Vicky Shrestha

WorldLink Communications

vickysh@wlink.com.np

- Introduction
- User Administration
- File System
- Physical and Console Security
- Securing Services
- Securing FreeBSD
- System Logging
- Backup
- Server Resource Monitoring
- File System Integrity
- Network Scanning
- Cryptography
- Firewall
- Intrusion Detection
- Spam

Introduction

"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

- 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

User Administration

- Each user in Unix has a unique User ID
- Each user is also given a separate Home Directory and a default shell
- User has one primary group and may belong to one or more secondary groups
- A valid user name is required to log in to a system
- Different users having the same privilege and access level requirements can be assigned to a group and group permissions can be assigned

- User root is the super user who has access to virtually everything within a system
- Use of root user should be avoided until absolutely needed
- Most administrative jobs can be done using a normal user account
- One should login as a normal user and use 'su' (Substitute User) only when root privileges are required
- We can also use sudo to run scripts or programs requiring root access
- If an attacker gains root access, then the only way to trust your machine is to have a complete reinstall
- You should disable direct login as user root in both “/etc/ttys” and sshd config

- User Details are stored in /etc/passwd
- Passwords are stored in encrypted format using one-way encryption
- Limitations/Weakness of '/etc/passwd'
  - '/etc/passwd' file needs to be readable by all
  - The local users can access the encrypted password if present in /etc/passwd file
  - They can use password crackers to crack the passwords and try to gain unauthorized access
- Many version of Unix now store the encrypted password in a separate file, which is only readable and writable by user root
- In FreeBSD it is stored as /etc/master.passwd
- In addition to Legacy password files, FreeBSD also stores a indexed password database in /etc/pwd.db and /etc/spwd.db

- You can control the default password policy by editing '/etc/login.conf' file
  - password_format=md5
    - Set the password hashing format to md5
  - passwordtime=90d
    - password is valid for 90 days
  - warnpassword=7d
    - warn the user about about his password expiry before 7 days
  - warnexpire=14d
    - warn the user about his account expiry before 7 days
  - minpasswordlen=8
    - Minimum number of characters required for a password
- New version of FreeBSD support Password policy using pam module
  - /etc/pam.d/passwd
  - man pam_passwdqc

- Chpass
  - chpass is a utility to change the password file and password policies

    For eg:
    - Setting the expiry date of a user
      - # chpass -e '06 19 2005' guest
    - Assigning a new Shell
      - # chpass -s /nonexistent guest
    - You can also change other parameters such as
      - UID/GID
      - Login name
      - class
      - GECOS
      - Password change time

- You can lock a User account so that he doesn't get any access to the server by using pw
  - pw lock guest
- You can unlock the account by
  - pw unlock guest
- You can also use chpass and add “!” or “*LOCKED*” before his Encrypted password

- You can check the logins of users by using the following commands
  - w
    - will show you the currently logged in users
  - last username
    - will show you the record of logins of username
- File related to these commands is
  - /var/log/wtmp

- All users in the system don't need shell access
- FTP/Mail users can be safely given a '/bin/false' shell
- sftp only users can be given a shell called '/bin/sftponly' (can be downloaded from Internet)
- Users should not be assigned a login shell unless absolutely required
- Restrict shell access from selected networks only
- Use of tcpwrappers and/or firewalls to restrict services such as ssh

- su (Substitute User)
- Instead of using root for your day to day job, you should consider using normal user account and use su only when root privileges are required
- su access can be further protected using PAM (Pluggable Authentication Modules)
- In FreeBSD, by default only users in the wheel group are allowed to use su
- su access is controlled from /etc/pam.d/su
- su access can be protected in other versions of Unix also

- There maybe more than one person handling the server and as a result may need superuser privileges to do their jobs
- These users maybe backup operators, mail administrators, log analyzers
- Generally one should not give root password to each and every users
- Some jobs can be better done by using sudo package

- sudo package can be configured such that users can have superuser privileges for preassigned commands only
- You can customize the sudo package by using 'visudo'
- visudo edits '/usr/local/etc/sudoers' where we specify which users can execute which commands by using sudo
- For eg.
  - $ sudo tail -f /var/log/messages


```
#Sudoer file

#User alias specification
User_Alias   ADMIN =  vicky

# Cmnd alias specification
Cmnd_Alias   USERADD = /usr/sbin/adduser

# User privilege specification
root ALL = (ALL) ALL
ADMIN   server = USERADD
```

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 3rd and 4th 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 specials 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 >= 1
- schg
  - Immutable flag
  - Cannot be edited , moved or replaced
  - Can only be set by user root
  - Cannot be removed when kernel Secure Level >=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
```

Physical and Console Security

- The system should be installed in a secured server room with limited access to authorized users only
- Proper monitoring systems such as CCTV should be installed
- Biometric authentication system is also desirable where needed
- Proper alarm system and preventive measures should be in place to prevent fire, break-ins ,theft etc

- Use of BIOS passwords - Not the default ones
- Disable boot from external floppy/CD drive/SCSI/USB drives etc
- Protection of CTRL+ALT+DEL key sequence
- Notifications of Case Removal
- Monitoring system such as nagios, to alert the System Administrators as soon as system is down/Unreachable
- Use of console lock programs such as vlock or xscreensaver
- Deny root access to Console “/etc/ttys”
- Enable password prompt in single mode “/etc/ttys”

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 know 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”

- You can configure the startup services from "/etc/rc.conf"
  - sendmail_enable=NO
    - Sendmail will only listen on the loopback address
  - sshd_enable=YES
    - Enable sshd service on startup
  - nfs_server_enable=NO
    - Disable NFS server
  - nfs_client_enable=NO
    - Disable NFS client
  - rpcbind_enable=NO
    - Disable Portmap services
  - syslogd_enable=YES
    - Enable syslog
  - inetd_enable=NO
    - Disable inetd daemon

- 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

- 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-ftpd,proftpd and bind have features for chrooting
- Services not having such support can also be run in a chroot enviroment 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
- Jail also offers virtual system image

- 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 updates

- 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 -F to download the latest audit database
- 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

- *default host=cvsup11.FreeBSD.org
  - Default Host to connect
- * default prefix=/usr
  - Default Prefix to use for updated files
- *default base=/var/db
  - Directory for cvsup bookkeeping
- *default release=cvs tag=RELENG_5_4
  - cvs means that the server should get its information from the main FreeBSD CVS Repository
  - tag
    - ./HEAD Current/ Latest
    - RELENG_5 Development branch of FreeBSD 5.x / FreeBSD 5-Stable
    - RELENG_5_4 Release for FreeBSD 5.4 For Security and other critical Fixes
    - RELENG_4 Development branch for FreeBSD 4.x FreeBSD 4-Stable

- *default delete
  - Allow cvsup to delete local files
- *default use-rel-suffix
  - cvsup maintains a bookkeeping based on the suffix constructed from release and tag
- *default compress
  - Enable gzip-style compression when doing cvsup
- src-all tag=RELENG_5_4
  - Sync all the sources from 5.4 Release branch
- ports-all tag=.
  - Sync all the ports from current
  - Get the latest ports collection

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

- 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 extenstions
- 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

/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

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 expect 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

System Logging

- You should use ntp for synchronizing time which is necessary for effective analysis of system logs
- You can either use 'ntpd' of '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

- 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_PASSWD=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 Resource 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

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
  - Tripwire
  - AIDE
  - yafic

- Tripwire is a File System Integrity checker that can report changes in files
- It does this by comparing major attributes of a file such as binary signature, size, md5sum etc against a previously built database
- Changes to the filesystem are reported
- You should edit the following files to suit your needs
  - twcfg.txt
    - Tripwire configurations
  - twpol.txt
    - Tripwire policy file
- You can then build a signed version of the policy file using twadmin or twinstall.sh shell script

- You can create a database using
  - # tripwire --init
- You can check for file alteration using
  - # tripwire --check
- You also have to make sure that you run the tripwire-check command on a daily basis via crontab
- You can generate reports using
  - # twprint -m p -r reportfile.twr
- You can update the tripwire database using
  - # tripwire --update -r reportfile.twr

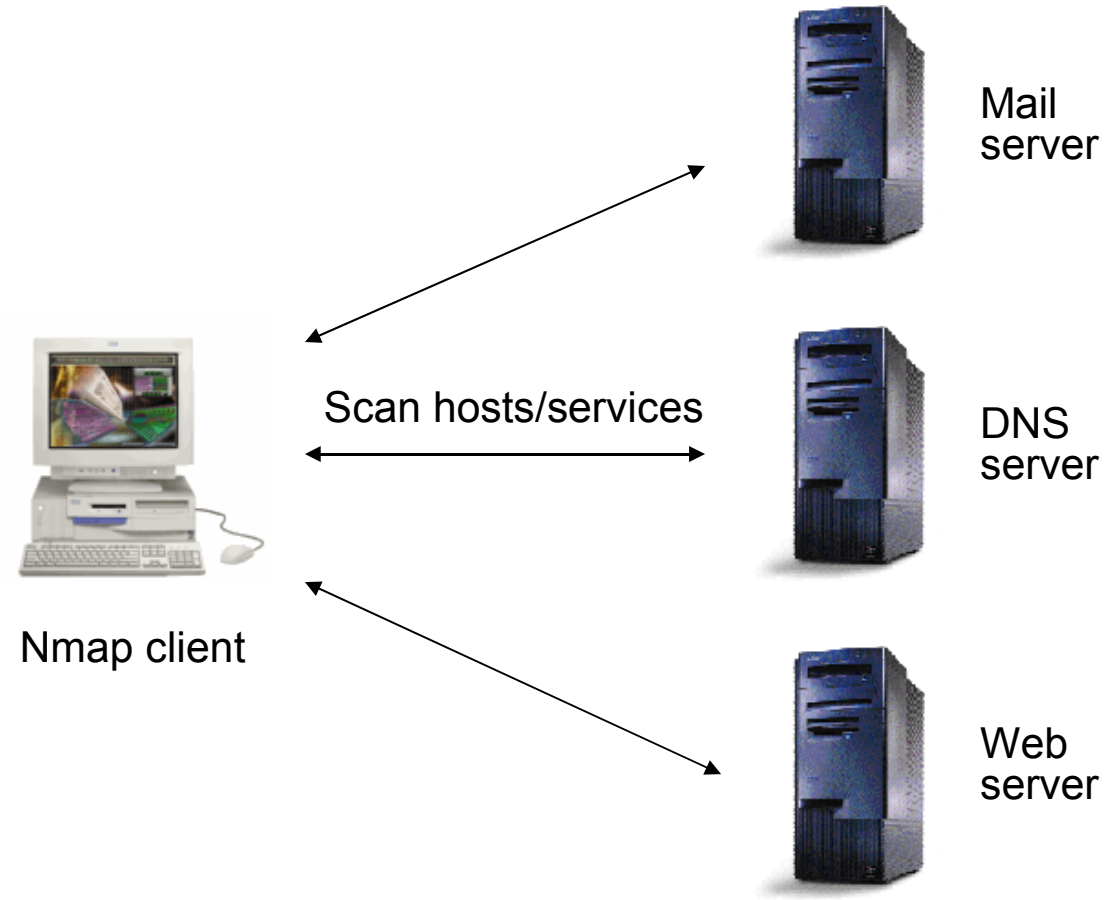
- Advance Intrusion Detection Environment
  - http://www.cs.tut.fi/~rammer/aide.html
- AIDE is a Host-based Intrusion detection system
- AIDE is a free alternative to Tripwire
- AIDE is used to detect changes to important system configuration files and binaries
- It makes a unique cryptographic hash 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 * * * /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


Mail
server
Scan hosts/services
DNS
server
Nmap client
Web
server

- 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

- Compares two nmap scans
- Allows monitoring your network for interesting changes in port states and visible hosts
- Eliminates the need to manually examine and compare nmap results
- Useful to network administrators to monitor large networks
- Ndiff is generally run from crontab
- Supports HTML output
- Requires Perl
- Supports various options for controlling output format

- Usage
  - $ nmap -m first_scan.nm 10.0.0/24
  - $ nmap -m second_scan.nm 10.0.0/24
  - $ ndiff -baseline first_scan.nm -observed second_scan.nm
- We are here comparing nmap outputs of first and second scan
- We designate first_scan.nm as the baseline for comparison

- 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

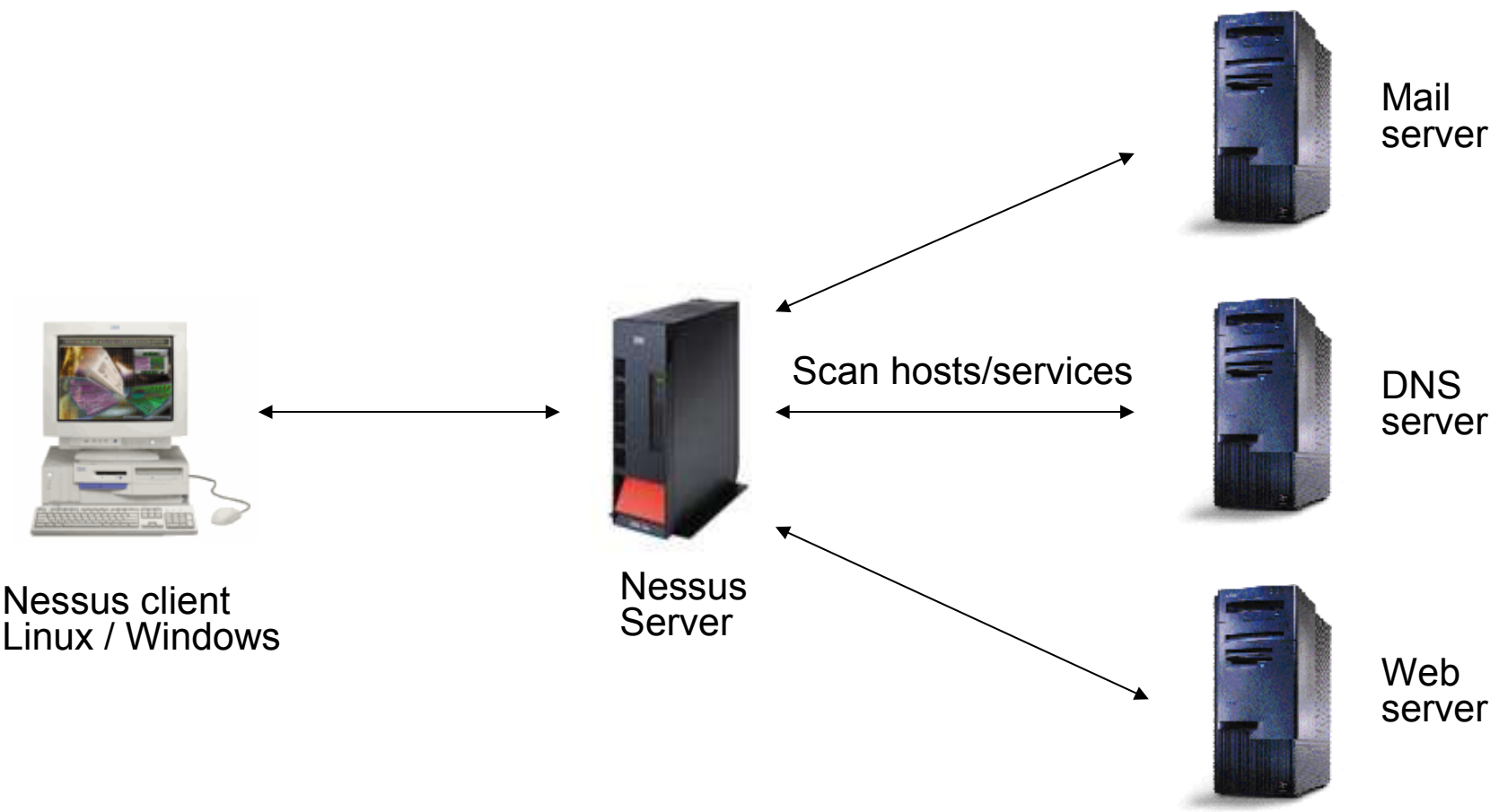
- Source network 192.168.0.0/24
  - # tcpdump -i em0 src net 192.168.0.0/24
- Without Hostname Translation
  - # tcpdump -i em0 -n src net 192.168.0.0/24
- Without Port name translation
  - # tcpdump -i em0 -nn src net 192.168.0.0/24
- Src 192.168.0.0/24 and Destination 192.168.0.1
  - # tcpdump -i em0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

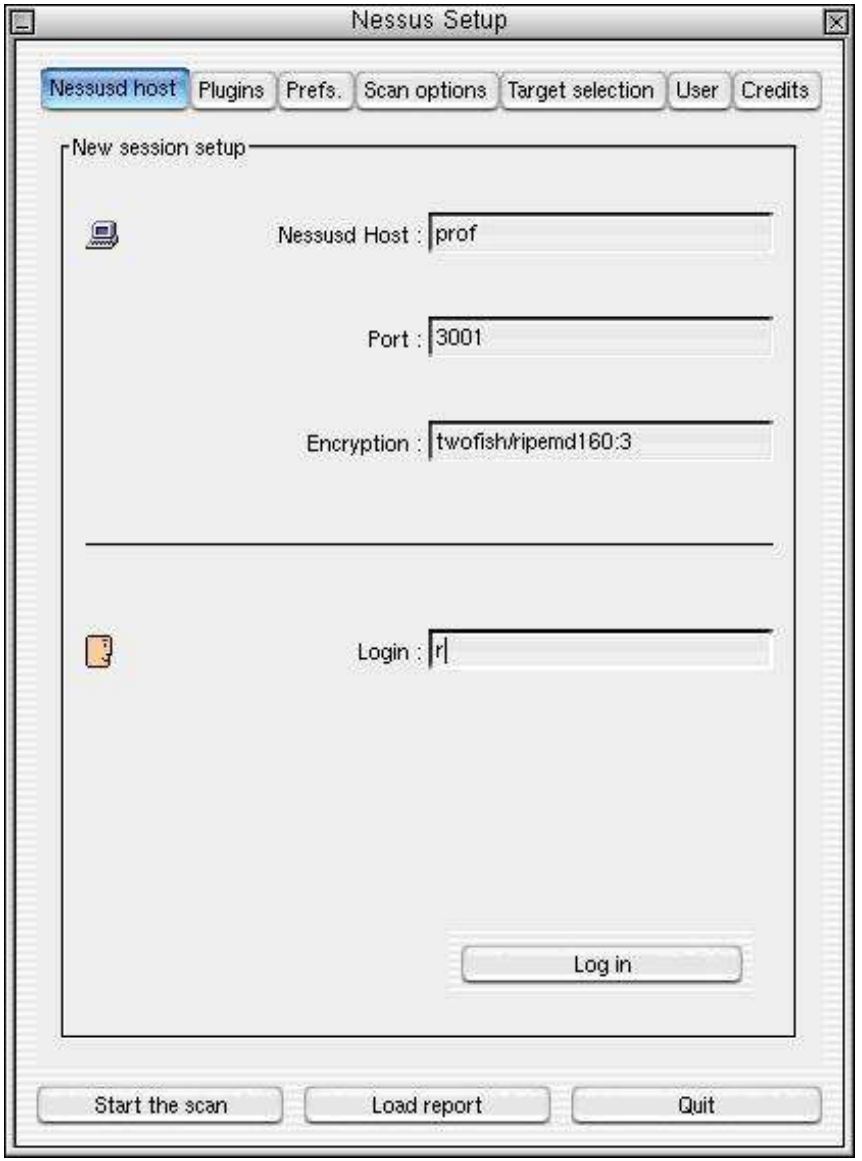
- Protocol Specific
  - # tcpdump -i em0 tcp
  - # tcpdump -i em0 ! arp
- Print packets in Hex and ASCII
  - # tcpdump -X -i em0

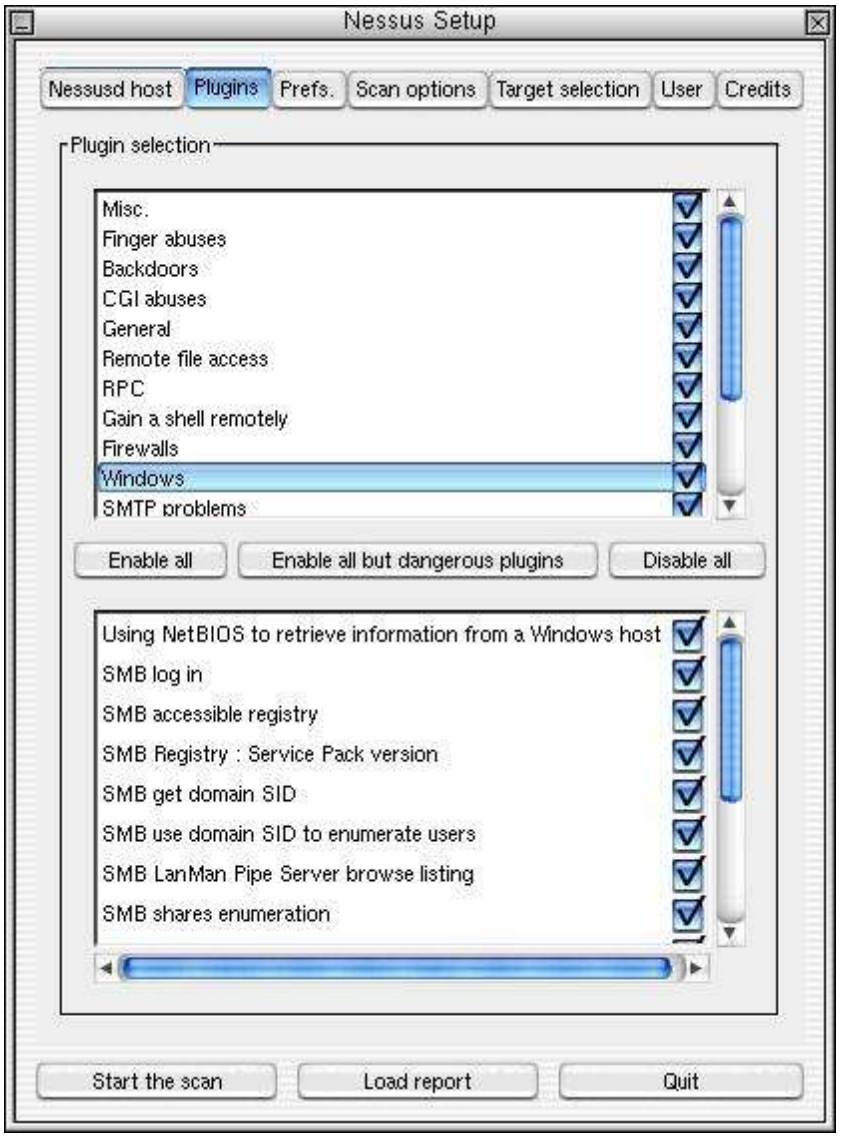
- A security Scanner
- Software to remotely audit the network
- Determines what services are running on which hosts and the vulnerabilities associated with them
- Nessus can also detect services running on non standard ports
- Very fast and reliable
- Has Modular architecture
- Plug-ins can be made according to need

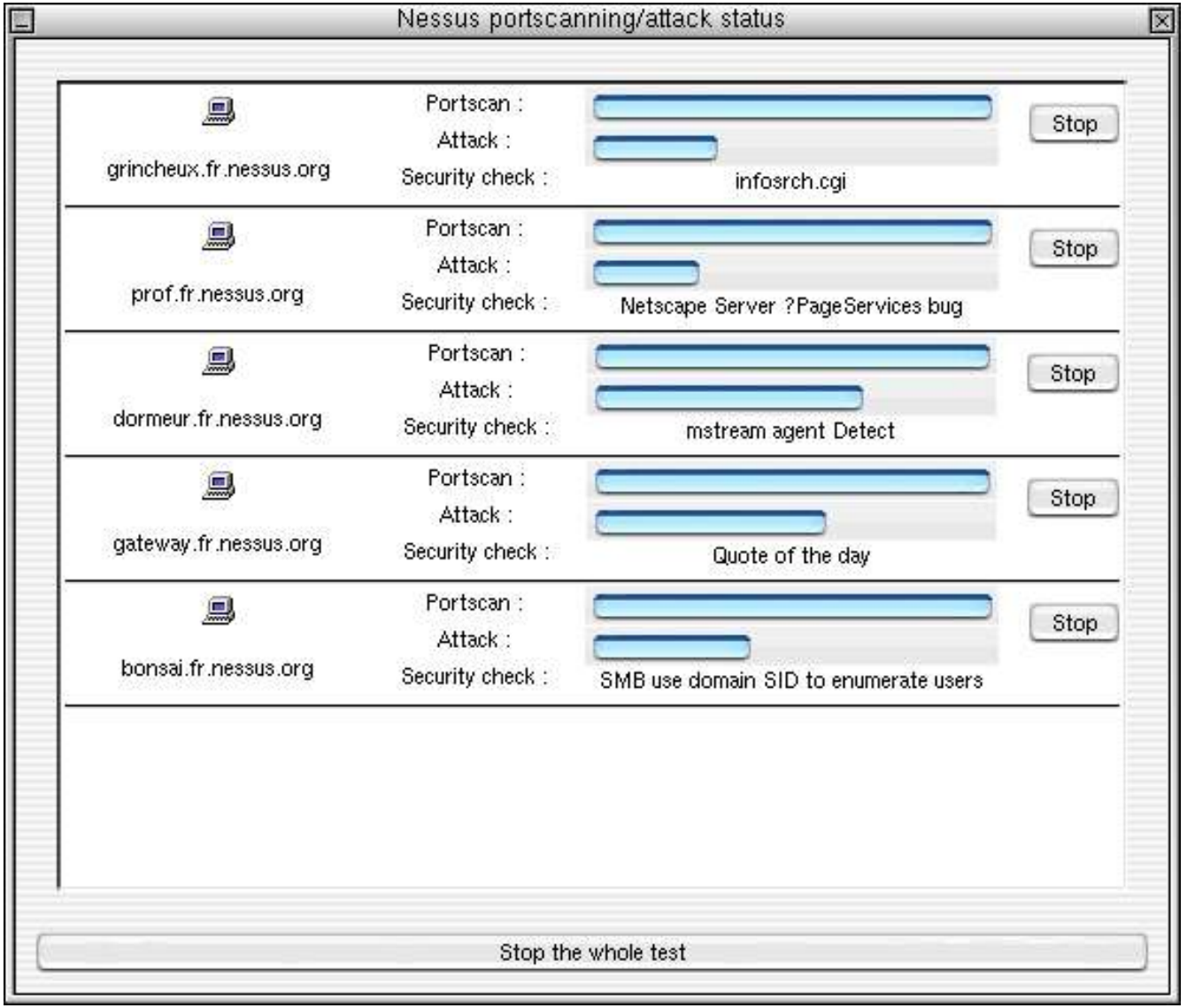
- up2date security vulnerabilities database
- Client Server Architecture
- Can test number of hosts at the same time
- Complete report of vulnerabilities, risk level, and fixes
- Exportable report - as ASCII, XML, HTML
- Full SSL support for https, imaps, smtps
- Destructive/Non-Destructive test option

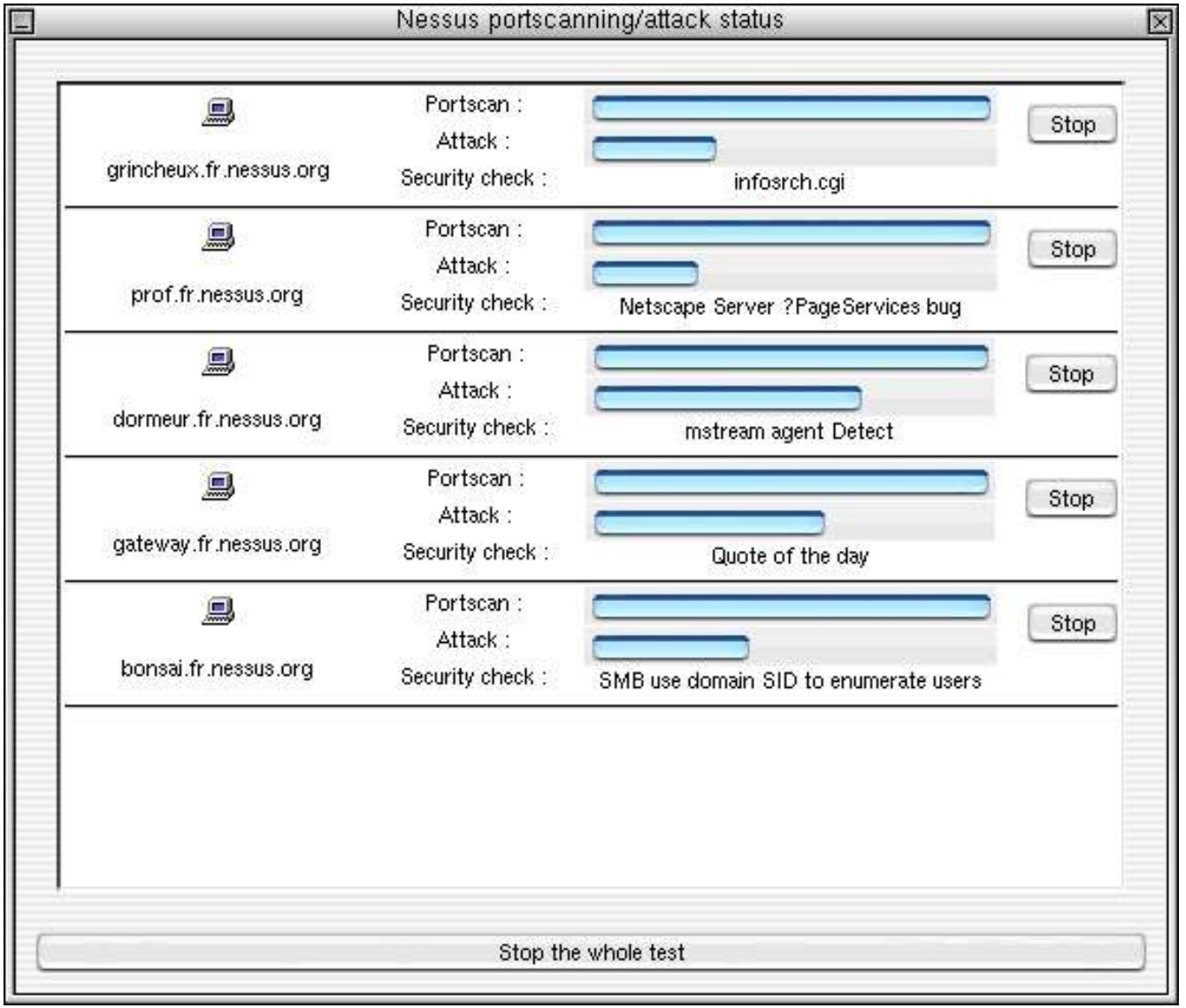
- Nessus is made up of two parts client and server
  - Server: Unix like system required
  - Client: Unix like system / Windows
- Comes as a standalone package that auto-installs itself
- Before running nessus please make sure that nessus can bypass the firewall policy for effective estimation of vulnerabilities


Mail
server
Scan hosts/services
DNS
server
Nessus client
Linux / Windows
Nessus
Server
Web
server

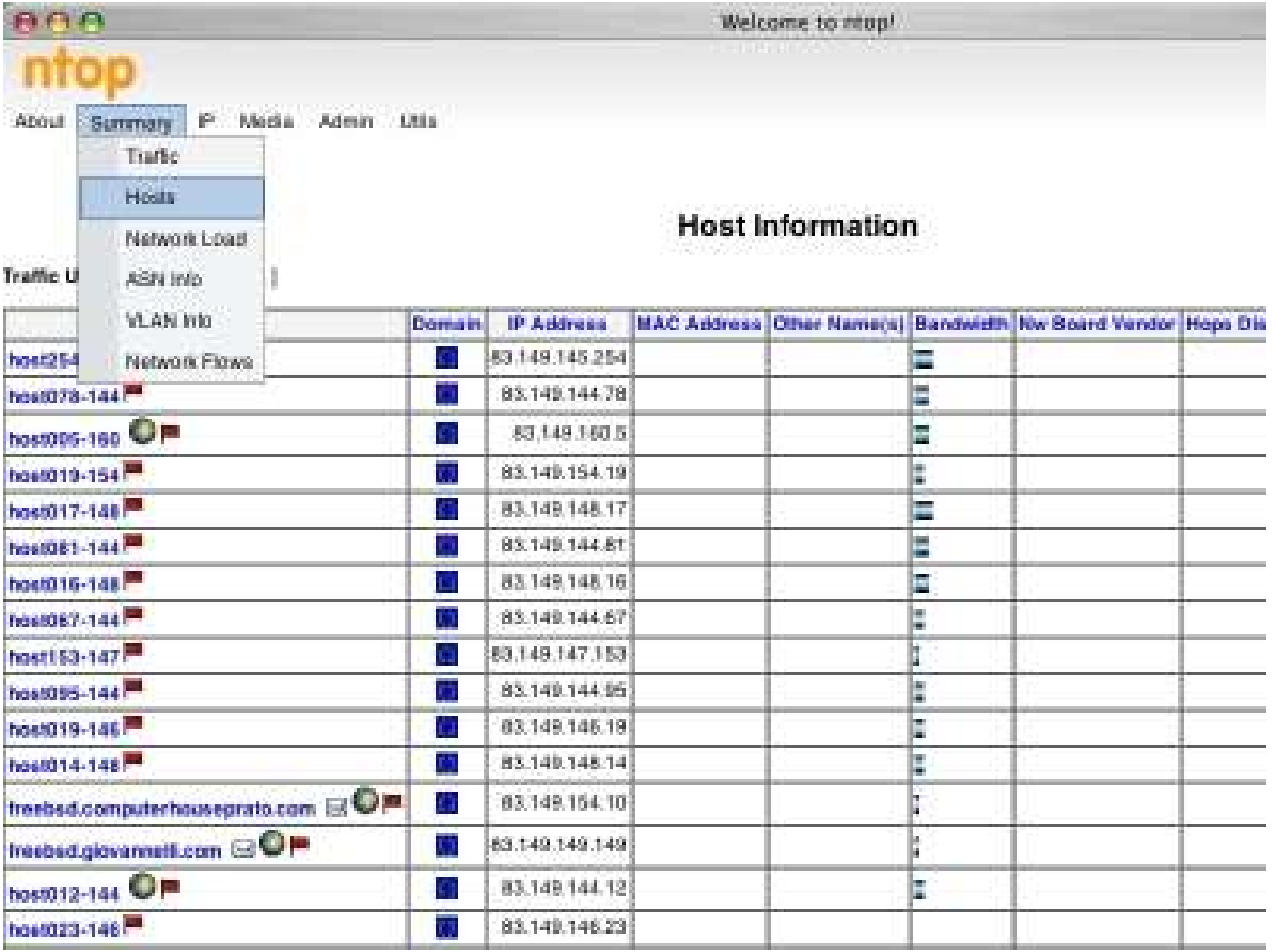

Nessus Setup
Nessusd host
Plugins
Prefs.
Scan options
Target selection
User
Credits
New session setup
Nessusd Host : prof
Port : 3001
Encryption : twofish/ripemd160:3
Login : r
Log in
Start the scan
Load report
Quit

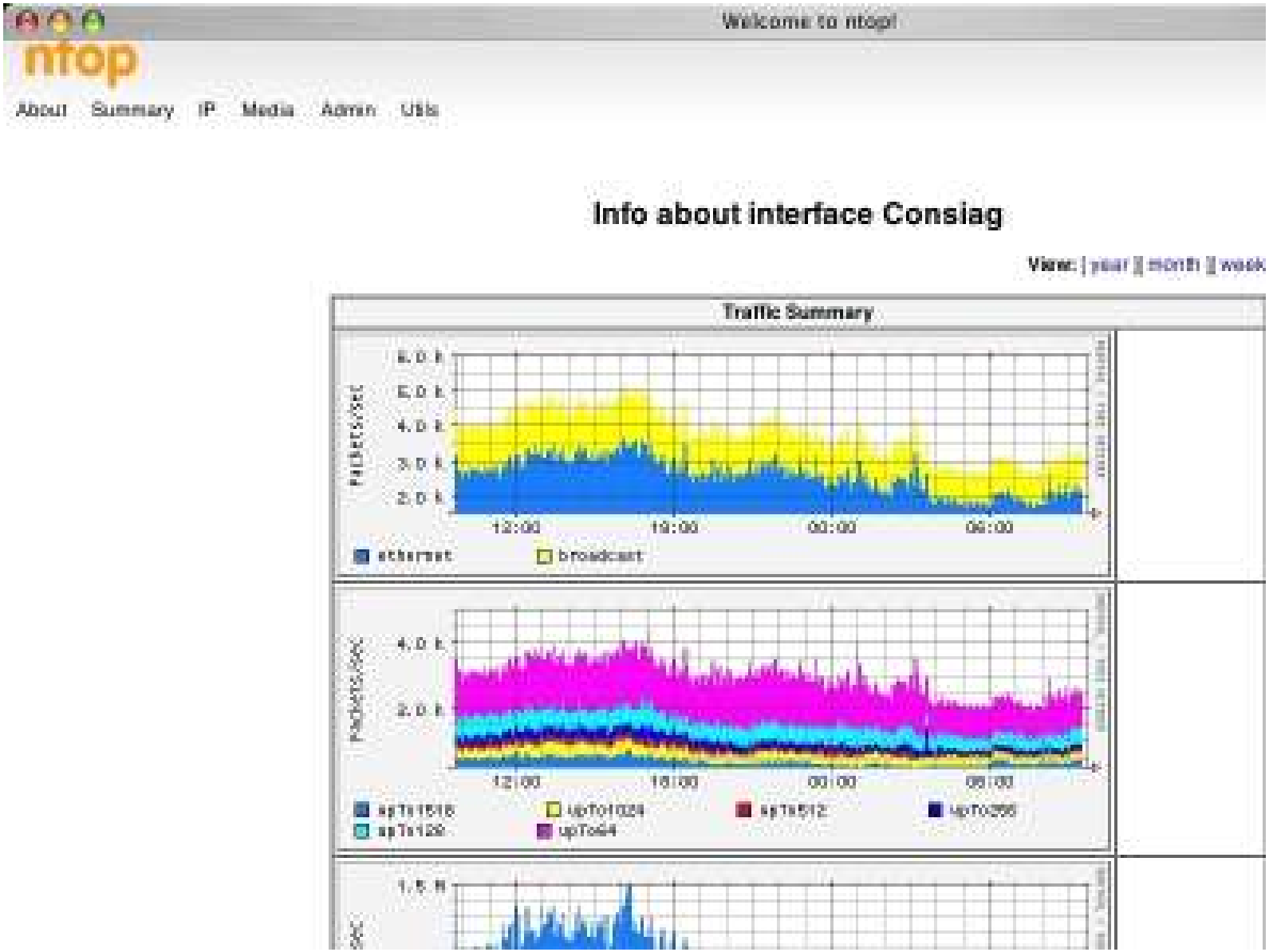

Nessus Setup
Nessusd host
Plugins
Prefs.
Scan options
Target selection
User
Credits
Plugin selection
Misc.
Finger abuses
Backdoors
CGI abuses
General
Remote file access
RPC
Gain a shell remotely
Firewalls
Windows
SMTP problems
Enable all
Enable all but dangerous plugins
Disable all
Using NetBIOS to retrieve information from a Windows host
SMB log in
SMB accessible registry
SMB Registry : Service Pack version
SMB get domain SID
SMB use domain SID to enumerate users
SMB LanMan Pipe Server browse listing
SMB shares enumeration
Start the scan
Load report
Quit


Nessus portscanning/attack status
grincheux.fr.nessus.org
Portscan :
Attack :
Security check :
infosrch.cgi
Stop
prof.fr.nessus.org
Portscan :
Attack :
Security check :
Netscape Server ?PageServices bug
Stop
dormeur.fr.nessus.org
Portscan :
Attack :
Security check :
mstream agent Detect
Stop
gateway.fr.nessus.org
Portscan :
Attack :
Security check :
Quote of the day
Stop
bonsai.fr.nessus.org
Portscan :
Attack :
Security check :
SMB use domain SID to enumerate users
Stop
Stop the whole test


Nessus portscanning/attack status
grincheux.fr.nessus.org
Portscan :
Attack :
Security check :
infosrch.cgi
Stop
prof.fr.nessus.org
Portscan :
Attack :
Security check :
Netscape Server ?PageServices bug
Stop
dormeur.fr.nessus.org
Portscan :
Attack :
Security check :
mstream agent Detect
Stop
gateway.fr.nessus.org
Portscan :
Attack :
Security check :
Quote of the day
Stop
bonsai.fr.nessus.org
Portscan :
Attack :
Security check :
SMB use domain SID to enumerate users
Stop
Stop the whole test

- 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
83.149.145.254
83.149.144.78
83.149.160.5
83.149.154.19
83.149.148.17
83.149.144.81
83.149.148.16
83.149.144.67
83.149.147.153
83.149.144.95
83.149.146.19
83.149.148.14
freebsd.computerhouseprato.com
83.149.154.10
freebsd.giovanelli.com
83.149.149.149
83.149.144.12
83.149.148.23

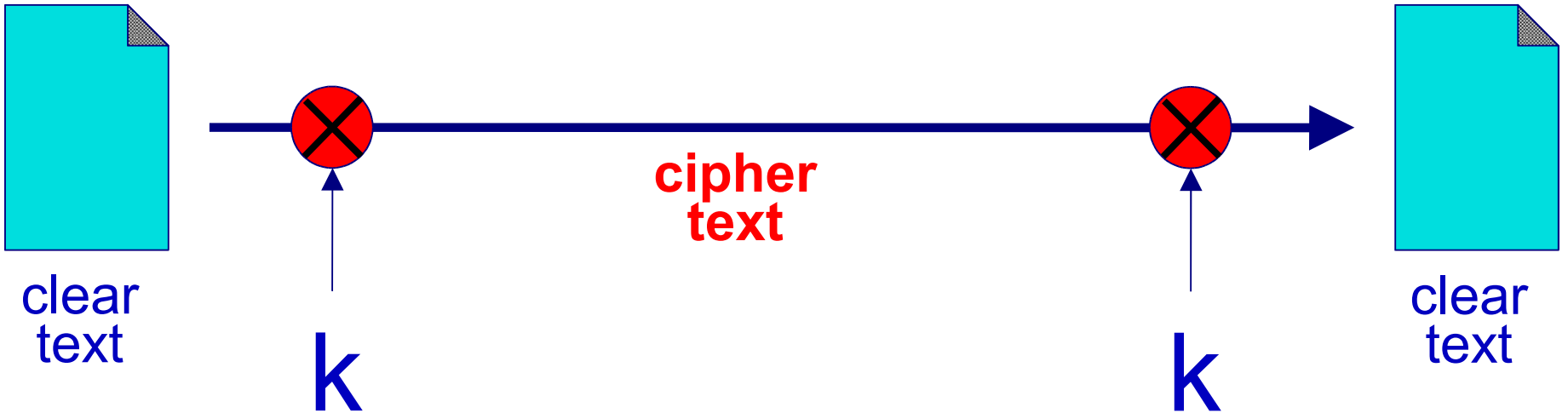

Welcome to ntop!
ntop
About
Summary
IP
Media
Admin
Utils
Info about interface Consiag
View: | year | month | week
Traffic Summary
12:00
18:00
00:00
06:00
ethernet
broadcast

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

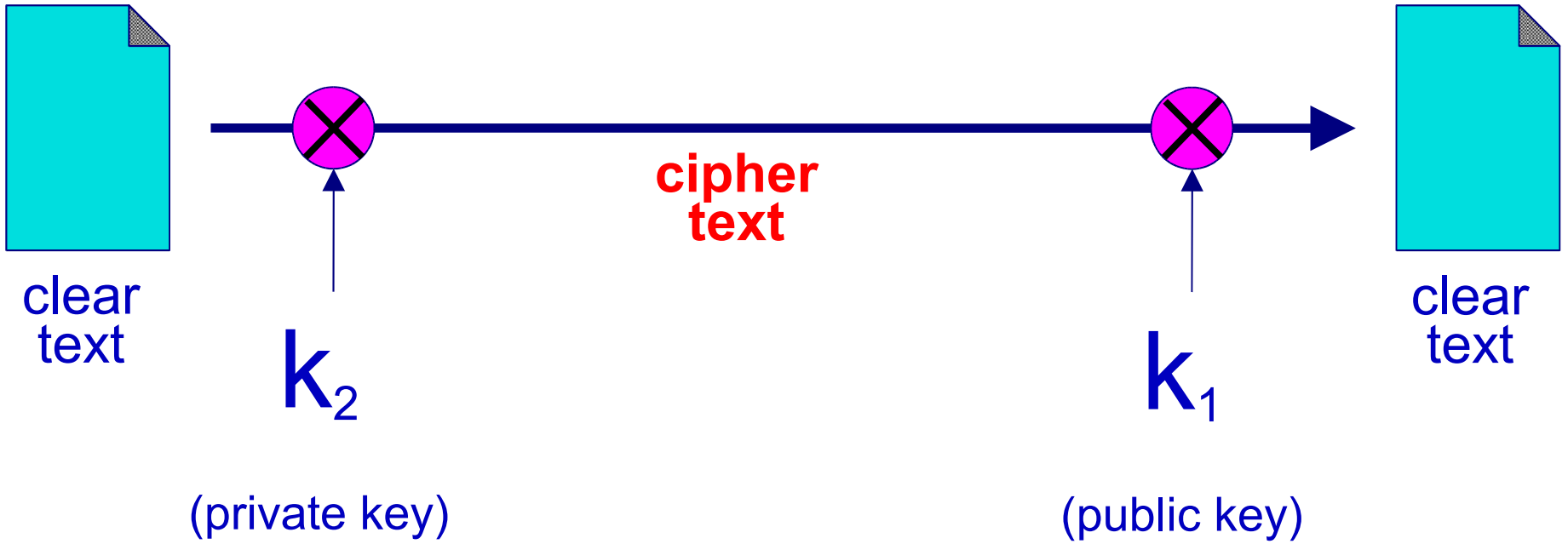
- Traditional Ciphers
  - Substitution ciphers
    - each letter of group of letters is replaced by another letter of group of letters
  - Transposition Ciphers
    - Reorder the the letters
  - One time pad
    - Here the plaintext is converted into a bit string which is Xor(exclusive OR) another one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


clear
text
k
cipher
text
k
clear
text

- 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


clear
text
k2
(private key)
cipher
text
k1
(public key)
clear
text

- 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
- Used 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 starndard output
- You can also use “vim” Editor to edit and save encrypted gpg files

- Create your Signing Key
  - $ gpg --gen-key
    - Select RSA
    - Enter Keysize (2048 should be sufficient)
    - Enter the Expiry date (0 means no expiration set)
    - Enter you Name
    - Enter your Email address
    - Type your passphrase (Donot use NULL Passphrase)
- Create your Encrypt key
  - $ gpg --edit-key “Your Email address”
    Command> addkey
    - Select RSA (encrypt only)
    - Enter Keysize (2048)
    - Enter the Expiry Date

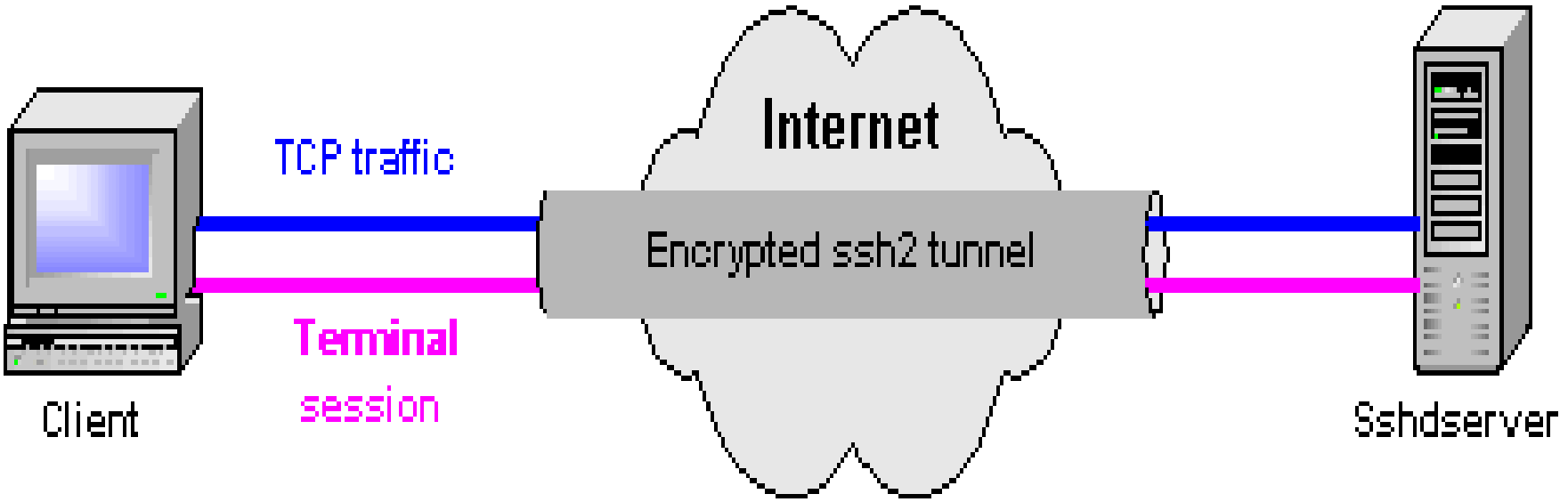
- FreeBSD provides File system encryption by GBDE
- Geometry Based Disk Encryption
- It uses AES 128 bit Encryption for Writing data to disk
- It uses AES 256 bit for storing the master key
- GBDE can be enabled in kernel by adding the following to the config file:
  - options GEOM_BDE
- GBDE can be loaded as module using
  - kldload geom_bde

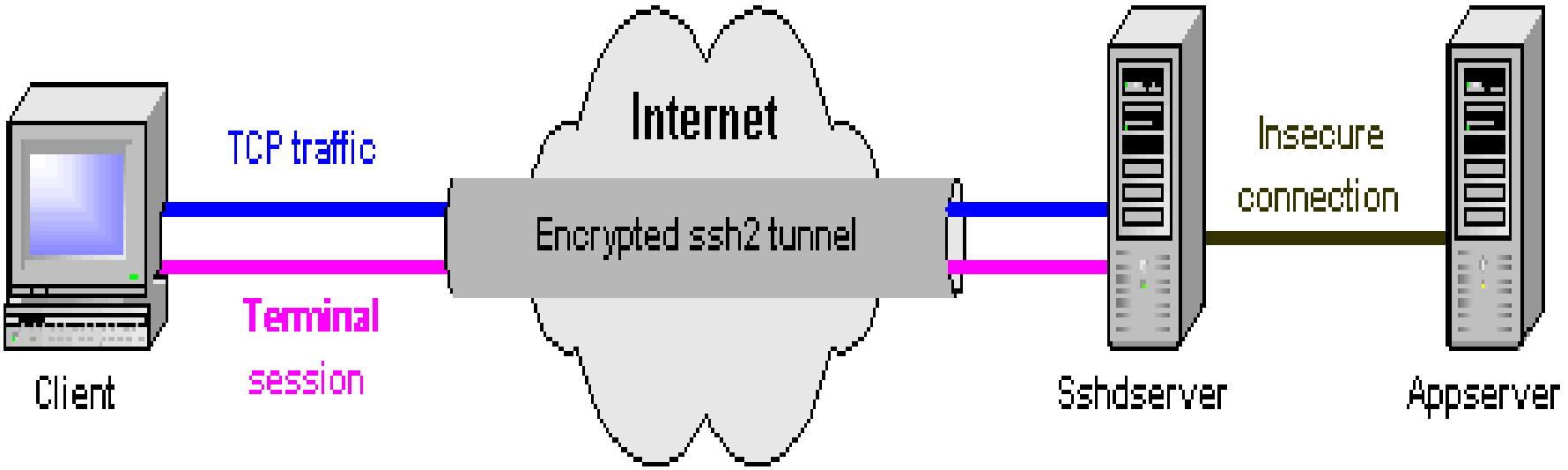
- Initialize gbde Disk
  - gbde init /dev/ad2s1a -i
  - Change sector_size to 2048 for UFS filesystem
  - Assign the password for the gbde disk
- Attach a gbde disk to the kernel
  - gbde attach /dev/ad2s1a
  - enter the disk password
  - A new dev entry will be created /dev/ad2s1a.bde
- Fill random data on the disk
  - dd if=/dev/random of=/dev/ad2s1a.bde
- Create a filesystem
  - newfs -U -O2 /dev/ad2s1a.bde
    - -U enable soft update
    - -O2 Use UFS2
- Mount the drive
  - mount /dev/ad2s1a.bde /private

- Unmount the encrypted disk
  - umount /private
- Detach the Encrypted disk
  - gbde detach /dev/ad2s1a
- The disk /dev/ad2s1a cannot be mounted without the password
- Trying to mount the disk /dev/ad2s1a will produce an error
- You mount the disk using gbde when data needs to read/written and unmount it afterwards
- Nobody without the valid disk password will be able to access the disk data

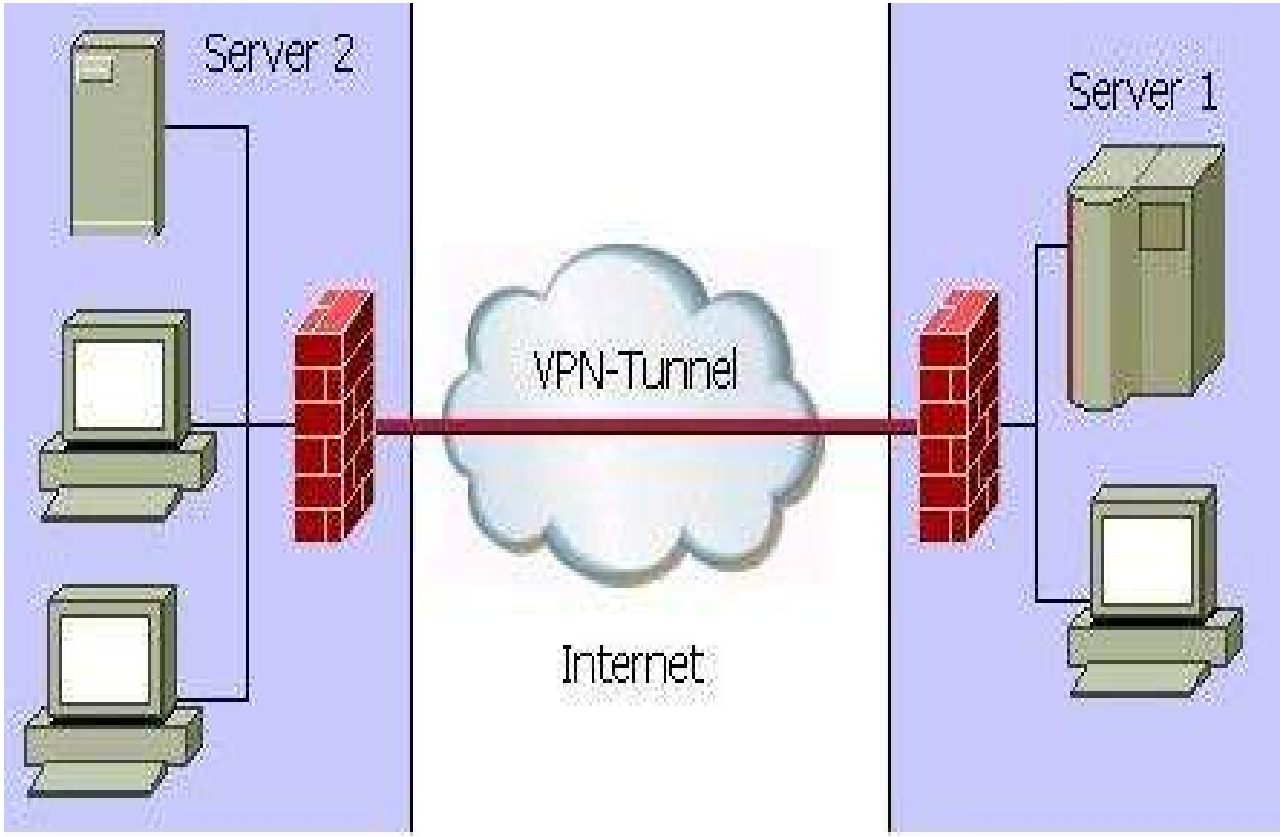
- Secure Shell
- SSH is a suite of protocols used as a secure replacement for rlogin, rsh and rcp
- It can use various ciphers to encrypt data between two hosts
- It supports Public key based authentication
- It allows secure login and secure copy of files
- It can provide data compression
- It can provide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


Internet
TCP traffic
Encrypted ssh2 tunnel
Terminal
session
Client
Sshdserver

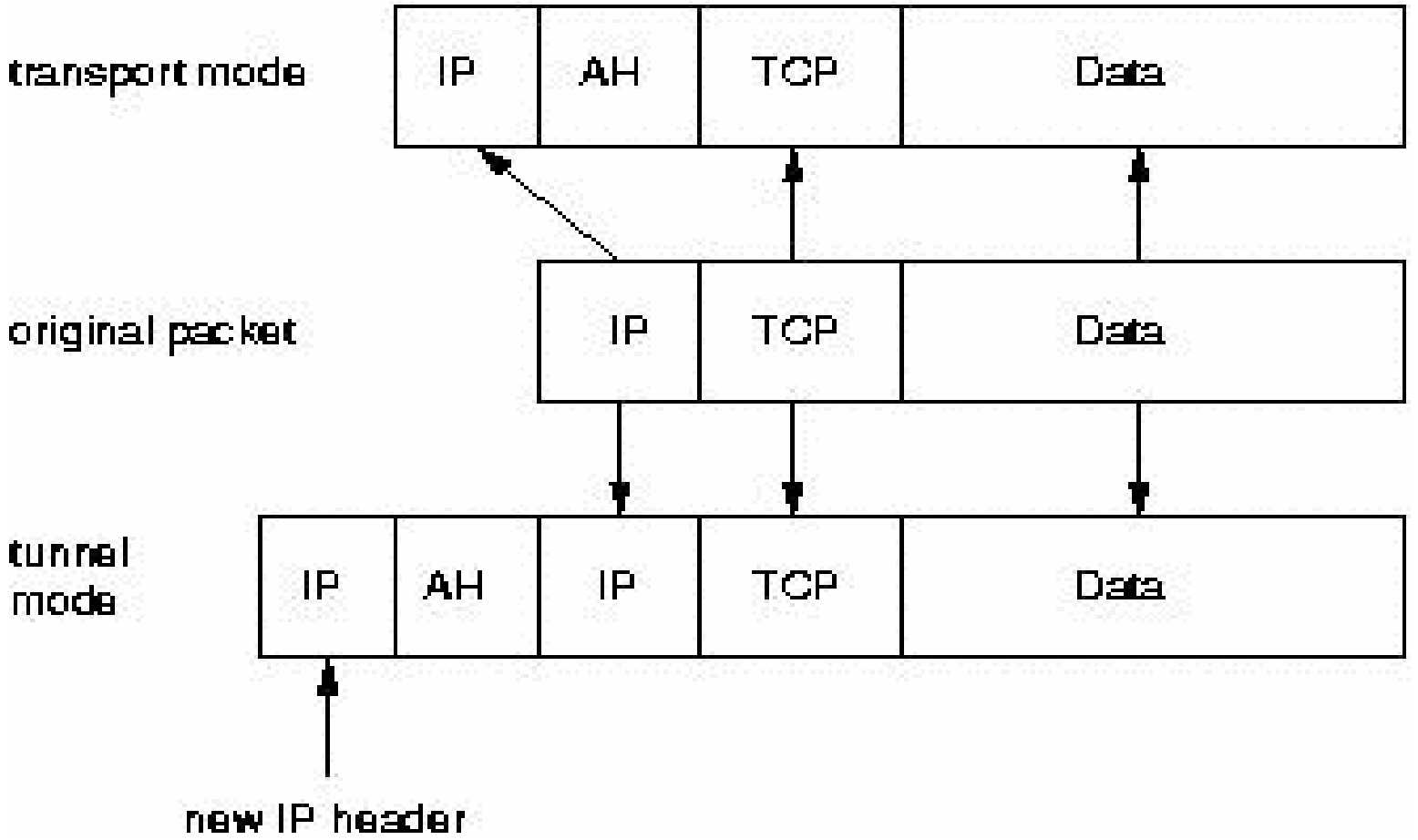

Internet
TCP traffic
Encrypted ssh2 tunnel
Terminal
session
Insecure
connection
Client
Sshdserver
Appserver

- Internet Protocol Security
- An effort by IETF to create cryptographically secure communications at the IP network level
- Uses Strong cryptography for both authentication and encryption services
- You can build secure tunnels through unsecure networks such as the Internet
- The data passing by this tunnel is encrypted by the originating gateway and decrypted by the terminating gateway
- It can be used to build Virtual Private Networks
- It was first developed for the IPv6 standard and then back ported to IPv4

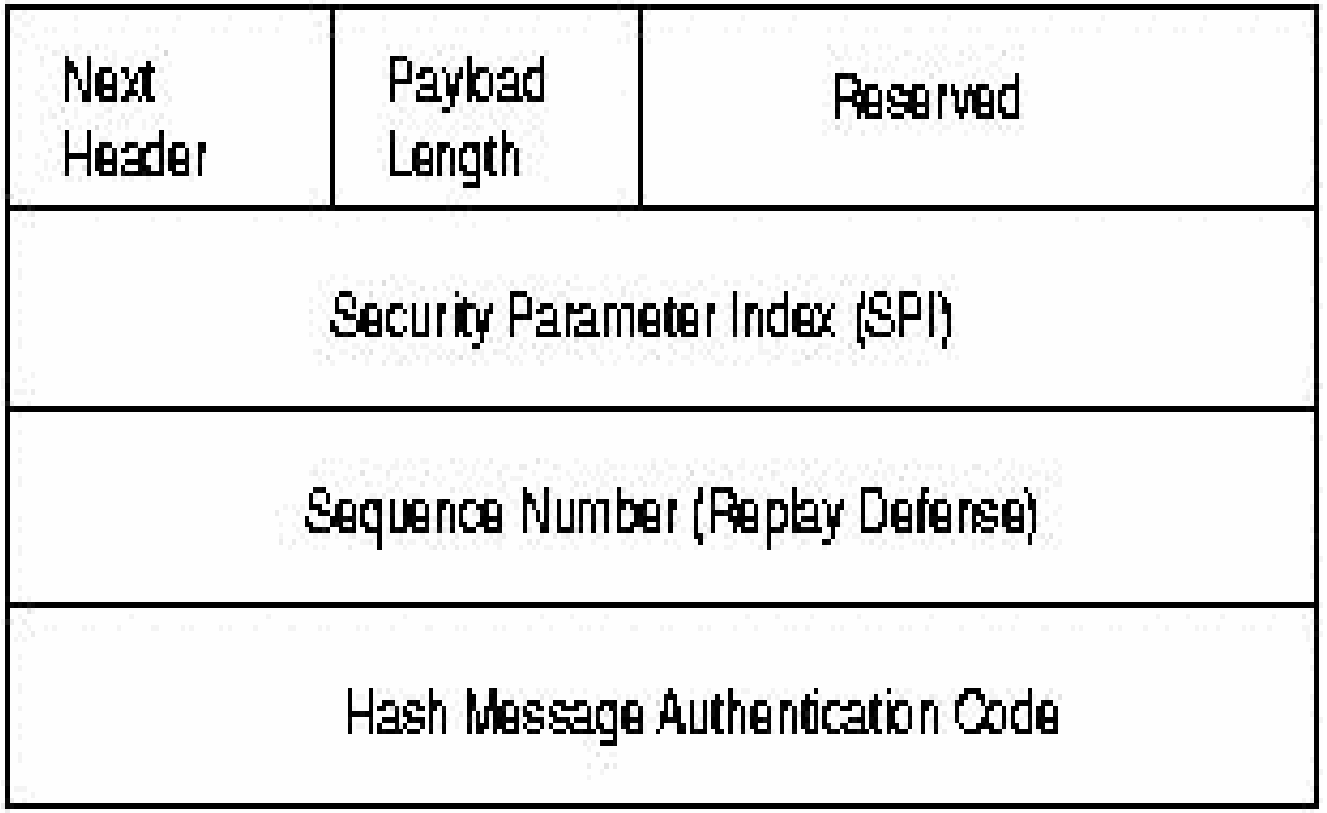

Server 2
Server 1
VPN-Tunnel
Internet

- To protect the integrity of the IP datagrams IPSEC uses Hash Message Authentication Codes (HMAC)
- To derive this HMAC , It uses hash algorithms such as MD5 and SHA to calculate a hash based on a secret key and the contents of the IP datagram
- This HMAC is then included in the IPSEC protocol header and the receiver of the packet can check the HMAC if it has access to the same secret key

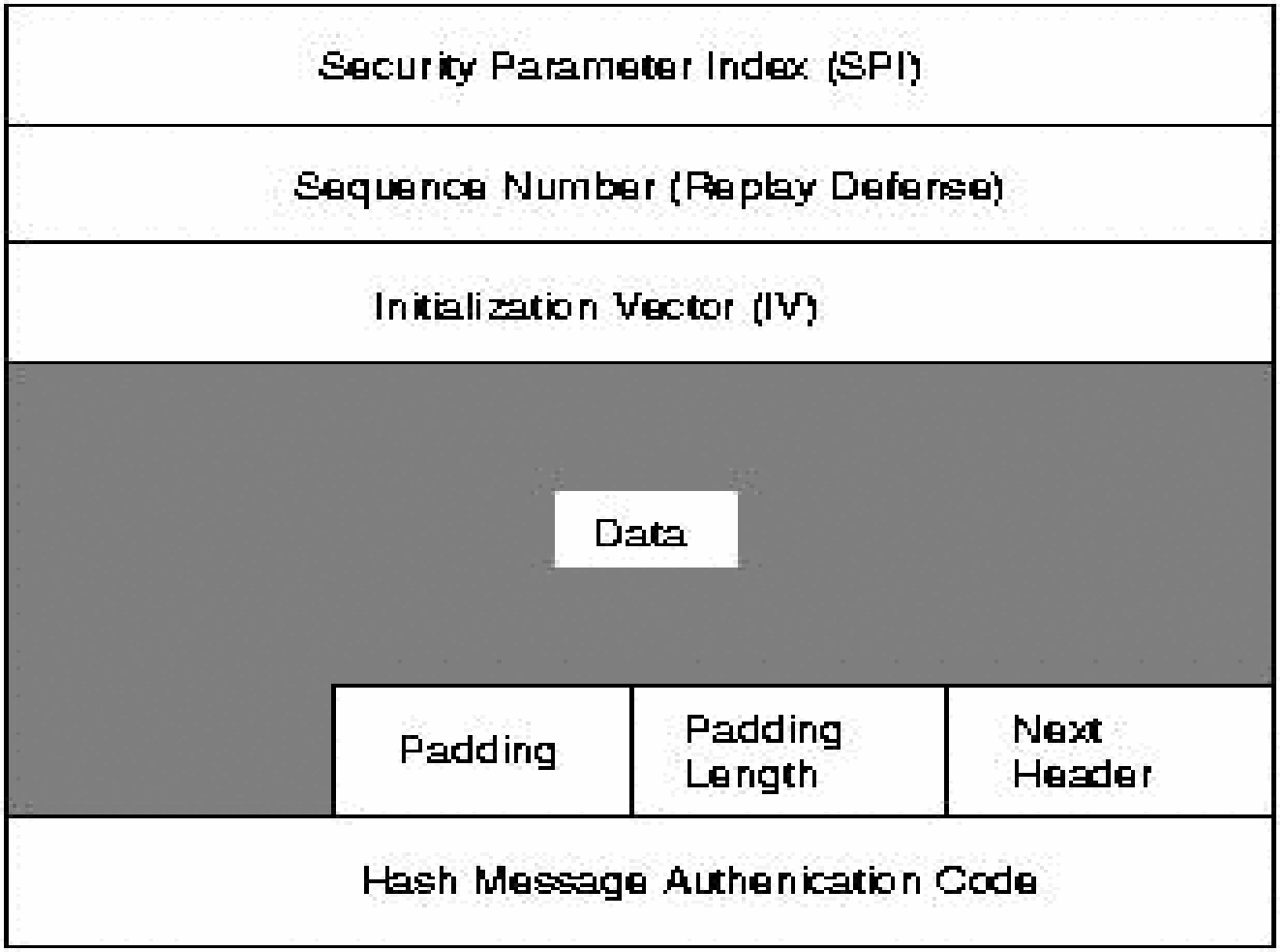
- To protect the confidentiality of the IP datagrams, IPSEC protocol uses standard symmetric encryption algorithms
- For the peer to be able to encapsulate and decapsulate the IPSEC packets they need a way to store the secret keys, algorithms and IP addresses involved
- All these parameters needed for the protection of the IP datagram are stored in a Security Association (SA).
- The Security Associations are in turn stored in a Security Association Database (SAD)


transport mode
IP
AH
TCP
Data
original packet
IP
TCP
Data
tunnel mode
IP
AH
IP
TCP
Data
new IP header

- Authentication Header (AH)
  - The AH protocol protects the integrity of the IP datagram
  - The AH protocol calculates a HMAC on the basis of the secret key, payload of the packet and immutable parts of the IP header like IP addresses
  - It then adds the AH header to the packet


Next Header
Payload Length
Reserved
Security Parameter Index (SPI)
Sequence Number (Replay Defense)
Hash Message Authentication Code

- Encapsulated Security Payload (ESP)
  - ESP protocol can both ensure the integrity of the packet using a HMAC and the confidentiality using encryption
  - After encrypting the packet and calculating the HMAC the ESP header is generated and added to the packet


Security Parameter Index (SPI)
Sequence Number (Replay Defense)
Initialization Vector (IV)
Data
Padding
Padding Length
Next Header
Hash Message Authentication Code

- IKE Protocol
  - The IKE protocol solves the most prominent problem in the setup of secure communication
    - Authentication of Peers
    - Exchange of symmetric keys
  - It then creates the security associations and populates the SAD
  - The IKE protocol usually requires a user space daemon and is not implemented in the Operating System
  - It used 500/udp port for its communication

- The IKE protocol functions in two phases
  - The first phase establishes a Internet Security Association Key Management Security Association (ISAKMP SA)
  - In the second phase the ISAKMP SA is used to negotiate and setup the IPSEC SAs
- The authentication of the peers in the first phase can usually be based on Pre-Shared Keys (PSK), RSA Keys and X.509 certificates

- IPSEC in FreeBSD is based on the Kame implementation
- IPSEC can be enabled by adding the following options to kernel config file during compilation
  - options IPSEC
  - options IPSEC_ESP
- racoon can be install from ports which is the IKE daemon
- If you want to create tunnels you need General Interface support in the Kernel
  - pseudo-device gif

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

- 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 - IPFIREWALL
  - 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 know patterns in real time
- IDS can give valuable information on unauthorized access to your network
- IDS can be Host based of 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
  - Tripwire

- Tool written to detect know Trojans and root kits installed in the system
- Check if the Ethernet interface is promiscuous mode
- Check if the lastlog/wutmp 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

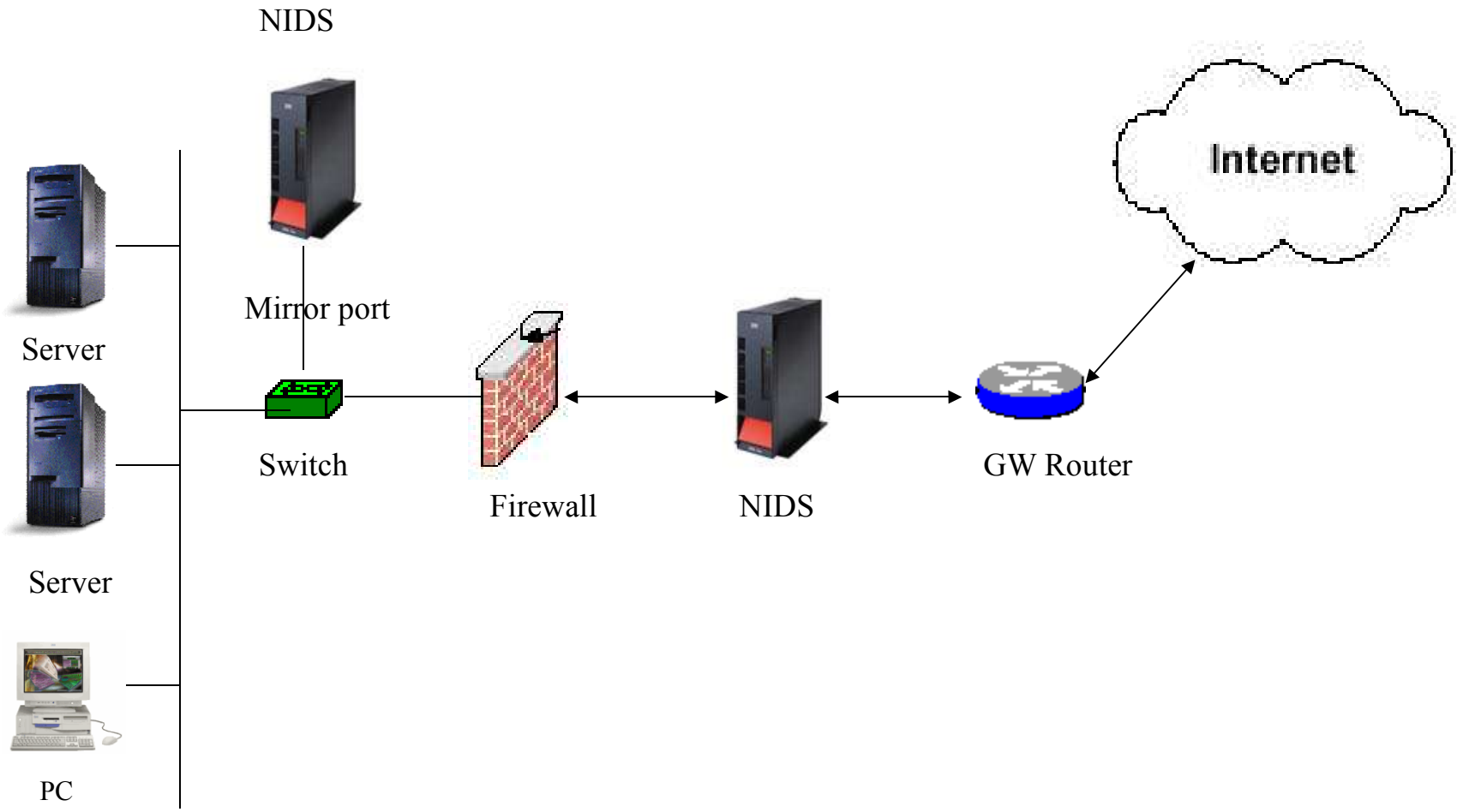
- Network Intrusion Detection System (NIDS)
- Inspects/Sniffs all network traffic for abnormal contents
- Provides a layer of defense which monitors network traffic for predefined suspicious activity or patterns
- Has built in signature base and anomaly detection
- String search for Known signatures with logging and reset features
- Help identify the source of incoming probes, scans or attacks
- Alert system administrators when potential hostile traffic is detected
- Similar to a security camera of a burglar alarm

- A cross-platform, lightweight network Intrusion detection tool
- 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

- Detection engine is programmed using simple language that describes per packet test and actions
- Ease of use simplifies and expedites the development of new exploit detection fules
- Useful when it is not cost effective to deploy commercial NIDS Sensors
- Architecture is focused on performance, simplicity and flexibility
- Works with libpcap sniffing library
- Available under the GNU Public License
- Free

- There are three primary subsystems
  - Packet Decoder
  - Detection Engine
  - Logging and Alerting

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


NIDS
Internet
Server
Mirror port
Server
Switch
Firewall
NIDS
GW Router
PC

SPAM

- Unsolicited Commercial Email (UCE)
- Mass mailing done by Spammer or Virus/Worms
- Waste of Bandwidth and Server Resources
- Annoying and Harassment for the recipient
- 85 % of all Email traffic are Spam (Based on the calculation done after implementing spam protection in my ISP)
- ISP needs to play a vital role in fighting against Spam

- Do not Run Open Relay SMTP Servers
- Use SMTP auth or POP Before SMTP services
- Use of Real Time Black Lists(RBL)
  - bl.spamcop.net
  - sbl.spamhaus.org
  - dnsbl.njabl.org
- Use of white list and Black list for “mail from” and “rcpt to”
- Use of Anti Spam software such as Spam Assassin in the mail servers
- Use of Anti-virus Software such as qmail-scanner and clamAV
- Use of Greylisting
- Use of Sender Policy Framework (SPF) database

- Restricting SMTP traffic from Host that does not Reverse DNS entry
- Restricting SMTP traffic that don't strictly follow the RFC
- Use of tarpit to allow only a certain number of “rcpt to” in one smtp transaction
- Have strict policy against SPAM and Spammer

- A more harsher method will be deny Direct SMTP connection to/from the client network
  - ```
    ! Permit the client to smtp only to the ISP SMTP Network
    access-list 150 permit tcp any 203.15.23.0 0.0.0.255 eq 25
    access-list 150 permit tcp 203.15.23.0 0.0.0.255 eq 25 any
    ! Deny any other SMTP traffic to/from Client Network
    access-list 150 deny tcp any any eq 25
    access-list 150 deny tcp any eq 25 any
    ! add other protection rules..
    ....
    !
    access-list 150 permit ip any any

    int Serial 0/0
    description 2 Mbps Link to Client A
    ip access-group 150 in
    ip access-group 150 out
    ```
    ```

Thank You