL
    - options IPFIREWALL\_VERBOSE
    - options IPFIREWALL\_VERBOSE\_LIMIT=200
    - options IPFIREWALL\_DEFAULT\_TO\_ACCEPT
      - Change the default rule to accept
    - options IPDIVERT
      - To enable NAT functionality using divert sockets
-



- 
- List Firewall rules
    - ipfw list
    - ipfw l
  - List Firewall rules with packet accounting information
    - ipfw -a list
  - List Firewall rules with timestamps of last matches
    - ipfw -t list
  - List the dynamic rules
    - ipfw -d list
  - List the expired dynamic rules
    - ipfw -d -e list
  - Clear the packet accounting counters
    - ipfw zero
    - ipfw zero RULENUM
-





- Adding a rule
    - `ipfw add deny ip from any to any`
  - Rule number
    - Each rule is associated with a rule number in range 1 - 65535. The last rule number is reserved for the default rule. If a rule number is not specified , it is automatically assigned by the kernel
    - `ipfw add 2000 allow ip from any to me`
  - Deleting a rule
    - `ipfw delete 100`
-



- The action of the rule define the fate of the packet
    - allow | accept | pass | permit
      - This action will accept the packet
    - check-state
      - This action will the check the packet against the dynamic rules table and if there is a match then executes the action associated with the rule which generated the dynamic rule
    - deny | drop
      - This action will discard the packet
  - If logging is desired for the packet matching the rule, “log” keyword can be added after the action. It may also be followed by “logamount” keyword to limit the logs
    - ipfw add deny log logamount 200 ip from any to any
-



- You can specify the protocols in an ipfw rule
  - Any protocols from /etc/protocols can be used
    - ip All IP Protocols
    - tcp TCP Packets
    - udp UDP Packets
    - icmp ICMP Packets
    - esp ESP Packet
  - ipfw add allow udp from me to any
    - Allow all udp packets from this host
  - ipfw add deny tcp from any to any
    - Deny all tcp packets from any to any
  - ipfw add deny icmp from any to me icmptypes 8
    - Disable icmp echo-request packets
  - You can also specify port numbers in the rule
    - ipfw add allow tcp from any to me 80
      - Allow tcp packets from any to this machine on port 80
-



- You can match on the basis of packet direction
    - ipfw add allow tcp from me to any out
      - Match outgoing tcp packets from this host
    - ipfw add allow tcp from any to me 80 in
      - Match incoming tcp packets to port 80
  - You can also use the via keyword to check the interface of the packet
    - ipfw add allow ip from any to any via lo0
      - Allow all packets coming in or going out from loopback interface
-



- You can match a start packet of a tcp session using the setup keyword. It will match the packets having the SYN only flag
    - ipfw add allow tcp from me to any 80 setup
  - You can match a packet which has ACK or RST flag set using the established keyword
    - ipfw add allow tcp from any to any established
  - keep-state keyword can be used to create stateful rules. keep-state will create dynamic rules which will be consulted during the processing of reply packets via check-state or the first keep-state rules
    - ipfw add allow tcp from me to any setup keep-state
    - ipfw add allow udp from me to any keep-state
-



- 
- ipfw also provides ratelimiting
  - limit keyword can be used to limit the number of packets matching based on :
    - source address
    - source port
    - destination address
    - destination port
  - ipfw add allow tcp from any to me 80 limit src-addr 10
    - Only allow 10 connection per host on port 80
-



---

# Intrusion Detection

---



- Intrusion Detection System (IDS) analyzes IP packets looking for known patterns in real time
  - IDS can give valuable information on unauthorized access to your network
  - IDS can be Host based or Network Based
-





- Watches for packets coming into a single host
  - The host based IDS doesn't listen on interfaces in promiscuous mode
  - Programs that parse the system log files for security related informations can also be termed as Host based IDS
  - HIDS reports incidents that seems suspicious and alert the administrators
  - Examples
    - Portsentry
    - fwlogwatch
    - chkrootkit
    - swatch
    - AIDE
    - Osiris
    - Tripwire
-



- Tool written to detect known Trojans and root kits installed in the system
  - Check if the Ethernet interface is promiscuous mode
  - Check if the lastlog/wtmp files are modified
  - Check for logs created by sniffer programs
  - <http://www.chkrootkit.org>
-



- Analyzes all IP packets coming into the network
  - NIDS are generally deployed with port mirroring facility provided by most managed switches
  - Such switches are configured to copy all data on port/ports and send it the port where NIDS is connected
  - NIDS generally listen on interface in promiscuous mode and analyzes all data it receives
  - Can also be used in Bridging Mode
  - Snort is one of the most popular Network based IDS, and is also open source
-



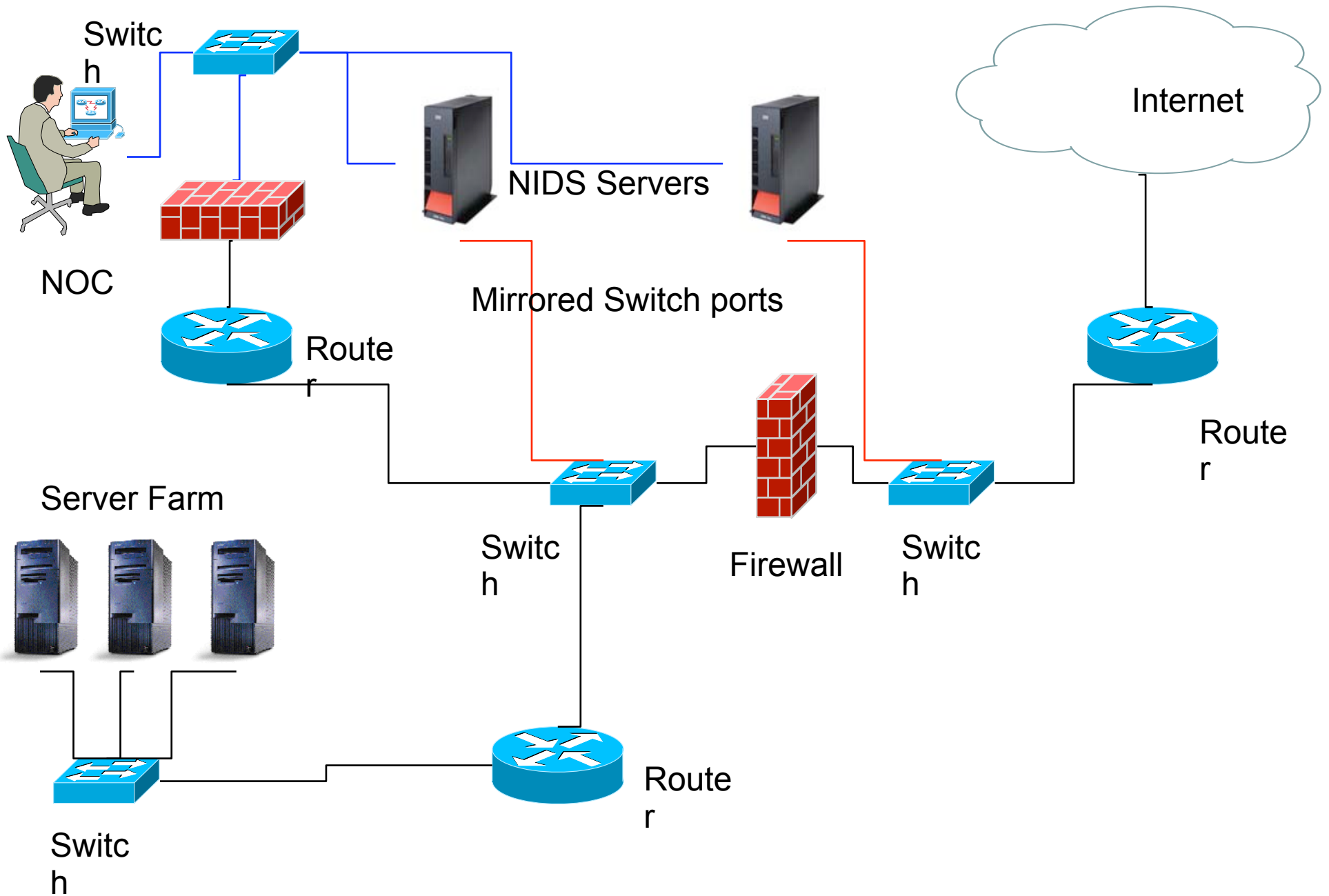
- 
- Network Intrusion Detection System (NIDS)
  - Inspects/Sniffs all network traffic for abnormal contents
  - Has built in signature base and anomaly detection
  - String search for Known signatures with logging and reset features
  - Rules based logging to perform content pattern matching detect a wide variety of attacks and probes
  - Detects buffer overflows, stealth port scans, CGI attacks, SMB probes and much more
  - Has Real-Time Alerting capability
    - syslog
    - SMB( Winpopup)
    - alert file
-



- Place snort before a firewall for maximum detection
  - Use a Mirrored port if available
  - Can also be operated in bridging mode
-



# Snort Placement





---

# Part 2

# Router and Routing Security

---