Common Threats & Vulnerabilties

- Threat:

  Any person, object, or event that, if realized, can potentially cause damage to the network or networked device

- Vulnerability:

  A weakness in a host or network that can be exploited by a threat

Common Threats

- Unauthorized Intrusions
- Denial of Service (DoS) Attacks
- Viruses, Worms, Trojan Horses (Backdoors)
- Website Defacements
- Internal Attacks
- Non-compliance

Unauthorized Intrusions

- An unauthorized entity gains access to an asset and has the possibility to tamper with that asset

By:

- Intercepting authentication information in transit over an insecure channel
- Using this information to attain unauthorized access
- Exploiting an inherent weakness in a technology or a product

Unauthorized Intrusions

- The damage created depends on the intruder's motives
- Confidential information maybe compromised and possibly altered/damaged
- Shared media networks (hubs) are particularly susceptible to eavesdropping

Unauthorized Intrusions

- Cryptographic authentication mechanisms can prevent impersonation attacks
- PKI, Public Key/Private Key, Digital Certificate
- Non-repudiation is also achieved
- Verification is critical for electronic financial transactions
- A digital signature confirms the identity of the sender and the integrity of the contents of the data being sent

Denial of Service

- Interruption of service either because the system is destroyed or is temporarily unavailable

- e.g.
  - Destroying a computer's hard disk
  - Severing the physical infrastructure
  - Using up all available system resource - CPU, memory, disk space
  - Consuming network bandwidth to the server

Denial of Service

- Can be avoided by applying vendor patches to affected software
- By securing always-on hosts with broadband connectivity – DSL, Cable, etc.
- Few DoS attacks cannot be stopped, but their scope of affected areas can be constrained
- Most common – SYN Flood attack

Viruses, Worms, Trojans (Backdoors)

- A virus requires a user to do something to continue the propagation
- A worm can propagate by itself - self-propagating malicious code
- Highly prevalent/common on the Internet
- Common distribution: e-mail, ftp, media sharing, hidden codes

Viruses, Worms, Trojans (Backdoors)

- Trojans (Backdoors) - Executable codes installed that enable entry without authorization
- Once installed the back door can be used by the attacker at their leisure
- Launching points for further security attacks (DDOS, SPAM)
- Variants include: Klez, Nimda, Code Red, Melissa, Back Orifice, or other Trojans
- Damage: Lost records, deleted system files, corrupted system/data files, distribution of private information/password

Viruses, Worms, Trojans (Backdoors)

- The highly-automated nature of the worms coupled with the relatively widespread
- nature of the vulnerabilities they exploit allows a large number of systems to be
- compromised within a matter of hours.
- (Code Red infected more than 250,000 systems in just 9 hours on July 19, 2001.)

Viruses, Worms, Trojans (Backdoors)

- Some worms include built-in denial-of-service attack payloads (Code Red)
- Creates a DoS in many parts of the Internet because of the huge amounts of scan traffic generated
- Cause much infrastructure damage due to high resource utilizations by processing unusually huge traffic

Website Defacements

- Intent: To create political propaganda based attacks
- Deface a website to make a political statement
- Launched primarily at Government Orgs, Media, Religious Groups
- Exploit vulnerabilities in websites or servers
- The attacker can plant codes or files to vandalize site
- Examples at: http://www.attrition.org/mirror/attrition

Internal Attacks

- Computer Security Institute/FBI and Ernst & Young say nearly 50% of all network attacks come from the inside
- Often, from unhappy workers
- 76% of the IT executives surveyed by NetVersant said they were concerned about inside attacks from unhappy employees
- Losses associated with insider attacks can be more damaging

Non-compliance

- Security policies and procedures not followed properly by all concerned staff
- Who cares how good your systems are if employees ignore them?
- NetVersant survey: 82% reported spotty or no compliance with their company's network security policies
- 85% say a properly-implemented firewall would still be at risk from a disgruntled employee
- And 75% say the firewall is at risk from employee incompetence

Other Common Attacks

- Connection (Session) hijacking
- IP source address spoofing
- Smurf attack
- Brute-force/Dictionary attacks (password guessing)

- Humans are often the weakest link

  "Hi, this is Bob, what's the root password?"

Vulnerabilities

- Insecure protocols/services running on a host
- Exploitable security hole on a host without latest patches or workarounds
- Poorly protected hosts and networks without firewalls, IDSs, etc.
- Use of weak or default passwords
- Insecure configuration of hosts
- Execution of malicious codes – Trojan, Backdoors
- Use of pirated or downloaded software from a public site without verifying checksum (integrity) and authenticity (signature)
- Social engineering

Motivation of Threat

- Understanding some of the motivations for an attack is important
- Can give you some insight about which areas of the network are vulnerable
- What actions an intruder will most likely take

- Common perception is that, in many cases, the attacks occur from the external Internet

- Attacks are equally likely to happen from internal users – more easily with the granted privileges

Common Motivations for Attacks

- Greed:
  The intruder is hired by someone to break into a corporate network to steal or alter information for the exchange of large sums of money.

- Prank:
  The intruder is bored and computer savvy and tries to gain access to any interesting sites.

- Notoriety:
  The intruder is very computer savvy and tries to break into known hard-to-penetrate areas to prove his or her competence.
  To gain the respect and acceptance of his or her peers.

Common Motivations for Attacks

- Revenge:
  The intruder has been laid off, fired, demoted, or in some way treated unfairly. Attacks result in damaging valuable information or causing disruption of services

- Ignorance:
  The intruder is learning about computers and networking and stumbles on some weakness, possibly causing harm by destroying data or performing an illegal act

- There is a large range of motivations for attacks
- Consider all these motivations as possible threats

Hardening Linux Installations

Hardening Linux Installations

The default installation of Linux distributions may not be optimized for security. However, we can customize an installation with security in mind.

Here are a few points to be considered while performing a Linux Server installation.

- Secure Partitions
- Service Installations
- Boot Loader Password
- Shadow Suite

Securing Partitions

When possible make separate partitions. I usually prefer the following partition scheme:

- /boot
- /
- /usr
- /var
- /tmp
- /home
- swap

The key benefit of separate partitions is that one can control what partitions are mounted with selected options. For example, one can use mount options such as

Securing Partitions ..contd

Here is my sample/etc/fstab

/dev/hda1	/boot	ext3	ro,noexec,nosuid,nodev	1 1
/dev/hda2	/	ext3	defaults	1 2
/dev/hda3	/usr	ext3	defaults	1 2
/dev/hda5	/var	ext3	noexec,nosuid,nodev	1 2
/dev/hda6	/tmp	ext3	noexec,nosuid,nodev	1 2
/dev/hda7	/home	ext3	noexec,nosuid,nodev	1 2
/dev/hda8	swap	swap	defaults	0 0

Service Installations

- Install only the services that you will be running on the server.
- You may not want to install services such as portmap, nfsd, rshd, bootpd, tftpd which are more vulnerable if not properly secured.

Boot Loader Password

- It's always advised to install a boot loader password so that unauthorized people cannot modify the boot parameters during start-up.

Boot Loader Password ..contd

Installation instructions for lilo (Linux Loader)

`#vi /etc/lilo.conf`

Add the following lines:

```
password your_password
restricted
```

By adding restricted keyword lilo will only prompt for password if you try to modify the boot options

After the modifications do:

`#/sbin/lilo -v`

The above command will reinstall the boot sector reflecting the changes made.

Boot Loader Password ..contd

Installation instruction for GRUB

Generate the password **`#grub-md5-password`**

It will prompt you for Password , enter it twice. Next copy the cryptic password generated above and add it in "/etc/grub.conf"

`#vi /etc/grub.conf`

`password --md5` **Crypt_password_from_grub-md5-password**

You do not need to reinstall grub to apply the changes like it was done in lilo.

Shadow Suite

- One should use shadow package rather than putting the encrypted password in "*/etc/passwd*" file.
- As *passwd* file needs to readable by all, if the local users can access the encrypted password, they can then use password crackers to crack the passwords of other users and gain unauthorized access.
- The shadow package suite creates a separate file "*/etc/shadow*" file that is only readable and writable by root, thereby protecting against local malicious users.
- **Files related to shadow suite are:**
  - /etc/shadow
  - /etc/gshadow

Questions & Answers

Securing Services

Securing Services

•There are different ways to secure services running on a Linux box depending on which services are being used.

- Use of TCP wrappers
- FTP security
- Telnet security
- Mail security
- Portmap
- R services
- Apache

TCP Wrapper

TCPwrapper is an utility that intercepts the packets and authorize it before delivering it to the final application.

There are two configuration files for TCP Wrapper.

- */etc/hosts.allow*
- */etc/hosts.deny*

As the name suggests, we use the *hosts.allow* file to allow access to services and *hosts.deny* to deny services.

The format of both files are :

Service : List of IPs/hosts

TCPwrappers ..contd

- I generally put “ALL:ALL” in “*/etc/hosts.deny*” file, meaning disallow all services to All ; And selectively allow access to services via “*/etc/hosts.allow*” file.

- Here is what it looks like:

/etc/hosts.deny

ALL : ALL

/etc/hosts.allow

sshd : 1.1.1.0/24

- However, all the services that can run on Linux can not be protected by TCPwrapper, some programs needs to be compiled with TCPwrapper support, others don not use TCPwrapper at all.

FTP Security

- Don't allow anonymous logins
- Use *chroot*, so that user can access only their own directory
- Deny FTP to system users such as root, bin, mail etc
- Modify the user shell to "*/bin/nologin*" or "*/bin/false*"
- Use sftp wherever possible if you have *ssh* installed

Telnet Security

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

Mail Server Security

- Do NOT run an open relay server. If you run an open relay server, any spammer on the Internet can use your SMTP server to send spam mails. Allow only your local LAN to send emails using your SMTP server.

- Use SMTP after POP if you have users all over the Internet who need to send emails using your SMTP server.

- Use qmail instead of sendmail. Sendmail is a very robust system but takes more time to secure.

- Use an anti-virus if possible.

- Use SSL to protect mail transactions if possible .Prefer SMTPS to SMTP and POP3S to POP3.

Portmap & R Services

- **Portmap**
  - You do not need to use Portmap unless you are using NFS or NIS. I simply disable it. If you are using NFS or other RPC servers then, you should protect it via firewall and TCPwrapper.

- **R services**
  - It is advisable to disable the R services such as *rlogin*, *rsh*, *rcp* etc. You can use *ssh* instead of *rsh* and *sftp* instead of *rcp*.

Apache Web Server

- Run Apache under normal user privilege
- Use “*.htaccess*” for access control
- Disable CGI access
- Use HTTPS to secure the communications between client and the server
- Enable access to http *DocumentRoot* only (*/usr/local/apache/htdocs*).Apache need not access anything outside this directory.
- Disable Index options unless needed

Questions & Answers

System Logging

System Logging

- Linux systems uses *syslogd* for system logging. Some of the critical files that needs to checked frequently for security audits are as follows:

/var/log/messages

/var/log/secure

/var/log/kern

- You will find critical information about network logins failures, failed *su* attempts, and various other useful information.

- You can also log to a central network logging server

System Logging ..contd

- Individual services also generate their own log files. Such as:
  - Apache:
    - /var/log/httpd/access.log
    - /var/log/httpd/error.log
  - proftpd:
    - /var/log/proftpd/auth.log
    - /var/log/proftpd/access.log
    - /var/log/proftpd/xferlog
- Log Analysis tools
  - As the number of servers grows, it can be quite cumbersome to audit logs of all servers. There are numerous programs available on the Internet that can do the job for you such as swatch, logwatch, webalizer etc.

User Administration

User & Group

- Each user in Linux has a unique UID. The 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 user must have a *username* in a system to log in.

- Creating user groups can ease user administration. For example, you can create a directory *"/home/hod"* to give access to all users belonging to the "*hod*" group.

- Generally all users may not need shell access to the server. FTP users, Mail users need not have shell access. You should restrict shell access whenever possible

Root user

- In Linux, the user “*root*” is the super user who has access to virtually everything within the system.

- One should login to root only when it is absolutely necessary. Most of the time one can work as a normal user . You can switch to the super user mode using *su* (substitute user) command.

Protecting su

- You can control who gets to do su and gain superuser priviledges by adding the following line in “/etc/pam.d/su”

auth required /lib/security/pam_wheel.so use_uid

- The above line will not allow su logins from users who are not in the wheel group.

Setting Password Policies

You should also enforce a strong password policy in the system.

Setting password Aging
Set password aging by using the *chage* command.

For example:
setting the expiry date of a user
#chage -E 2004-3-30 guest

setting the max lifetime of a password
#chage -M 30 guest

The password will be valid for 30 days after which the user is prompted to change the password.

You can also change other parameters such as minimum password age, inactive period, warning days etc.

Locking Users

- You can lock a User Account, so that user does not get access to the server.

#usermod -L username

- You can unlock the account by:

#usermod -U username

User Logging

- You can see check the logins of users by using the following commands:

- The **`w`** command shows the logged in users currently in the system
- **`last username`** shows the record of logins of a particular username
- **`lastlog`** shows the record of the last login date and time of all users in the system.

- Files related to these commands are :

/var/log/wtmp
/var/log/lastlog

sudo

- There maybe more than one person handling the server and as a result may need super user privileges to do their job. These users maybe backup operators, mail administrators, Log analyzers who need super user privileges to complete certain tasks.

- Generally one should not give root password to each and every user. Some jobs can be better done by using sudo packages that can be configured such that users can have super user privileges for pre-assigned commands only.

- You can customize the sudo package by editing the "*/etc/sudoers*", where we specify which users can execute which command by using sudo.

- For eg.

```
$ sudo tail -f /var/log/messages
```

sudo ..contd

```
# sudoers file.

# User alias specification
User_Alias      ADMIN = vicky

# Cmnd alias specification
Cmnd_Alias      ADDUSER  = /usr/local/sbin/adduser.sh
Cmnd_Alias      DELUSER  = /usr/local/sbin/deluser.sh
Cmnd_Alias      USER_MGT = ADDUSER,DELUSER

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

File Permissions

- File and Directory permissions basically can be used to control who gets access to a specific resource. Since everything in linux is a file, you can control virtually everything by means of file permission.

Here is the permission of my home directory

```
$ ls -l /home/
total 4
drwx------   29 vicky    vicky        4096 Feb  9 21:08 vicky
```

- The first character denotes that it's a directory.
- The following 3 characters means read, write and execute access to the owner.
- the last 6 characters are blank(-), meaning access is not allowed to group and other users, therefore securing my home directory from other users.

File Permissions .. contd

- Here is the tabulated form of the permissions

1.	d	File type	(directory)
2.	r	read access to owner(vicky)	yes
3.	w	write access to owner	yes
4.	x	execute access to owner	yes
5.	r	read access to group	no
6.	w	read access to group	no
7.	x	execute access to group	no
8.	r	read access to all	no
9.	w	write access to all	no

File Permissions .. contd

- Take a look at the following permissions

```
$ ls -l /tmp
-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 access to owner, group and all other). You must notice the blank '**-**' as the first character; this means it is a regular file.

File Permissions .. contd

- You can change permissions using *chmod*

The syntax is :

#chmod u+rwx filename	give the user full rights, ie., read, write and execute
#chmod u-x filename	remove execute rights from user
#chmod g+rx filename	give the users of the group read and execute rights
#chmod g-x filename	remove execute right from the users of the group
#chmod o+r filename	give read access to all (other)
#chmod o+x filename	remove execute access to all (other).

- The user and group is shown on the 3rd and 4th column when we do a '*ls -l*'.

SUID & SGID

- You should generally be careful about these files. SUID (Set UserID) and SGID (Set GroupID) has special file permissions attribute set , that enables a executable to executed on the privilege of the owner or group. You should probably 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     root        16128 Jun  6  2003 /usr/bin/passwd
```

- The executable passwd will run with the priviledge of user root . This is necessary as it needs to edit “/etc/shadow” file that holds the encrypted passwords of users and is only accessible to root. Normal users normally don't want to chase the System administrators to change their passwords.

Questions & Answers

Best Security Practices

Best Security Practices

- **Hardening System Security**
  - It is not necessary that you use the default configuration that comes with a service. We can configure services to become secure. We can use various tools provided by Linux to tighten it.
- **Shell access**
  - Provide shell access only when absolutely needed. You probably do not need to give shell access to the CEO, who has no knowledge of Linux.
- **Password policy**
  - Enforce strong password policy, so that the password are not dictionary based words and easily crackable. You should also change the passwords frequently.
- **Firewalls**
  - Use Firewall to filter out access to unnecessary services from unnecessary networks.

Best Security Practices .. contd

- **Services**
  - Disable any unnecessary services. There is no point in running portmap and nfsd on a web server that does not need it.
- **TCPwrapper**
  - Use TCPwrapper where possible
- **IDS**
  - Use an intrusion detection system to detect unauthorized access
- **Patches and Updates**
  - Keep your services up to date and watch out for bugs and patch them as soon as possible. Monitor service mailing lists.
- **SSH**
  - Use ssh for remote network logins and file transfers.
- **File Integrity**
  - Use tripwire to verify the integrity of the system files.
- **Backups**
  - Take regular backups

Questions & Answers

Main Data Security Concerns

- Confidentiality

  Keeping our data safe from prying eyes

- Integrity

  Protecting our data from loss or unauthorized alteration

- Authentication and Authorization

  Is this person who they claim to be?

  Is this person allowed to do this?

- Availability

  Are our systems working when we need them?
  (Denial of Service attacks)

Plain-text Passwords

- Can be guessed
- If too complex, users tend to write them down
- If sent unencrypted, can be "sniffed" from the network and re-used

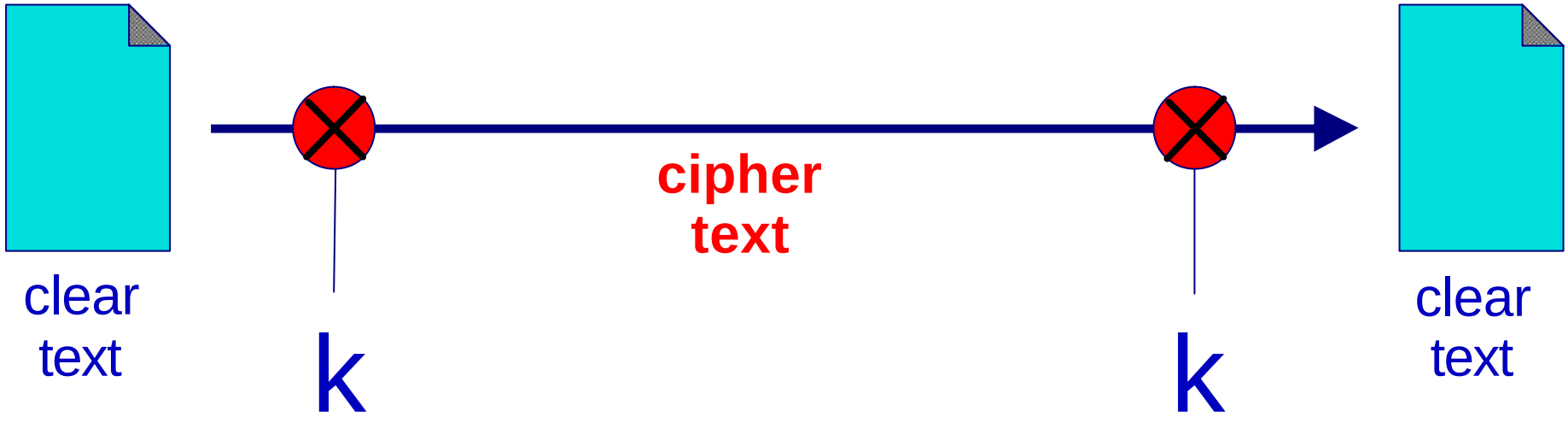
Cryptographic Auth Methods

- Can provide REALLY SECURE solutions to authentication, privacy and integrity
- Some are hard to implement, many different tools, usually requires special clients

Simple Combinations

- The lock on your front door can be picked
- Two locks are better than one
- The thief is more likely to try somewhere else

"Private key" or "symmetric" ciphers



The same key is used to encrypt the document before sending and decrypt it at the far end

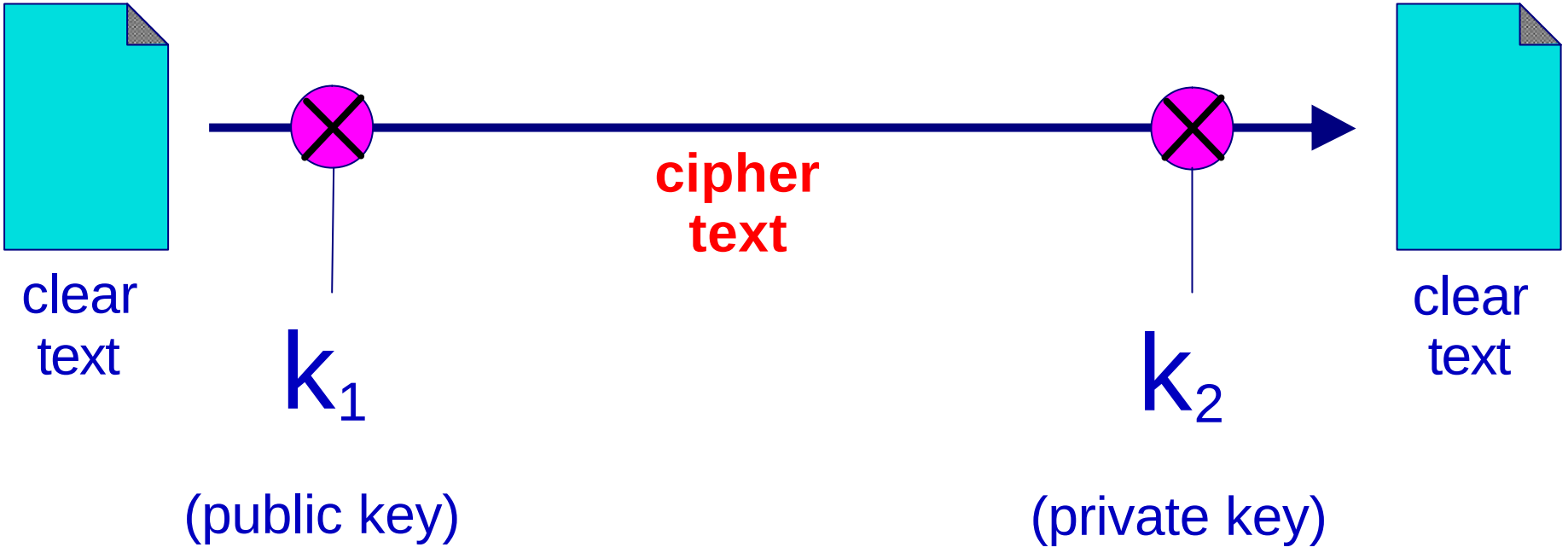
Features of Symmetric Ciphers

- Fast to encrypt and decrypt, suitable for large volumes of data
- A well-designed cipher is only subject to brute-force attack; the strength is therefore directly related to the key length
- Current recommendation is a key length of at least 90 bits
- i.e. to be fairly sure that your data will be safe for at least 20 years
- Problem - how do you distribute the keys?

Examples of Symmetric Ciphers

- DES - 56 bit key length, designed by US security service
- 3DES - effective key length 112 bits
- AES (Advanced Encryption Standard) - 128 to 256 bit key length
- Blowfish - 128 bits, optimized for fast operation on 32-bit microprocessors
- IDEA - 128 bits, patented (requires a license for commercial use)

"Public key" ciphers



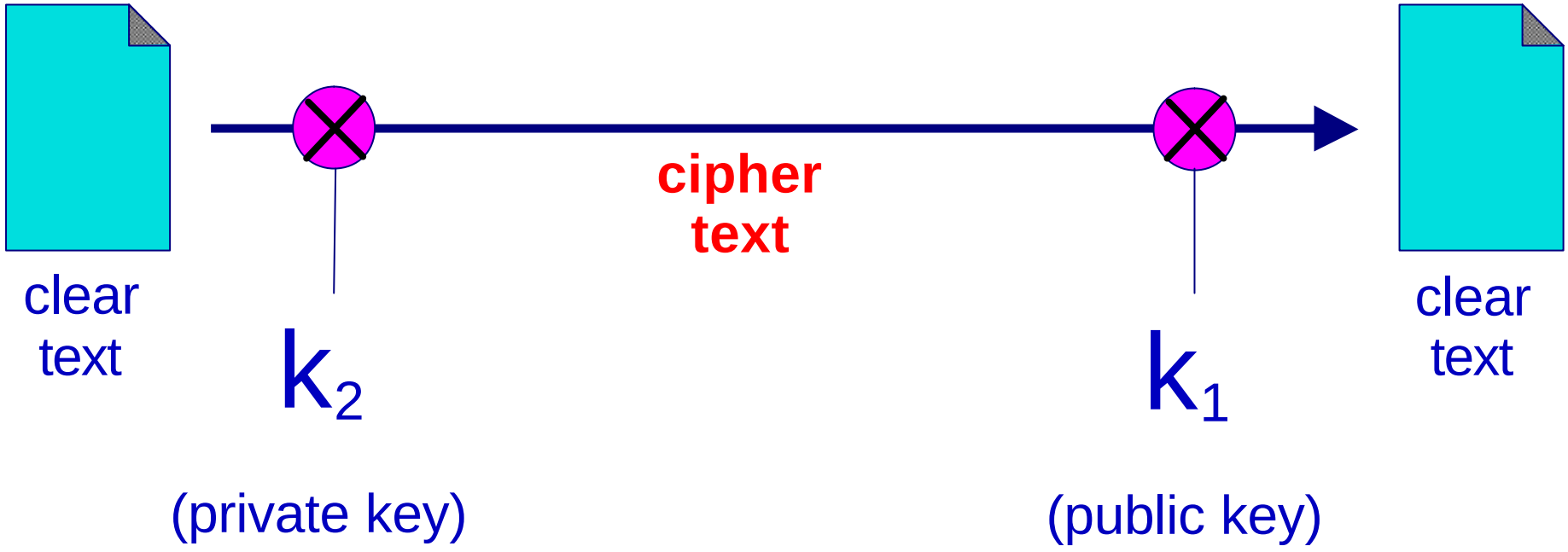
- One key is used to encrypt the document, a different key is used to decrypt it

Public key and Private key

- The Public key and Private key are mathematically related (generated as a pair)
- It is easy to convert the Private key into the Public key. It is not easy to do the reverse.
- Key distribution problem is solved: you can post your public key anywhere. People can use it to

- encrypt messages to you, but only the holder of the private key can decrypt them.
- Examples: RSA, Elgamal (DSA)

Use for authentication



- If you can decrypt the document with the public key, it proves it was written by the owner of the private key (and was not changed)

Protecting the private key

- The security of the private key is paramount: keep it safe!
- Keep it on a floppy or a smartcard?
- Prefer to keep it encrypted if on a hard drive
- That means you have to decrypt it (using a passphrase) each time you use it
- An attacker would need to steal the file containing the private key, AND know or guess the passphrase

Key lengths

- Attacks on public key systems involve mathematical attempts to convert the public key into the private key. This is more efficient than brute force.
- Recent developments suggest that 1024-bit keys might not be secure for long
- Recommend using 2048-bit keys

Cryptographic Applications

- At the data link layer
  PPP encryption

- At the network layer
  IPSEC

- At the transport layer
  TLS (SSL)

- At the application layer
  SSH, PGP/GPG

Secure Shell - SSH

Secure Shell - SSH

- program that allows secure network services over an insecure network, such as the Internet
- Internet protocol that allows a user to connect to a remote host via an encrypted link
- program to securely log into another computer over a network
- to execute commands safely in a remote machine
- to securely copy/move files from one machine to another
- is intended as a replacement for insecure "Berkeley services“:
  - telnet, rlogin, rsh, and rcp
- provides secure X connections over the network
- provide secure encrypted and authenticated communications between two hosts
- secure forwarding of arbitrary TCP connections/services

Secure Shell - SSH

- How do SSH, Telnet and Rlogin differ ?
  - SSH is a recently designed, high-security protocol
  - SSH uses strong cryptography to protect your connection against eavesdropping, hijacking and other attacks
  - Telnet and Rlogin are both older protocols offering minimal security
  - SSH and Rlogin both allow you to log in to the server without having to type a password
  - SSH allows you to connect to the server and automatically send a command
  - If you are connecting across the open Internet, then we recommend you use SSH
  - If you are behind a good firewall, it is more likely to be safe to use Telnet or Rlogin, but we still recommend you use SSH.

Secure Shell - SSH

- A typical SSH connection

Secure Shell - SSH

- Why should I use it ?
  - Traditional BSD 'r' - commands (rsh, rlogin, rcp) are vulnerable to different kinds of attacks
  - Somebody who has physical access to the wire, can gain unauthorized access to systems in a variety of ways
  - Protect against eavesdrop and logging of all the traffic to and from your system, including passwords
  - SSH offers strong host and user authentication methods
  - X Window System also has a number of severe vulnerabilities
  - Powerful guardian against the numerous security hazards that nowadays threaten network communications

Secure Shell - SSH

- What kinds of attacks does SSH protect against ?
  - Eavesdropping a transmission - looking for passwords, credit card numbers, or business secrets
  - Hijacking - taking over a communication and redirect
  - Interception of communication between two systems - inspect or modify any data being transmitted thru itself
  - Impersonation of a particular host - an intercepting system pretends to be the intended recipient
    - IP spoofing, or faking network addresses or routing information
    - fool access control mechanisms
    - redirect connections to a fake server

    All techniques cause information to be intercepted, possibly for hostile reasons

Secure Shell - SSH

- SSH – Secure Shell
  - **SSH never trusts the network !**
  - Can only force SSH to disconnect, but cannot decrypt or play back the traffic, or hijack the connection

- What kinds of attacks does SSH not protect against ?
  - Anything that compromises your host's security in some other way
  - Has gained root access to a machine, he can then subvert SSH
  - Has access to your home directory, then security is nonexistent

Secure Shell - SSH

- What software packages are available for implementing SSH?

  Client / Servers:

- OpenSSH – www.openssh.org - open source
- fressh – www.fressh.org - open source
- SSH Secure Shell – www.ssh.com - commercial
- F-secure – www.f-secure.com - commercial
- PuTTY – www.chiark.greenend.org.uk/~sgtatham/putty/ - free

OpenSSH

- Open Source Project
- Free Licensing
- Strong Encryption (3DES, Blowfish)
- X11 Forwarding (encrypt X Window System traffic)
- Port Forwarding (encrypted channels for legacy protocols)
- Strong Authentication (Public Key, One-Time Password)
- Agent Forwarding (Single-Sign-On)
- Interoperability (Compliance with SSH 1.3, 1.5, and 2.0 protocol Standards)
- SFTP client and server support in both SSH1 and SSH2 protocols.
- Kerberos and AFS Ticket Passing
- Data Compression

OpenSSH Server configuration

Install the **openssh-server** and **openssh** rpm packages

- `/usr/sbin/sshd` – SSH server daemon
- `/etc/ssh/sshd_config` – server configuration file
- The default config file is sufficient
- `/sbin/service sshd start` – to start the SSH service
- `/sbin/service sshd stop` – to stop the SSH service
- `/sbin/service sshd restart` – to restart the SSH service

- OpenSSH packages also require the `openssl` package, which installs several important cryptographic libraries that help OpenSSH provide encrypted communications

OpenSSH Server configuration file

Edit the default config file: # vi /etc/ssh/sshd_config

- `Port 22`
  - Specifies the port number that **sshd** listens on
- `Protocol 2,1`
  - Specifies the protocol versions **sshd** supports
- `ListenAddress 0.0.0.0`
  - Specifies the local addresses **sshd** should listen on
- `HostKey /etc/ssh/ssh_host_rsa_key`
  - Specifies a file containing a private host key used by SSH
- `LoginGraceTime 600`
  - Specifies the time limit for a user to log in
- `PermitRootLogin yes`
  - Specifies whether root can login using ssh
- `StrictModes yes`
  - Specifies whether **sshd** should check file modes and ownership of the user's files and home directory before accepting login

OpenSSH Server configuration file:

- `PubkeyAuthentication yes`
  - Specifies whether public key authentication is allowed
- `PasswordAuthentication yes`
  - Specifies whether password authentication is allowed
- `PermitEmptyPasswords no`
  - When password authentication is allowed, it specifies whether the server allows login to accounts with empty password strings
- `MaxStartups 10`
  - Specifies the maximum number of concurrent unauthenticated connections to the **sshd** daemon
- `KeepAlive yes`
  - Specifies whether the system should send TCP keepalive messages to the other side
- `VerifyReverseMapping no`
  - Specifies whether **sshd** should try to verify the remote host name
- `Subsystem sftp /usr/libexec/openssh/sftp-server`

OpenSSH Client configuration

- Install **openssh-clients** and **openssh** packages on the client
- Install the desired Windows SSH client software
- `/etc/ssh_config` - system wide ssh client configuration file
- `~/.ssh` - user custom configuration file

- Configuration data is parsed as follows:
  - 1. command line options
  - 2. user-specific file
  - 3. system-wide file

OpenSSH commands

- `ssh`
  - The basic rlogin/rsh-like client program
- `sshd`
  - The daemon that permits you to login
- `ssh-agent`
  - An authentication agent that can store private keys
- `ssh-add`
  - Tool which adds keys to in the above agent
- `sftp`
  - FTP-like program that works over SSH1 and SSH2 protocol
- `scp`
  - File copy program that acts like rcp
- `ssh-keygen`
  - Key generation tool
- `sftp-server`
  - SFTP server subsystem (started automatically by sshd)

OpenSSH Port Forwarding

- Secure otherwise insecure TCP/IP protocols via port forwarding
- the SSH server becomes an encrypted conduit to the SSH client
- mapping a local port on the client to a remote port on the server
- the mapped port numbers do not need to match for it to work

- To create a TCP/IP port forwarding channel which listens for connections on the localhost, use the following command:

```
ssh -L local-port:remote-host:remote-port
  user@remote-host
```

Note: Setting up port forwarding to listen on ports < 1024 requires root access

OpenSSH Port Forwarding – type I

To check your email on a server called mail.domain.com using POP through an encrypted SSH connection to the POP server, you can use the following command:

```
ssh -L 1100:mail.domain.com:110 mail.domain.com
```

Once the port forwarding channel is in place between the two machines, you can **direct** your **POP mail client** to use port **1100 on localhost** to check for new mail. Any requests sent to port 1100 on your system will be directed **securely** to the mail.domain.com server's port 110.

OpenSSH Port Forwarding – type II

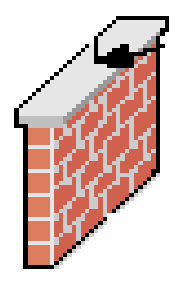
If mail.domain.com is not running an SSH server daemon, you can log in via SSH to a machine on the same network and still use SSH to secure a part of the POP connection.

```
ssh -L 1100:mail.domain.com:110 ssh-server.domain.com
```

You are forwarding your POP request from port 1100 on your machine through SSH connection on port 22 to ssh server.domain.com. Then, ssh-server.domain.com connects to port 110 on mail.domain.com to allow you to check for new mail. Only the connection between your system and ssh-server.domain.com is secure.

OpenSSH Lab

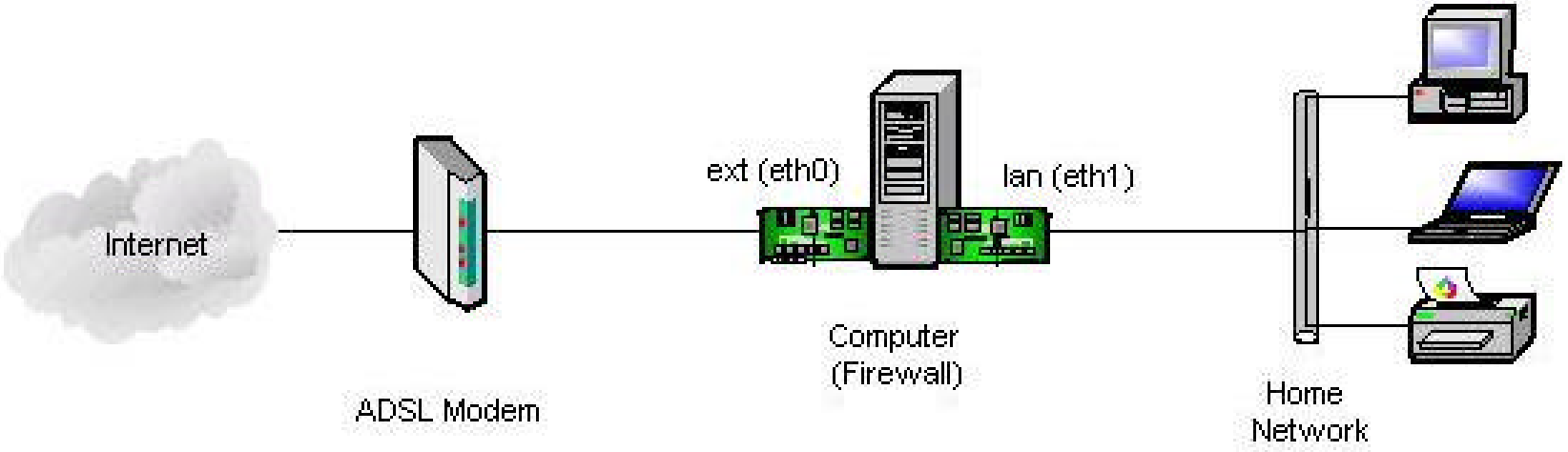
- OpenSSH server configuration
- Edit/view the sshd_config file
- Managing the sshd service
- OpenSSH Client configuration
- Using the ssh command with password authentication
- Using the ssh command with public key authentication
- Using the scp command
- Using the sftp command
- Port forwarding



Firewalls

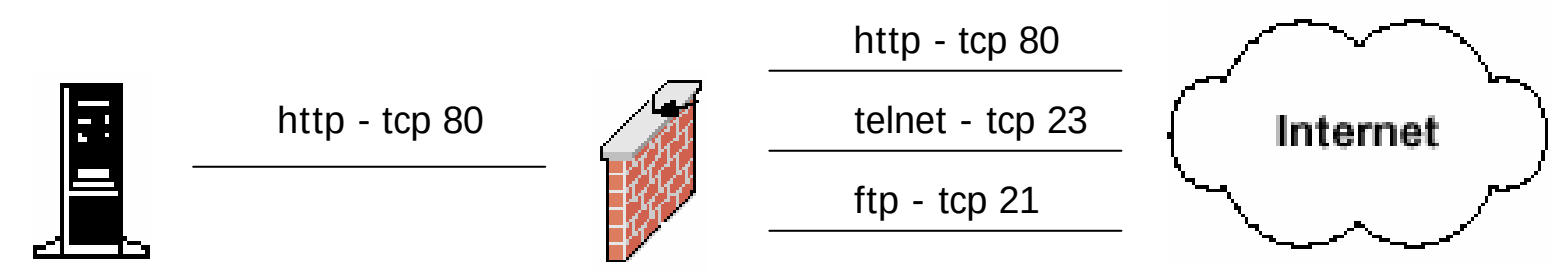
- Many people might think that a firewall is a single device on your network configured to protect your internal network from the external world
- A firewall is a system (or a group of systems) that enforces an access control policy between two networks
- Disallow unauthorized and/or malicious traffic from traveling on your network – in both directions
- Firewalls can't protect you from attacks that don't go through it
- If there's another entry point to your network not protected by a firewall, then your network isn't secured
- Firewalls do not verify the content of the traffic through it

A typical firewall setup



Packet filtering firewalls

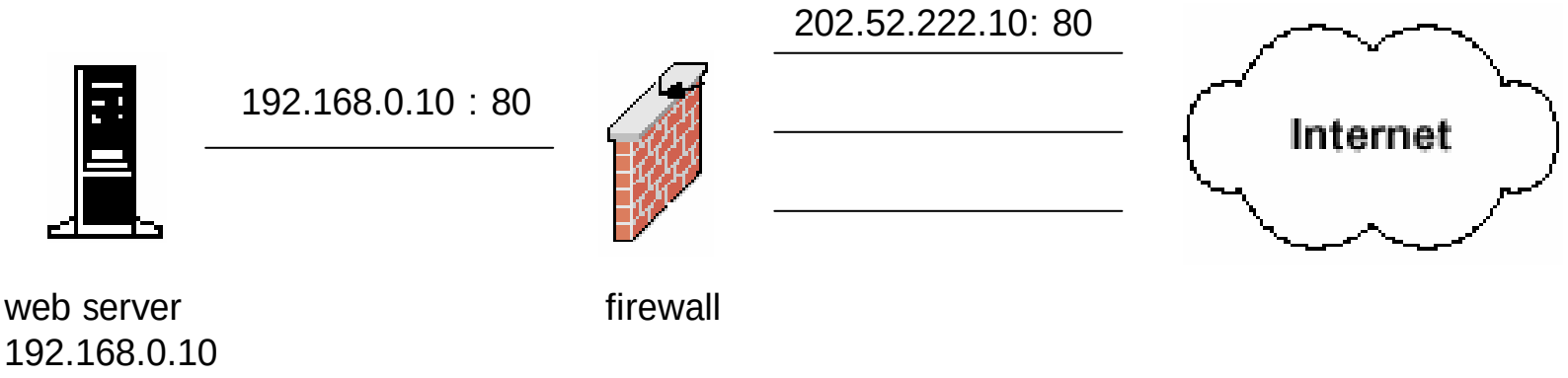
- examines the source and destination address of the data packet and either allows or denies the packet from traveling the network
- blocks access through the firewall to any packets, which try to access ports which have been declared "off-limits"



- Allow only http - tcp 80
- Drop ip any

Application layer firewalls

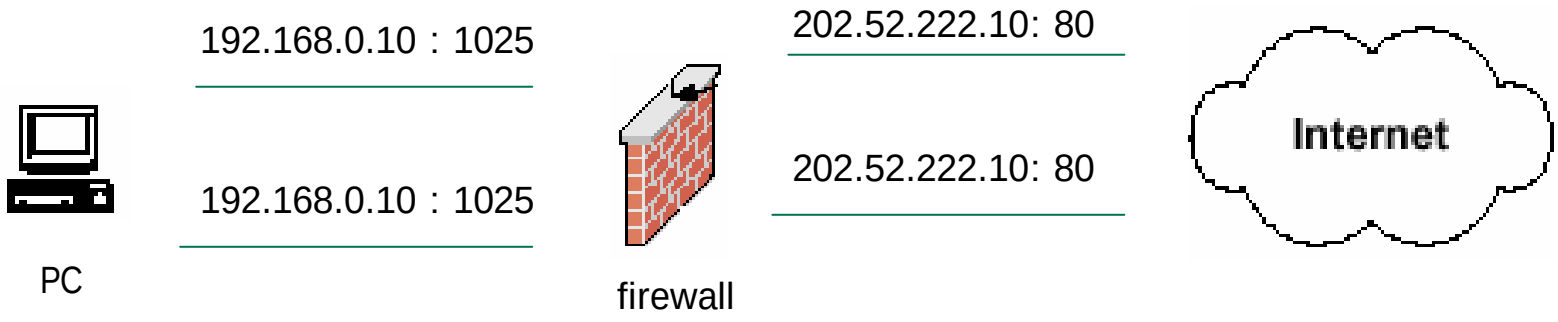
- Also known proxy firewalls, application gateway
- attempts to hide the configuration of the network behind the firewall by acting on behalf of that network/servers
- All requests for access are translated at the firewall so that all packets are sent to and from the firewall, rather than from the hosts behind the firewall



- Translates 202.52.222.10 : 80 to 192.168.0.10 : 80

Stateful inspection firewalls

- Examines the state and the context of the packets
- Remembers what outgoing requests have been sent and only allow responses to those requests back through the firewall
- Attempts to access the internal network that have not been requested by the internal network will be denied



- Only allows reply packets for requests made out
- Blocks other unregistered traffic

Firewall Best Practices

- Explicitly deny all traffic except for what you want
- The default policy should be that if the firewall doesn't know what to do with the packet, deny/drop it
- Don't rely only on your firewall for the protection of your network - remember that it's only a device, and devices do fail
- Make sure you implement what's called "defense in depth." - multiple layers of network protection
- Make sure all of the network traffic passes through the firewall
- If the firewall becomes disabled, then disable all communication

Firewall Best Practices

- If there's another way in to the network (like a modem pool or a maintenance network connection), then this connection could be used to enter the network completely bypassing the firewall protection

- Disable or uninstall any unnecessary services and software on the firewall - limit the number of applications

- Use firewalls internally to segment networks between different departments and permit access control based upon business needs

Firewall products

– Iptables	www.iptables.org
– Cisco PIX	www.cisco.com
– Checkpoint	www.checkpoint.com
– Border Manager	www.novell.com
– Netscreen	www.netscreen.com
– Winroute	www.winroute.com

IPTables

- Features:
  - Linux kernel contains advanced tools for packet filtering
  - the framework inside the Linux 2.4.x kernel
  - re-designed and heavily improved successor of the previous 2.2.x ipchains and 2.0.x ipfwadm systems
  - Provides functionality of packet filtering (stateless or stateful)
  - All kinds of network address translation (NAT)
  - Packet mangling (manipulation)
  - flexible and extensible infrastructure
  - Large number of additional features as modules / patches
  - generic table structure for the definition of rulesets
  - consists of classifiers (matches) and one connected action (target)

What all can I do with iptables ?

- Build internet firewalls based on stateless and stateful packet filtering
- Use NAT and masquerading for sharing internet access where you don't have enough addresses
- Use NAT for implementing transparent proxies
- Aid the tc+iproute2 system used to build sophisticated QoS routers
- Do further packet manipulation (mangling) like altering the TOS field of the IP header

So What's A Packet Filter?

- the process of controlling network packets as they attempt to enter, move through, and exit your system
- looks at the *header* of packets as they pass through, and decides the fate of the entire packet

  - **DROP** the packet - discard the packet as if it had never received it
  - **ACCEPT** the packet - let the packet go through
  - **LOG** the packet – just log the information for monitoring purpose

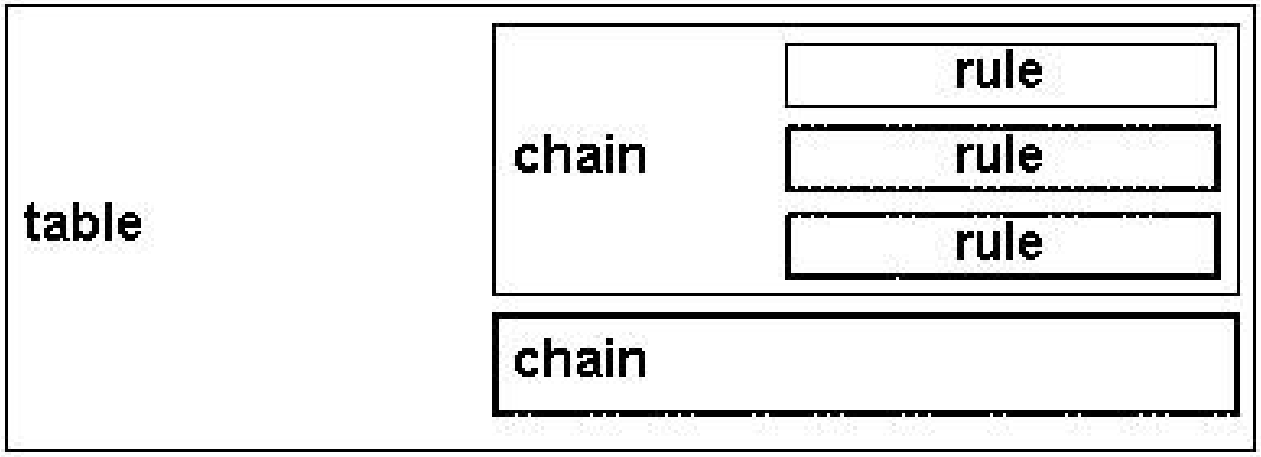
- or something more complicated!

- Under Linux, packet filtering is built into the kernel

Why Would I Want to Packet Filter?

- Control – allow certain types of traffic, and disallow others
- Security – prevent unauthorized access or attacks to/from your network
- Watchfulness - monitor abnormal / suspicious activity to/from your network

IPTables components



- Rule – operation to be performed on a packet
- Chains – collection of rules
- Table – collection of chains

- Iptables – userspace command for configuring the system
- Modules – kernel modules for diff. features / tasks

IPTables “tables”

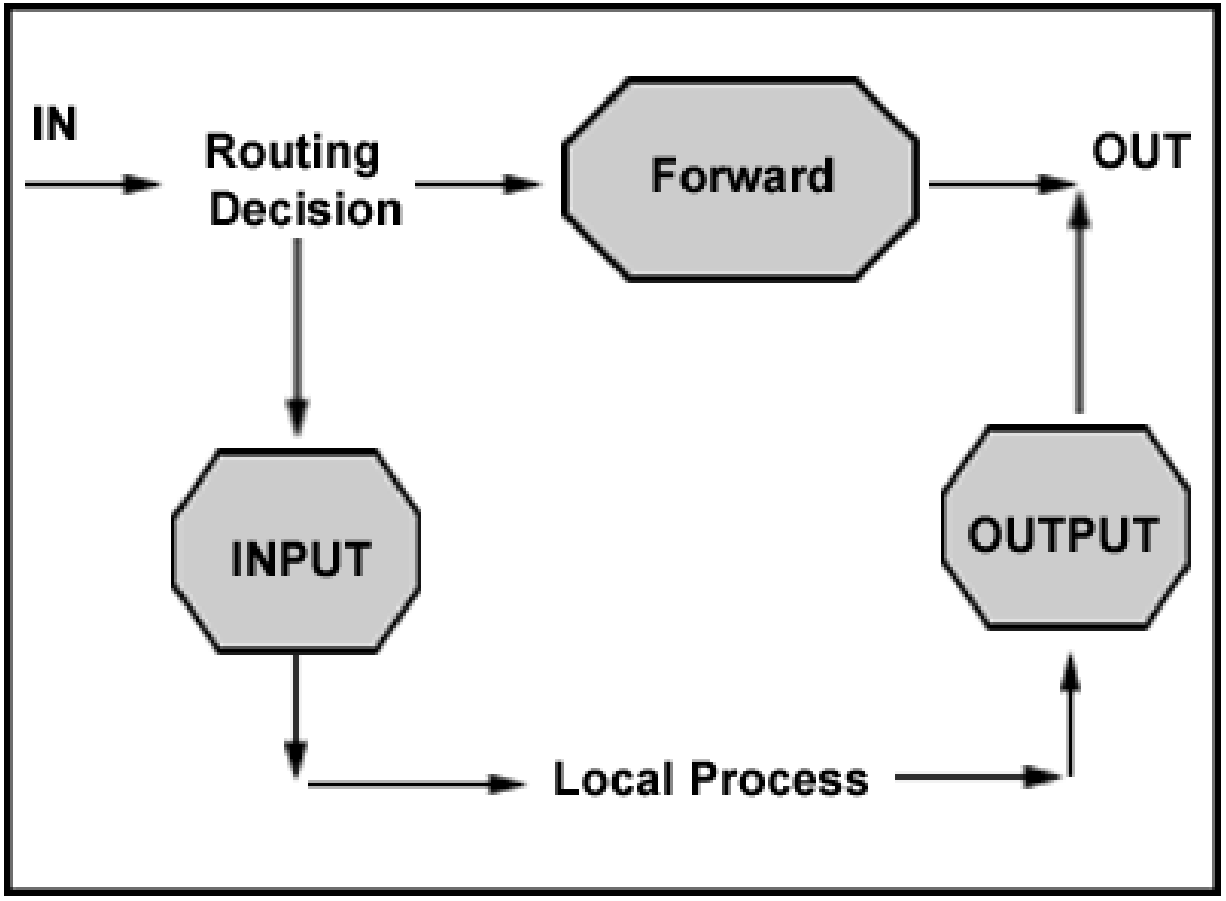
IPTables has three tables:

- Filter – performs packet filtering
- NAT – performs address translation between hosts on internal network and external addresses on the internet
- Mangle - modifies contents of specific packet header fields

Filter table “chains”

- INPUT – packets destined for a local interface
- FORWARD – routable packets to another network
- OUTPUT – packets originating from a local interface

How packets traverse the filter table chains?



IPTables configuration

To manage chains in a table:

- Creating a new chain -N
- Delete an empty chain -X
- Flush all rules in a chain -F
- Change the default policy of a chain -P
- List all the rules in a chain -L

To manage rules in a chain:

- Append a rule in a chain -A
- Insert a rule at some position -I
- Replace a rule at some position -R
- Delete a rule -D

IPTables configuration

Possible targets for a rule in a filter table chain:

ACCEPT	– to accept & let the packet pass through
DROP	– to simply drop the packet w/o any error message
REJECT	– to deny the packet w/ an error message (polite)
LOG	– to log info of the packet to syslog for monitoring

IPTables configuration

Default chain policies:

```
INPUT ACCEPT any any
FORWARD ACCEPT any any
OUTPUT ACCEPT any any
```

Change the default policy to DROP all packets:

```
iptables –P INPUT DROP
```

IPTables configuration

```
iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP

iptables -D INPUT 1
iptables -D INPUT -s 127.0.0.1 -p icmp -j DROP

iptables -A INPUT -s 0/0 -j DROP
iptables -A INPUT -j DROP
```

To get help:

```
iptables –h
iptables -p tcp –h
iptables -m state –h
iptables –j ACCEPT -h
```

IPTables configuration

```
iptables –A INPUT –p tcp –dport 25 –j ACCEPT

iptables –A OUTPUT –p tcp dport 25 –j ACCEPT

iptables –A FORWARD –i eth0 –o eth1 –p tcp –dport
25 –j ACCEPT

iptables –A FORWARD –s 202.52.225.0/24 –d
  202.52.255.1 –p tcp –dport 25 –j ACCEPT

iptables –A FORWARD –s 202.52.225.0/24 –d
  202.52.255.5 –p udp –dport 53 –j ACCEPT

iptables –A FORWARD –d 202.52.255.5 –p tcp –j LOG
  –log-prefix “TCP log ”
```

IPTables - Connection Tracking

- Stateful connection tracking of traversing packets

NEW - packets that create a NEW connection

ESTABLISHED - packets belonging to an existing connection

RELATED - packets related to an existing connection

INVALID - packets not corresponding to any existing connection

```
iptables –A FORWARD –d 202.52.225.0/24 –p tcp –m
  –state --state ESTABLISHED –j REJECT

iptables –A FORWARD –m –state --state INVALID –j
  DROP
```

NAT with IPtables

- NAT is a standard that enables a network to use one set of IP addresses for moving data packets on the local area network and a second set of IP addresses for external traffic the Internet

- The firewall acts as the address translation device between addresses on the home side of the network and addresses on the internet side of the network

- NAT enables the user to shield address on the internal network from address on the Internet network

NAT table “chains”

- PREROUTING – before routing decision is made and packet enters the system
- OUTPUT – packets leaving the system
- POSTROUTING – after routing decision is made and packet leaves the system

NAT chain “targets”

- DNAT - mainly used in cases where you have a public IP and want to redirect accesses to the firewall to some other host

- SNAT - mainly used for changing the source address of packets

- MASQUERADE - used in exactly the same way as SNAT, but the MASQUERADE target automatically checks for the IP address to use, instead of doing as the SNAT target does - just using the single configured IP address.

NAT chain rules

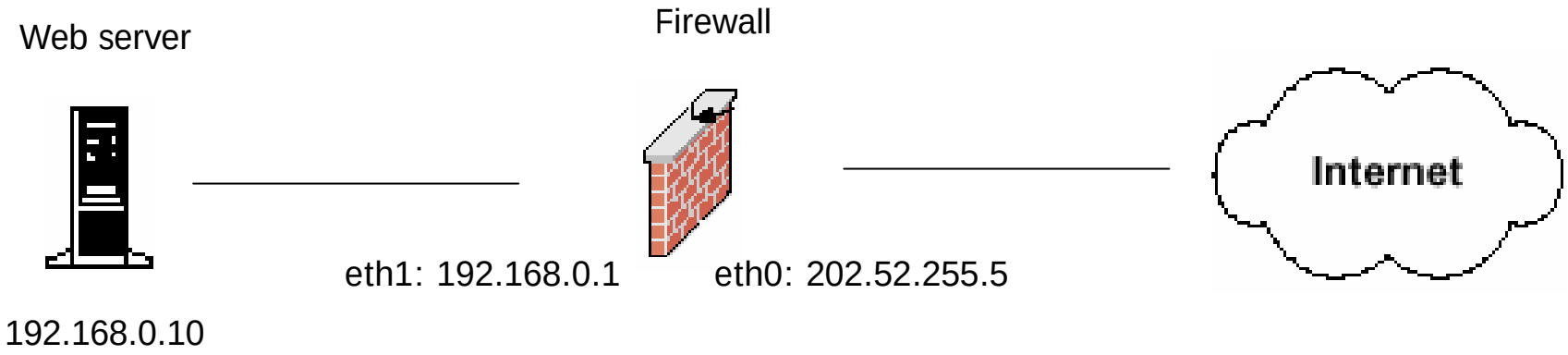
To NAT all outbound traffic with the IP of eth0:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Mapping the web port with squid:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp
–-dport 80 -j DNAT --to 202.52.202.52:3128
```

NAT: Fixed IP mapping (inbound)

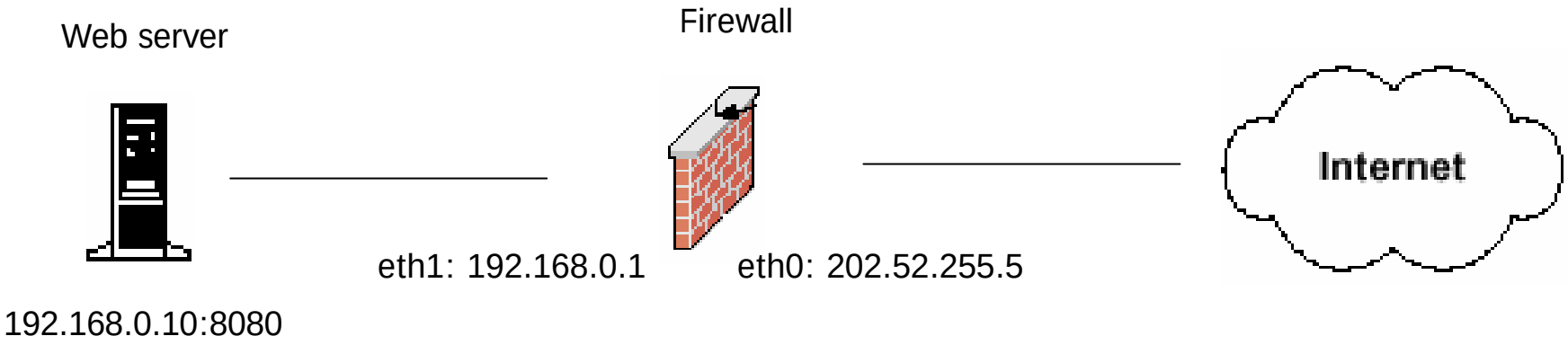


Makes it possible to hide internal server IP from the Internet

```
iptables -t -nat -A PREROUTING -i eth1 -d 202.52.255.5
-j DNAT –to-destination 192.168.0.1
```

This rule maps the IP addresses in both the requests sent to the server and the server's reply

NAT: Port mapping (inbound)

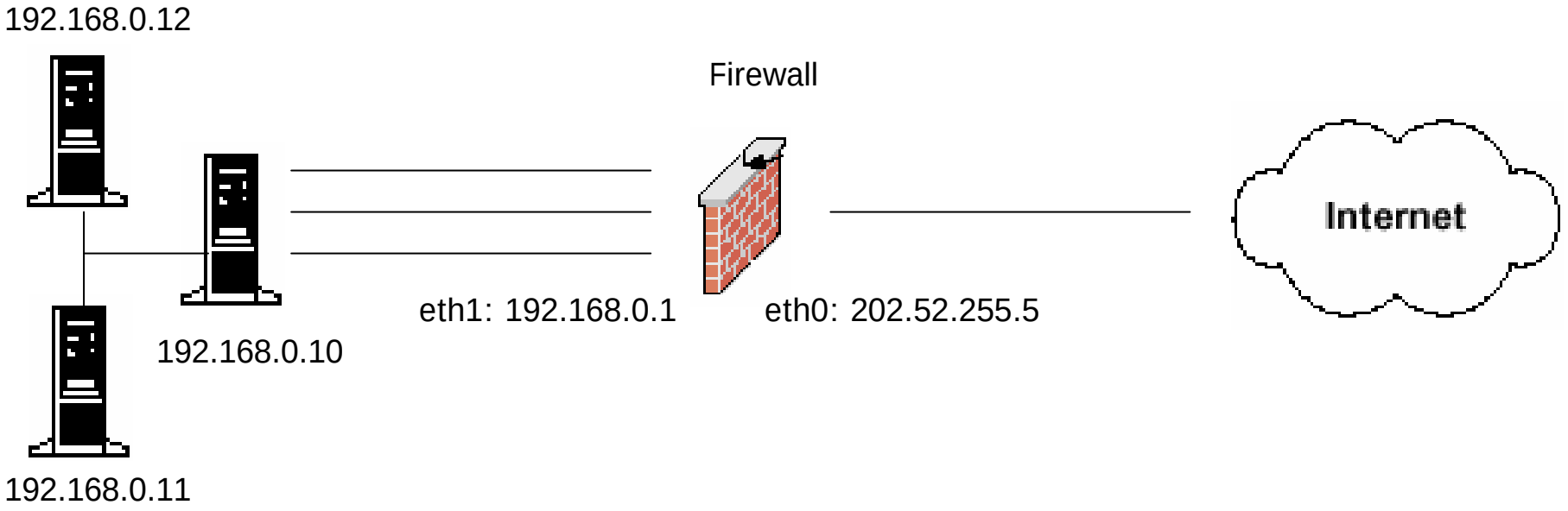


Entails the modification of the destination port and enables clients to access a service via destination port other than that on which service listens.

```
iptables -t -nat -A PREROUTING -i eth0 -d 202.52.255.5 -p tcp
-m tcp –dport 80 -j DNAT –to-destination 192.168.0.1:8080
```

This rule maps port 80 of host with IP 202.52.255.5 to port 8080 of the internal host having IP 192.168.14.2

NAT: IP Masquerading (outbound)

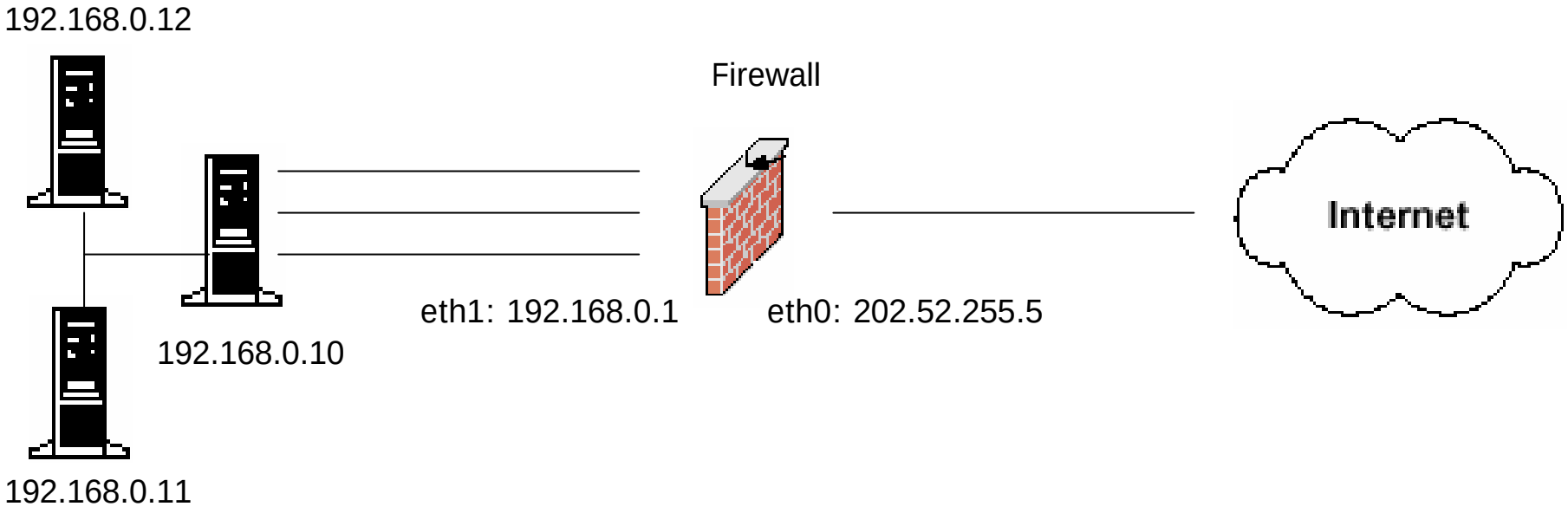


Outbound packets receive the IP of the output interface as their source address. It is useful when there is no fixed IP addresses of output interface.

```
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

This rule translates the source IP of all outbound packets to 202.52.255.5, the IP of eth0

NAT: SNAT (outbound)

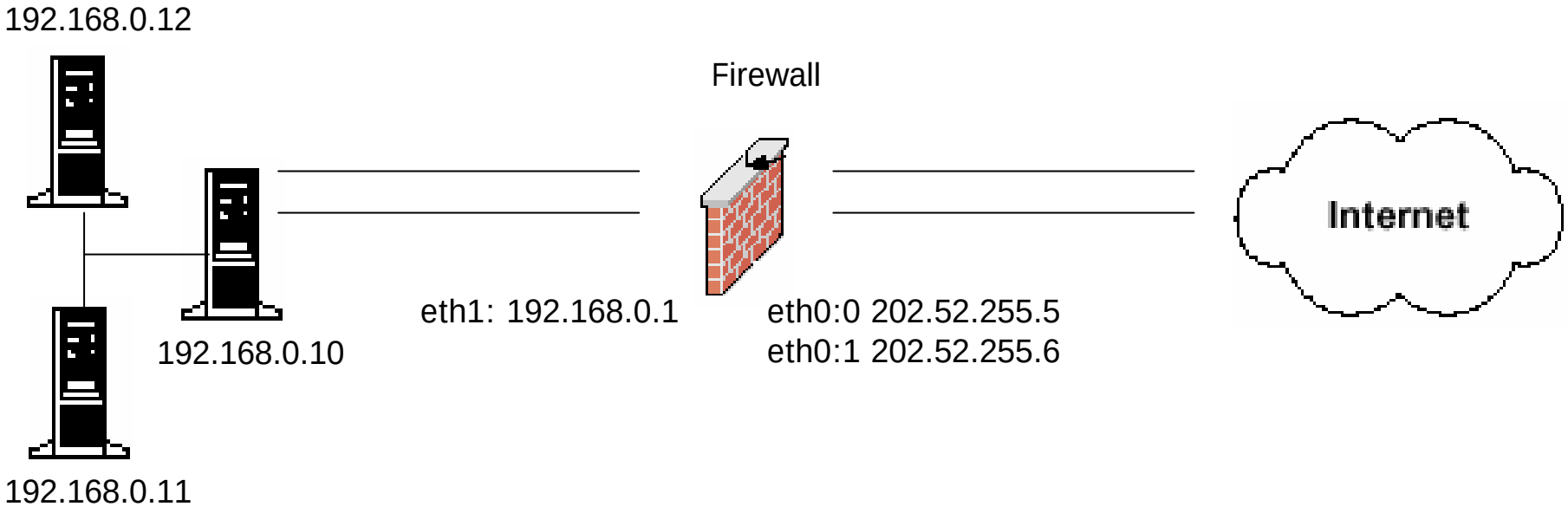


Another way to sharing a single public IP by all private hosts is to SNAT (Source NAT) it.

```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.0/24
-j SNAT –to-source 202.52.255.5
```

The source IP of all outbound packets will be converted to 202.52.255.5

NAT: Fixed IP mapping (outbound)



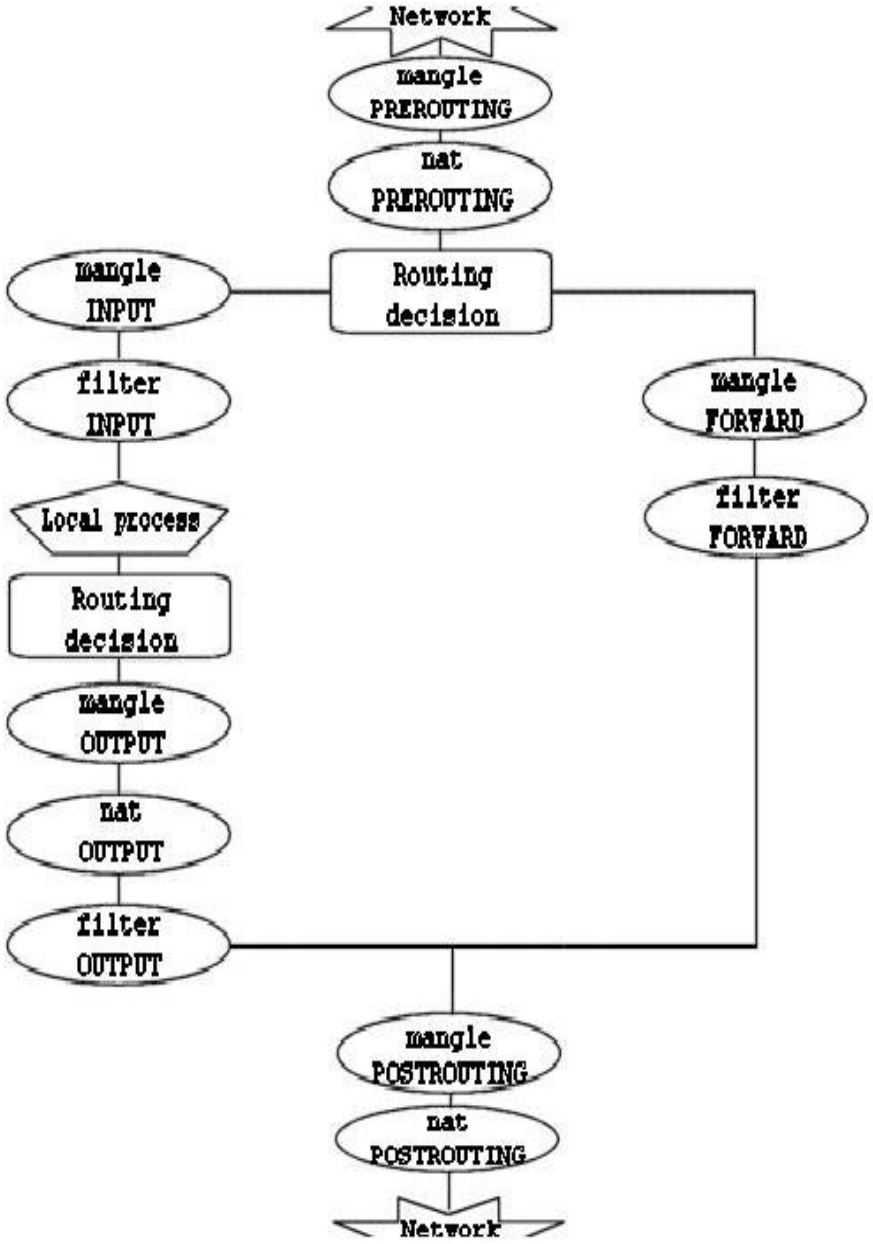

```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.11
-j SNAT –to-source 202.52.255.5

iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.12
-j SNAT –to-source 202.52.255.6
```

The source IP of 192.168.10.11 will be converted to 202.52.255.5

The source IP of 192.168.10.12 will be converted to 202.52.255.6

How packets flow thru IPtables?



IPTables Lab

- Installation
- Using Iptables
  - To view all iptables command line options
  - To list all current default rules/chains
  - To set the default policies for all chains
  - To allow ping to work to/from your host to anywhere
  - To allow ping to work only across the firewall but not to/from it
  - To allow all internal users to access websites on the Internet
  - To allow some external users access to SSH, SMTP, POP, HTTP, DNS servers in your internal network
  - To save all rules/chains to /etc/sysconfig/iptables to make permanent
  - To manage iptables service
  - NAT configuration

Intrusion Detection System

What is IDS?

- Intrusion Detection System (IDS) analyzes IP packets looking for known patterns of intrusion in real time. Intrusion detection system can give valuable information on unauthorized access to your network.

- Intrusion Detection system can be Host based or Network Based. In host based intrusion detection system , it only watches for packets coming into a Single Host. The Host based IDS doesn't listen on interfaces in promiscuous mode. Programs that parses the system log files for security related issues can also be termed as host based IDS. These programs will generally report incidents that seems suspicious and alert the administrator.

- Examples of Host based Intrusion detection system are : portsentry,Logsentry, Logwatch,chkrootkit.

What is IDS? ..contd

- Network Based IDS analyzes all IP packets coming into the network. Network based IDS are generally deployed with port mirroring facility provided by most managed switches. Such switches are configured to copy all data on certain port/ports and send it the port where NIDS is connected. NIDS generally listens on promiscuous mode and analyze all the data it receives. Snort is one of the most popular Network based IDS, and is also open source.

Host Base IDS

- Portsentry
  - Portsentry is Host based IDS system that is highly customizable.
  - Portsentry has number of ways to detect port scan and when it detects it can log to syslog or add "ALL: offending host" in /etc/hosts.deny, or reject the route with "route add -host offending_host reject" or block it with dynamic firewall rules using ipchains or iptables.
  - You can download portsentry from http://sourceforge.net/projects/sentrytools/

Host Based IDS - Portsentry

- The installation of Portsentry is fairly simple.

```
$ tar xzvf portsentry-1.2.tar.gz
$ cd portsentry_beta

$ edit portsentry_config.h
$ vi portsentry_config.h
```

- **Change the following lines**

```
#define CONFIG_FILE "/usr/local/psionic/portsentry/portsentry.conf"
to
#define CONFIG_FILE "/etc/portsentry/portsenty.conf"

#define SYSLOG_FACILITY LOG_DAEMON
to
#define SYSLOG_FACILITY LOG_LOCAL0
```

- Ofcourse you can also install using the defaults.

Host Based IDS - Portsentry

- **Edit portsentry.conf**

```
$ vi portsentry.conf
```

- It is better to leave the TCP_PORTS, UDP_PORTS, ADVANCED_PORTS_TCP, ADVANCED_PORTS_UDP untouched. The defaults are quite secure.
- However you should remove ports on which legitimate daemons are listening from TCP_PORTS,UDP_PORTS. For eg. If you are running IMAP server then you should remove port 143 from TCP_PORTS. Portsentry will bind on the ports listed in TCP_PORTS and UDP_PORTS.

- ADVANCED_PORTS_TCP and ADVANCED_PORTS_UDP instructs portsentry to run advanced stealth detection mode below the defined ports. It should also be noted that ports listed in TCP_PORTS and UDP_PORTS are ignored by Advance stealth detection.

- You may need to edit the two options ADVANCED_EXCLUDE_TCP, ADVANCED_EXCLUDE_UDP to add ports used by other services running on the host.

- For eg.
- You must add port 22 to ADVANCED_PORTS_TCP if you are running ssh server.

Host Based IDS - Portsentry

- **Change the following lines**

```
IGNORE_FILE="/usr/local/psionic/portsentry/portsentry.ignore"
to
IGNORE_FILE="/etc/portsentry/portsentry.ignore"


HISTORY_FILE="/usr/local/psionic/portsentry/portsentry.history"
to
HISTORY_FILE="/var/log/portsentry/portsentry.history"


BLOCKED_FILE="/usr/local/psionic/portsentry/portsentry.blocked"
to
BLOCKED_FILE="/var/log/portsentry/portsentry.blocked"


You can also disble DNS resolution by changing
RESOLVE_HOST = "1"
to
RESOLVE_HOST = "0"
```

Host Based IDS - Portsentry

- **Uncomment the following line to add a drop route**

```
KILL_ROUTE="/sbin/route add -host $TARGET$ reject“
```

OR this one to block the host via firewall

```
KILL_ROUTE="/sbin/iptables -I INPUT -s $TARGET$ -j DROP"
```

- You don't need to use TCP wrapper options if you have "ALL:ALL" in “/etc/hosts.deny”.

- Next open portsentry.ignore files to add hosts that needs to ignored.

```
$ vi portsentry.ignore
```

- Don't remove the two lines already present:

```
127.0.0.1/32
0.0.0.0
```

Host Based IDS - Portsentry

- **Edit the Makefile**

```
$ vi Makefile
```

- **Change**

```
CC = cc
to
CC = gcc
```

- **Change**

```
INSTALLDIR = /usr/local/psionic
to
INSTALLDIR = /etc
```

- **Now compile**

```
$ make linux
```

- **Note:** You may fail to compile if you use new versions of “gcc”, as the newer versions has removed support for Multi-line string literals.

- **Workaround**

```
$ vi portsentry.c
```

Host Based IDS - Portsentry

- **Add the following line to syslog.conf**

```
# vi /etc/syslog.conf
local0.*     /var/log/portsentry/portsentry.log
```

- You may also add “local0.none” in “/var/log/messages” to stop filling your messages file.

- **Restart syslogd**

```
# service syslog restart
```

- **Configure Logrotate**

```
# vi /etc/logrotate.d/syslog
```

- **Add**

```
/var/log/portsentry/portsentry.log
```

cat /etc/logrotate.d/syslog

```
/var/log/portsentry/portsentry.log /var/log/kern.log
   /var/log/messages /var/log/secure /var/log/maillog
   /var/log/spooler /var/log/boot.log /var/log/cron {
     sharedscripts
```

Host Based IDS - Portsentry

- **Add *logrotate* for logs generated by *portsentry***

```
# cd /etc/logrotated.d
# vi portsentry
```

- **And add the following**

```
/var/log/portsentry/portsentry.history {
   weekly
   postrotate
         /usr/bin/killall -HUP portsentry
   endscript
}
```

- You can have portsentry run on different modes:

```
portsentry -tcp     (basic port-bound TCP mode)
portsentry -udp     (basic port-bound UDP mode)
portsentry -stcp        (Stealth TCP scan detection)
portsentry -atcp        (Advanced TCP stealth scan detection)
portsentry -sudp        ("Stealth" UDP scan detection)
portsentry -audp        (Advanced "Stealth" UDP scan detection)
```

- Only one mode can be started at a time.

Host Based IDS - Portsentry

- **-tcp - Basic port-bound TCP mode**
  - PortSentry will check the config files and then bind to all the TCP ports in the background.
- **-udp - Basic port-bound UDP mode**
  - PortSentry will check the config files and then bind to all the UDP ports in the background.
- **-stcp - Stealth TCP scan detection mode**
  - PortSentry will use a raw socket to monitor all incoming packets. If an incoming packet is destined for a monitored port it will react to block the host. This method will detect connect() scans, SYN/half-open scans, and FIN scans.
- **-sudp - "Stealth" UDP scan detection mode**
  - This is same as above but for UDP ports. This does not bind any sockets.

Host Based IDS - Portsentry

- **-atcp - Advanced TCP stealth scan detection mode**

  PortSentry will start by making a list of all the ports listening in the port
  area under the ADVANCED_PORTS_TCP option and will then create an exclusion list based on these ports. Any host connecting to *any port* in this range that is *not excluded* is blocked

- **-audp - Advanced UDP "stealth" scan detection mode**

  This is the same as above except for the UDP protocol.

Host Based IDS - Portsentry

- **Configuration Files** (/etc/portsentry**)**
  - *–portsentry.conf*
  - *–portsentry.ignore*

- **Log Files (/var/log/portsentry)**
  - *–portsentry.blocked.atcp*
  - *–portsentry.blocked.audp*
  - *–portsentry.blocked.stcp*
  - *–portsentry.blocked.sudp*
  - *–portsentry.blocked.tcp*
  - *–portsentry.blocked.udp*
  - *–portsentry.history Blocked history*

- ***Starting up***

```
# /etc/portsentry/portsentry -tcp
# /etc/portsentry/portsentry -udp
# /etc/portsentry/portsentry -atcp
# /etc/portsentry/portsentry -audp
# /etc/portsentry/portsentry -stcp
```

Host Based IDS - chrootkit

- chkrootkit is tool written to detect known Trojans and root kits installed in a system. It can also detect if your Ethernet interface is in promiscuous mode or if your lastlog entries are modified.

- You can download a recent version from www.chkrootkit.org.

Host Based IDS - chrootkit

- **Output of chkrootkit:**

```
# ./chkrootkit
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not infected
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'... not tested
Checking `inetdconf'... not found
Checking `identd'... not found
Checking `init'... not infected
Checking `killall'... not infected
Checking `ldsopreload'... not infected
Checking `login'... not infected
Checking `ls'... not infected
```

- ….

Host Based IDS - chrootkit

```
Checking `lsof'... not infected
Checking `mail'... not infected
Checking `mingetty'... not infected
Checking `netstat'... not infected
Checking `named'... not found
Checking `passwd'... not infected
Checking `pidof'... not infected
Checking `pop2'... not found
Checking `pop3'... not found
Checking `ps'... not infected
Checking `pstree'... not infected
Checking `rpcinfo'... not infected
Checking `rlogind'... not found
Checking `rshd'... not found
Checking `slogin'... not infected
Checking `sendmail'... not infected
Checking `sshd'... not infected
Checking `syslogd'... not infected
Checking `tar'... not infected
Checking `tcpd'... not infected
Checking `tcpdump'... not infected
Checking `top'... not infected
Checking `telnetd'... not found
Checking `timed'... not found
Checking `traceroute'... not infected
Checking `vdir'... not infected
Checking `w'... not infected
Checking `write'... not infected
```

Host Based IDS - chrootkit

```
Searching for sniffer's logs, it may take a while... nothing found
Searching for HiDrootkit's default dir... nothing found
Searching for t0rn's default files and dirs... nothing found
Searching for t0rn's v8 defaults... nothing found
Searching for Lion Worm default files and dirs... nothing found
Searching for RSHA's default files and dir... nothing found
Searching for RH-Sharpe's default files... nothing found
Searching for Ambient's rootkit (ark) default files and dirs... nothing found
Searching for suspicious files and dirs, it may take a while...
/usr/lib/perl5/5.8.0/i386-linux-thread-multi/.packlist /usr/lib/qt-3.0.5/etc/settings/.qtrc.lock
    /usr/lib/qt-3.0.5/etc/settings/.qt_plugins_3.0rc.lock /usr/lib/qt-3.1/etc/settings/.qtrc.lock
    /usr/lib/qt-3.1/etc/settings/.qt_plugins_3.1rc.lock
    /usr/lib/openoffice/share/gnome/net/.directory /usr/lib/openoffice/share/gnome/net/.order
    /usr/lib/openoffice/share/kde/net/applnk/OpenOffice.org/.directory
    /usr/lib/openoffice/share/kde/net/applnk/OpenOffice.org/.order
Searching for LPD Worm files and dirs... nothing found
Searching for Ramen Worm files and dirs... nothing found
Searching for Maniac files and dirs... nothing found
Searching for RK17 files and dirs... nothing found
Searching for Ducoci rootkit... nothing found
Searching for Adore Worm... nothing found
Searching for ShitC Worm... nothing found
Searching for Omega Worm... nothing found
Searching for Sadmind/IIS Worm... nothing found
Searching for MonKit... nothing found
Searching for Showtee... nothing found
Searching for OpticKit... nothing found
Searching for T.R.K... nothing found
Searching for Mithra... nothing found
Searching for LOC rootkit ... nothing found
Searching for Romanian rootkit ... nothing found
Searching for HKRK rootkit ... nothing found
```

Host Based IDS - chrootkit

```
Searching for Suckit rootkit ... nothing found
Searching for Volc rootkit ... nothing found
Searching for Gold2 rootkit ... nothing found
Searching for TC2 Worm default files and dirs... nothing found
Searching for Anonoying rootkit default files and dirs... nothing
   found
Searching for ZK rootkit default files and dirs... nothing found
Searching for ShKit rootkit default files and dirs... nothing found
Searching for anomalies in shell history files... nothing found
Checking `asp'... not infected
Checking `bindshell'... not infected
Checking `lkm'... nothing detected
Checking `rexedcs'... not found
Checking `sniffer'... Checking `w55808'... not infected
Checking `wted'... nothing deleted
Checking `scalper'... Warning: Possible Scalper Worm installed
Checking `slapper'... not infected
Checking `z2'... nothing deleted
```

Questions & Answers

Snort

- **Network Intrusion Detection System (NIDS)**
- Inspects/sniffs all network traffic passing thru it for any abnormal content
- Provides a layer of defense which monitors network traffic for predefined suspicious activity or patterns
- Has built in signature-base and anomaly detection, providing the capability to look for set "patterns" in packets
- String search signature (i.e. look for confidential), logging and TCP reset features
- Provides worthwhile information about malicious network traffic
- Help identify the source of the incoming probes, scans or attacks
- Alert sys admins when potential hostile traffic is detected
- Similar to a security "camera" or a "burglar alarm“
- Alerts security personnel that a Network Invasion maybe in progress

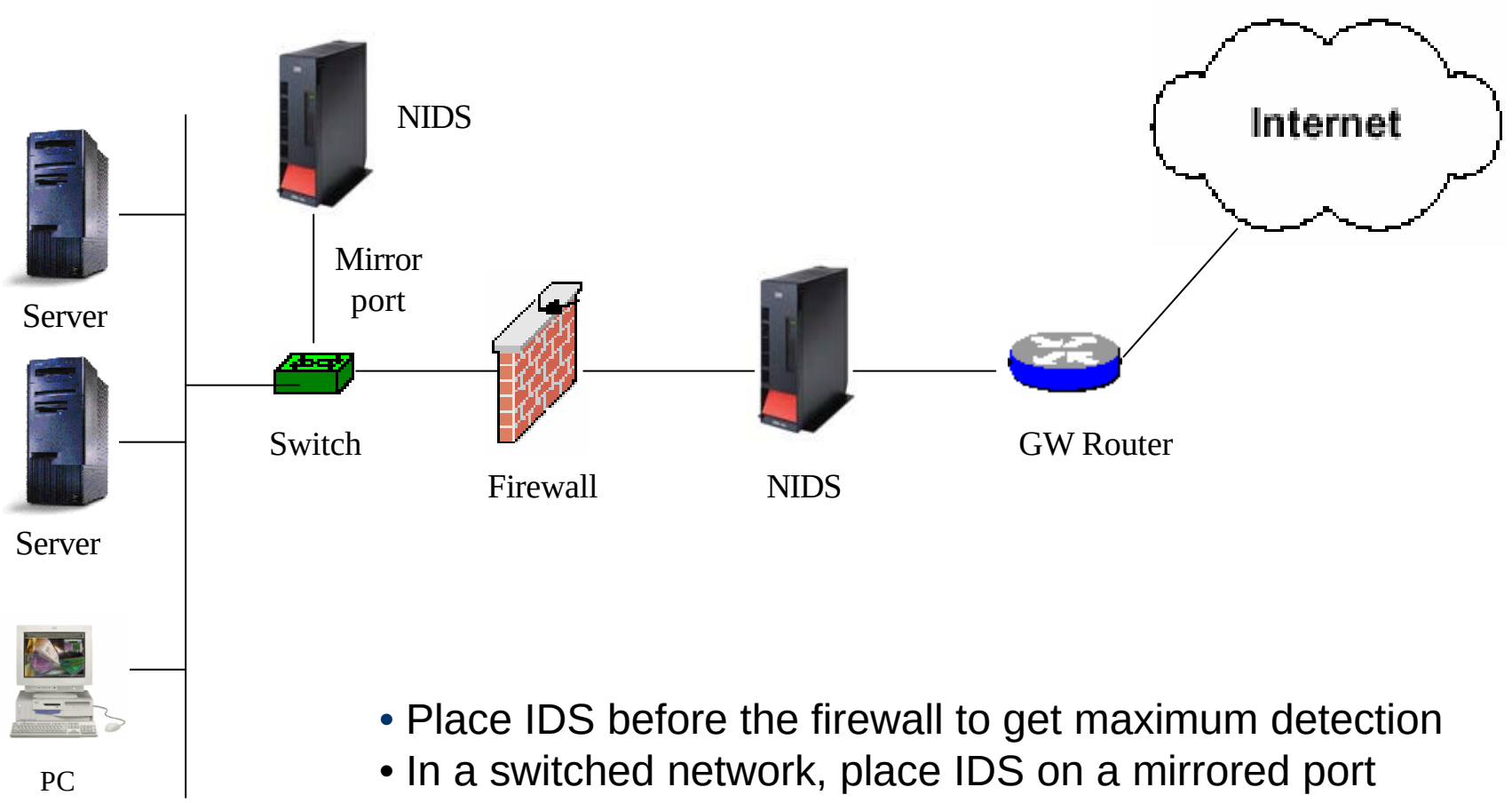
Snort Features

- a cross-platform, lightweight network intrusion detection tool
- rules based logging to perform content pattern matching
- detect a variety of attacks and probes
- buffer overflows [ALE96], stealth port scans, CGI attacks, SMB probes, and much more
- has real-time alerting capability - syslog, SMB "WinPopup" messages, or a separate "alert" file
- detection engine is programmed using a simple language that describes per packet tests and actions
- Ease of use simplifies and expedites the development of new exploit detection rules
- detect a wide variety of suspicious network traffic as well as outright attacks
- is useful when it is not cost efficient to deploy commercial NIDS sensors
- Architecture is focused on performance, simplicity, and flexibility
- is available under the GNU General Public License, and is free for use

How does Snort work?

- Sniffs, decodes the application layer data of a packet
- Can be given rules to collect traffic that has specific data contained within its application layer
- Detect many types of hostile activity, including buffer overflows, CGI scans, etc.
- Its decoded output display is somewhat more user friendly than tcpdump's output
- Can provide administrators with enough data to make informed decisions on the proper course of action in the face of suspicious activity
- Alerts administrators in real time via various methods

Snort – NIDS placement



- Place IDS before the firewall to get maximum detection
- In a switched network, place IDS on a mirrored port
- Make sure all network traffic passes the IDS host
- Best to run IDS in bridge mode for transparent network operation

Snort Architecture

There are three primary subsystems:

1. The packet decoder
2. The detection engine
3. The logging and alerting subsystem

- These subsystems ride on top of the libpcap promiscuous packet sniffing library, which provides a portable packet sniffing and filtering capability
- Program configuration, rules parsing, and data structure generation takes place before the sniffer section is initialized
- Keeps the amount of per packet processing to the minimum required to achieve the base program functionality

Snort Architecture

1. The packet decoder

   - The decode engine is organized around the layers of the protocol stack present in the supported data-link and TCP/IP
   - Speed is emphasized in this section
   - majority of the functionality of the decoder consists of setting pointers into the packet data for later analysis by the detection engine
   - provides decoding capabilities for Ethernet, SLIP, and raw (PPP)data-link protocols
   - ATM support is under development

Snort Architecture

2. The detection engine

- Snort maintains its detection rules in a two dimensional linked list of what are termed **Chain Headers** and **Chain Options**
  - **Chain Headers** - list of common attributes
  - **Chain Options** - the detection modifier options
- To speed the detection processing, the commonalities are condensed into a single Chain Header and then individual detection signatures are kept in Chain Option structures
- All rule chains are searched recursively for each packet in both directions
- The detection engine checks only those chain options which have been set by the rules parser at run-time
- The first rule that matches a decoded packet in the detection engine triggers the action specified in the rule definition and returns

Snort Architecture

3. The logging/alerting subsystem

   - is selected at run-time with command line switches
   - three logging and five alerting options are available

   - **Logging options:**
     - log packets in their decoded, human readable format to an IP-based directory structure or
     - OR in tcpdump binary format to a single log file
     - Decoded format logging allows fast analysis of data collected by the system
     - Tcpdump format is much faster to record to the disk and should be used in instances where high performance is required
     - Logging can also be turned off completely -- leaving alerts enabled for even greater performance improvements

Snort Architecture

3. The logging/alerting subsystem…contd.

- **Alerting options:**
  - Sent to syslog
  - Logged to an alert text file in two different formats – full, fast
  - Sent as WinPopup messages using the Samba program

  - syslog alerts are sent as security/authorization messages that are easily monitored with tools such as swatch
  - WinPopup alerts allow event notifications to be sent to a user-specified list of Microsoft Windows consoles
  - Full alerting writes the alert message and the packet header information
  - fast alert option writes a condensed subset of the header information - allowing greater performance under load
  - a fifth option to completely disable alerting, which is useful when alerting is unnecessary or inappropriate

Writing Snort rules

- Snort rules are simple to write, yet powerful enough to detect a wide variety of hostile or merely suspicious network traffic
- There are three base action directives that Snort can use when a packet matches a specified rule pattern: **pass, log**, or **alert**
- **Pass** rules simply drop the packet
- **Log** rules write the full packet to the logging routine that was user selected at run-time
- **Alert** rules generate an event notification using the method specified by the user at the command line
- … and then log the full packet using the selected logging mechanism to enable later analysis

Using Snort

There are three main modes in which Snort can be configured:

1. Sniffer Mode

- simply reads the packets off of the network and displays them for you in a continuous stream on the console

2. Packet logger mode

- logs the packets to the disk

3. Network intrusion detection mode

- is the most complex and configurable configurations
- allows Snort to analyze network traffic for matches against a user defined rule set
- perform several actions based upon what it sees.

Snort Lab

- Installation
- Sniffer mode
- Packet Logger Mode
- Network Intrusion Detection Mode
- NIDS Mode Output Options
- High Performance Configuration
- Changing Alert Order
- Miscellaneous

File System Integrity

File System Integrity – the security perspective

- A security administrator must make sure that the configuration files and binaries crucial to system security are not being tampered. It is generally seen that crackers after penetrating 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 the file system.

  - RPM
  - Tripwire
  - AIDE

File System Integrity – RPM

- You can use the RPM (RedHat Package Manager) to verify the integrity of the system files by using the following command

```
# rpm -Vf filename
```

- The rpm command will verify the major attributes with a built in database and report changes by printing the following characters.

  - S size differs
  - M permission and file type differs
  - 5 MD5 sum differs
  - D Device major/minor mismatch
  - L Symbolic link differs
  - U User ownership differs
  - G Group ownership differs
  - T modified time differs

File System Integrity – Tripwire

- Tripwire is a File system checker that can report changes in the files. It does this by comparing major attributes of file such as binary signature, size,md5sum etc against a previously built database. If a change is detected it will report it.

- Tripwire generally comes preinstalled in most Linux Distribution. However we do need to configure it before using.

  - **Configuration**

  - The tripwire configuration files are located in /etc/tripwire in redhat systems.

```
# cd /etc/tripwire
```

  - Edit “twcfg.txt” to suit your needs. The defaults are usually good.

```
# vi twcfg.txt
```

File System Integrity – Tripwire

- **Edit your Policy File to reflect you system**

```
# vi /etc/twpol.txt
```

- **Change**

```
HOSTNAME=localhost;
to
HOSTNAME="your_hostname";
```

- You should also add or remove entries depending what binaries or configurations you have on your system.

- After you have edited the twpol.txt, run "*twinstall.sh*"

```
# ./twinstall.sh
```

  - It will then ask you for site keyfile passphrase and local passphrase ; Both are needed to sign different files such as configuration and database files of tripwire.

  - After generating the keys it will ask you to enter the site keyfile passphrase to build the signed verion of tripwire configuration and tripwire policy files from twcfg.txt and twpol.txt to tw.cfg and tw.pol

File System Integrity – Tripwire

- **Output from twinstall.sh**

```
# ./twinstall.sh

---------------------------------------------
The Tripwire site and local passphrases are used to sign a
variety of files, such as the configuration, policy, and
database files.

Passphrases should be at least 8 characters in length
and contain both letters and numbers.

See the Tripwire manual for more information.

---------------------------------------------
Creating key files...

(When selecting a passphrase, keep in mind that good
passphrases typically have upper and lower case letters,
digits and punctuation marks, and are at least 8 characters
in length.)
```

File System Integrity – Tripwire

```
Enter the site keyfile passphrase:
Verify the site keyfile passphrase:
Generating key (this may take several minutes)...Key generation
complete.

(When selecting a passphrase, keep in mind that good passphrases
typically have upper and lower case letters, digits and punctuation
marks, and are at least 8 characters in length.)

Enter the local keyfile passphrase:
Verify the local keyfile passphrase:
Generating key (this may take several minutes)...Key generation
complete.

-------------------------------------------
Signing configuration file...
Please enter your site passphrase:
Wrote configuration file: /etc/tripwire/tw.cfg

A clear-text version of the Tripwire configuration file
/etc/tripwire/twcfg.txt has been preserved for your inspection.  It is
recommended that you delete this file manually after you have examined
it.
```

File System Integrity – Tripwire

```
---------------------------------------
Signing policy file...
Please enter your site passphrase:
Wrote policy file: /etc/tripwire/tw.pol
```

- A clear-text version of the Tripwire policy file

```
/etc/tripwire/twpol.txt
has been preserved for your inspection.  This
implements
a minimal policy, intended only to test essential
Tripwire functionality.  You should edit the policy
file
to describe your system, and then use twadmin to
generate
a new signed copy of the Tripwire policy.
```

- **Verify the creation of files “tw.cfg” and “tw.pol”**

File System Integrity – Tripwire

- After building the policy files , you can initialize the tripwire database by issuing

```
“tripwire --init” , Enter your local passphrase to continue.
```

- **Note**: if you haven't edited the twpol.txt to reflect your system , you might get File system Errors.

- For eg.

```
# tripwire –init
....
### Warning: File system error.
### Filename: /proc/scsi
### No such file or directory
### Continuing...
```

- You can go edit twpol.txt can remove the line that contain "/proc/scsi" and rebuild the policy file

File System Integrity – Tripwire

- **Database File**
  - You will find the newly created database inside “/var/lib/tripwire/” directory.
  - The file is named hostname.twd.
- **Running Tripwire Check daily**
  - You should also make sure that you have a file inside “/etc/cron.daily” named “tripwire-check” to run the tripwire check everyday.
  - “tripwire-check” will generate the report files inside “/var/lib/tripwire/report” directory.

File System Integrity – Tripwire

- **Printing Report**

```
# twprint -m p -r
  /var/lib/tripwire/report/reportfilename.twr
```

- **Updating Expected changes**
  - You can update the tripwire database if the changes reported are expected by running:

```
# tripwire --update -r
  /var/lib/tripwire/report/reportfilename.twr
```

  - It will open the report file in your configured editor .You should not remove the 'x' inside '[ ]' for the change to be updated in the tripwire database.

Questions & Answers

Incident Management

Managing Incidents

You have just been attacked or compromised, what next? Some of the best practices during an attack are :

- **Source of Attack**
  - First find out the source of attack. If the IP are being spoofed then it will be more difficult to find out the source. Generally spoof attacks can be blocked by means of spoof protection on firewalls or Cisco routers.

- **Firewalls**

```
# /sbin/iptables -i ! INTERNAL_IF -s INTERNAL_NET -j DROP
```

Or

```
# /sbin/iptables -i EXTERNAL_IF -s INTERNAL_NET -j DROP
```

- **Cisco Routers**

```
router(config)# int xxx
router(config-if)# ip verify unicast reverse-path
```

Managing Incidents ..contd

- **Block the source of attack**

```
# /sbin/iptables -I INPUT -s ATTACKER -j
  DROP
```

OR

```
# /sbin/route add -host ATTACKER reject
```

- However, you should block the attacker from your borders routers. We can achieve this in Cisco routers using :

```
router(config)# ip route attackerip 255.255.255.255
```

Managing Incidents ..contd

- **Limiting access to Legitimate Ports**

  –If you are being attacked on a legitimate port running legitimate services from multiple hosts and you can not block them all but you can limit the access by using limit module of *iptables.*

```
# /sbin/iptables -I INPUT -m limit --limit
  300/second -p udp --dport 53 -j ACCEPT
```

  –Ports that does not need to be opened globally must be blocked from outside world using the firewall

Managing Incidents ..contd

- **Some of the Best practices after a compromise are:**

  –Unplug the Ethernet cable from the server.

  –Find out what changes have been made by the cracker/hacker.

  –Analyze the system log files. System Log files may be tampered, so if you have network logging, check the logs in the log server instead.

  –Find the changes in system configuration files made, by using the file integrity checker such as *tripwire. If* you find that your binaries has been modified or tampered with, install a fresh trusted copy from the CDROM or floppy.

Managing Incidents ..contd

- **Some of the Best practices after a compromise are ..contd**

  – Find out what processes are running and compare it with the services you normally run on the server.

  – Kill all the processes you find suspicious and close ports not opened by you. Verify that no modifications have been done to the system init files such as
    - */etc/inittab*
    - */etc/rc.d/**

  – Verify that no additional users has been added to the "*/etc/passwd*" file.

  – Restore your server with the backup , but also make sure that the backup itself doesn't contain the modified system files.

  – It is always better to install a fresh copy of the OS and copy files specific to services running on the server.

  – Report the Incident to the authorities . Do not delete away the log files as these are evidence.

Questions & Answers

Backups

Why the need for backups?

- Backups are very crucial because if there is any problem that leads to a loss of your data, you can recover from such problems through your backups.
- You should always do backups regularly.
- Maintain full and reliable backups of all data, log files
- Archive all software (purchased or freeware), upgrades, and patches “off-line”
- Backup configurations used by the operating systems or applications
- Consider the media, retention requirements, storage, rotation methods
- Keep copy of a full backup off-site for disaster recovery

Different kinds of backups

- There are different kinds of backups you can perform on Linux systems.
  - System Configuration backups
  - User Data
  - Binaries

- **Configuration file backups**

  - When editing system configuration files , you should always make a backup before editing.

  - **For eg.**

```
# cd /etc/
# cp syslog.conf syslog.conf.bck-2004-02-21
```

Then

```
# vi syslog.conf
```

Backup Tools in Linux

- Tar is easily most popular backup tool used by Linux administrators . It is a very easy to use backup tool.

- You can create a compressed tar archive using the following commands.

```
# mkdir -p /backups/2004-02-21/
# cd /backups/2004-02-21/
# tar czvf usr.tar.gz /usr
# tar czvf etc.tar.gz /etc/
# tar czvf home.tar.gz /home
```

- You must copy the tar files to other media and transport it somewhere else for offsite backup. Backups on the same machine is similar to having no backups at all.

- You can restore from the tar archive using:

```
# cd /
# tar xzvf home.tar.gz
```

Backup Tools in Linux ..contd

- **CPIO**
  - *cpio* (copy in, copy out) creates archives of your files and directories for storage.
  - You can create backups from *cpio* using the following command

```
# ls /  |  cpio -o > full.cpio
```

  - You can restore from the cpio archive using

```
# cd /
```

Backup Tools in Linux ..contd

- **Dump**
  - Dump can do be used for incremental and/or full backups.
  - For eg.

```
# /sbin/dump 0uf /dev/tape /dev/hda2
```

  - Here 0uf means ,do a full backup, update /etc/dumpdates after dump is successful and the output is a file.

- **Restore**
  - One can restore from the dumped file as follows:

```
# /sbin/restore rf /dev/tape
```

Backup Tools in Linux ..contd

- **Rysnc**
  - ***rsync*** (primarily used as a remote synchronizing tool) can be also used to backup data. You can install a backup server using *rsync* daemon mode.
  - You can run *rsync* as daemon from *xinetd*

```
# cat /etc/xinetd.d/rsync
service rsync
{
        disable = no
        socket_type             = stream
        protocol                = tcp
        wait                    = no
        user                    = root
        server                  = /usr/local/bin/rsync
        server_args             = rsyncd --daemon
}
```

Backup Tools in Linux ..contd

- **Configuration File (/etc/rsyncd.conf)**

```
# cat /etc/rsyncd.conf
[test]
  comment = Test Rsync Server
  path = /backup
  read only = no
  uid = 501
  gid = 501
  auth users = bckoper
  secrets file = /etc/rsyncd.secrets
  hosts allow =  1.1.1.1
  hosts deny = *

# cat /etc/rsyncd.secrets
bckoper:98kd93k
```

Backup Tools in Linux ..contd

- **Backing up to rsync server**

```
# export RSYNC_PASSWORD=98kd93k
# /usr/local/bin/rsync /home bckoper@rsync_server::test/home
```

- **Rsync Security**
  - You can also use SSH to protect the data in transit.

- **Additional Packages**
  - There are different backup Packages availabe on the Internet also such as:
    - **Amanda**
    - **Mondo**

Questions & Answers

Server Resource Monitoring

Why is Server Resource Monitoring Important?

- Server Resource Monitoring is critical in maintaining the health of the System both in terms of Services and Security.

- There are different commands available for Server Resource Monitoring in Linux:

  - *uptime*
  - *free*
  - *ps*
  - *top*
  - *netstat*

Server Resource Monitoring

- **uptime**
  - *uptime* command will show you details such as current time, time since last reboot , number of users online and the system load.
  - Here is the output of a typical uptime command.

```
$ uptime
5:51pm  up 186 days, 12 min,  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,12 min
6 users are online on the sytem
Load average for 1 minute = 0.06
Load average for 5 minute = 0.06
Load average for 15 minute = 0.05
```

Server Resource Monitoring ..contd

- **System Load Average**
  - The system load average provides a convenient way to summarize the activity on a system. System load average is the average number of processes in the kernel's run queue during an interval.

- **free**
  - *free* will show you memory status. *'free*' produces the output from "/proc/meminfo".
  - Here is the output of a typical *free* command :

```
$ free
total       used       free     shared    buffers     cached
Mem:        514836     485724   29112  0  212108     111448
-/+ buffers/cache:     162168     352668
Swap:       875500       8420     867080
The default values are in KB.
```

Server Resource Monitoring ..contd

- **ps**
  - ps is one of the most important tools for system resource monitoring. ps reports the running processes in the sytem. ps can be used with different options.
  - For eg.
  - 'ps -aux' will print All the processes with/without controlling terminal and will also display the user running the process:

$ ps -aux

```
USER       PID %CPU %MEM    VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   1388  188 ?        S    Jan15   0:04 init
root         2  0.0  0.0      0    0 ?        SW   Jan15   0:04 [keventd]
root         3  0.0  0.0      0    0 ?        SWN  Jan15   0:00 [ksoftirqd_CPU0]
root         4  0.0  0.0      0    0 ?        SW   Jan15   1:01 [kswapd]
root         5  0.0  0.0      0    0 ?        SW   Jan15   0:00 [bdflush]
root         6  0.0  0.0      0    0 ?        SW   Jan15   0:00 [kupdated]
root         7  0.0  0.0      0    0 ?        SW<  Jan15   0:00 [mdrecoveryd]
root        11  0.0  0.0      0    0 ?        SW   Jan15   0:00 [kjournald]
root        69  0.0  0.0      0    0 ?        SW   Jan15   0:00 [khubd]
root       291  0.0  0.0      0    0 ?        SW   Jan15   0:00 [kjournald]
root       296  0.0  0.0      0    0 ?        SW   Jan15   0:00 [kjournald]
root       573  0.0  0.0   1464  480 ?        S    Jan15   0:04 syslogd -m 0
root       577  0.0  0.0   1388  404 ?        S    Jan15   0:03 klogd -x
```

Server Resource Monitoring ..contd

- **top**
  - *top* is just like ps , however it updates its stats in real time.

```
$ top
18:13:56  up 28 days,  6:39,  1 user,  load average: 0.00, 0.00, 0.00
47 processes: 44 sleeping, 2 running, 0 zombie, 1 stopped
CPU states:   1.4% user   1.3% system   0.3% nice   0.0% iowait  96.8%
   idle
Mem:   514836k av,  486024k used,   28812k free,       0k shrd,  212268k
   buff
       189980k active,             156268k inactive
Swap:  875500k av,    8420k used,  867080k free                  111456k
   cached

  PID USER     PRI  NI  SIZE SWAP  RSS SHARE STAT %CPU %MEM   TIME
   COMMAND
    1 root       8   0   216   28  188   168 S     0.0  0.0   0:04 init
    2 root       9   0     0    0    0     0 SW    0.0  0.0   0:04
   keventd
    3 root      19  19     0    0    0     0 SWN   0.0  0.0   0:00
   ksoftirqd_CPU0
    4 root       9   0     0    0    0     0 SW    0.0  0.0   1:01
```

Server Resource Monitoring ..contd

- **netstat**
  - 'netstat' will show you the network socket informations of the server.
  - For eg:
- ‘ netstat -nl' will show you the currently listening sockets

```
$ netstat -nl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address
   State
tcp        0      0 0.0.0.0:2001            0.0.0.0:*
   LISTEN
Active UNIX domain sockets (only servers)

'netstat -nt' will show you the currently open tcp sockets.
$ netstat -nt
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address
   State
tcp        0      0 1.2.3.4:2001            1.2.3.5:37826
   ESTABLISHED
```

Questions & Answers

Server Maintenance

- Some set a goal to fully and completely secure a system
- Impractical and usually an impossible goal
- A realistic goal is to set up a regular routine
- Identify/correct as many vulnerabilities as practical

Realistic Security Goals

- To make it difficult for an attacker to gain access
- Many sites have minimal or no security
- Attackers usually gain access relatively quickly and easily
- With some security, exploiting systems are decreased significantly
- The intruder will probably move on to a more vulnerable site

Realistic Security Goals

- “The idea is not that you should protect a system to the point it cannot be compromised, but to secure it at least enough that most intruders will not be able to break in easily, and will choose to direct their efforts elsewhere”

- e.g. it is just like putting iron bars and locks on our windows and doors.
- We do it not to "keep the robbers out", but to persuade them to turn their attention to our neighbors

Server Maintenance

- Setup your servers very restrictively/securely
- Run the server on a hardened operating system
- Keep current on OS and application updates
- Make sure you test updates in non-production server
- One server patch may undo a previous patch
- Scan the server after the patching up
- Hackers usually attack servers with well known/old bugs
  - e.g. MS-SQL Slammer worm attack saturated the Internet traffic
- Disable file sharing an all critical machines

Server Maintenance

- Regularly Scan Systems
- Scans will help determine that only required ports are open
- Services running on the open ports are not vulnerable
- Determine if systems have been compromised – if new open ports are found

- Perform full port scans using nmap, nessus on a regular basis
- Port scans should cover all ports TCP/UDP 1-65,535

Nmap

- it’s a Network MAPper
- powerful utility for network exploration or security auditing
- rapidly scan large networks or single host
- determine what hosts are available on the network
- what services (ports) they are offering
- what operating system (and OS version) they are running
- what type of packet filters/firewalls are in use
- runs on most types of computers
- both console and graphical versions are available
- free software, available with full source code - GNU GPL

Nmap Features

- Flexible - Supports advanced techniques for mapping out networks filled with IP filters, firewalls, routers
- Port scanning mechanisms - TCP & UDP, OS detection, pings sweeps
- Powerful - scan huge networks of hundreds of thousands of machines
- Portable - Most operating systems are supported – Linux, BSD, MacOS
- Easy – can be used with simple commands or GUI options
- Free – freely downloadable, comes with full source codeWell documented - comprehensive and up-to-date man pages, whitepapers, and tutorials
- Supported – well supported by the author
- Acclaimed - has won numerous awards, featured in magazines
- Popular – thousands download everday, included in many OS distros

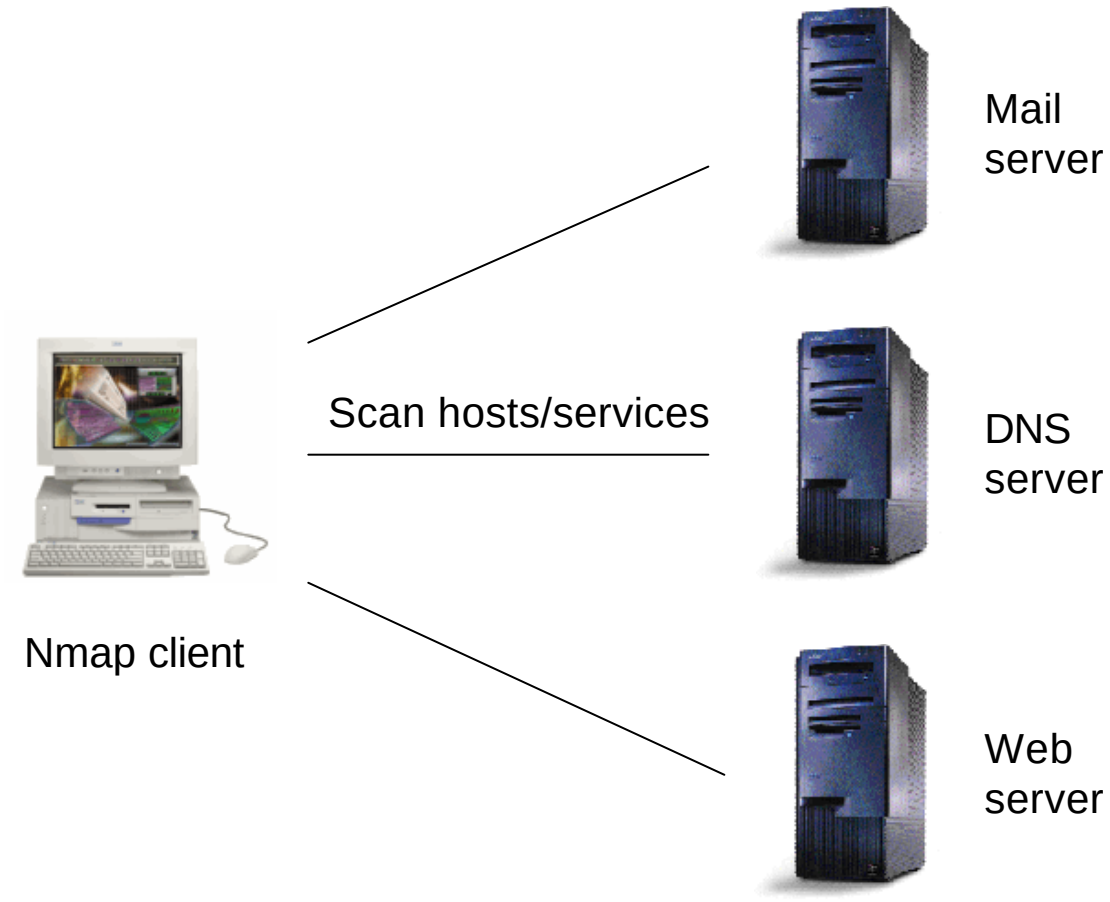
Using Nmap

NMAP does three things:

1 - ping a number of hosts to determine if they are alive or not
2 - portscan hosts to determine what services are listening
3 - attempt todetermine the OS of hosts

- NMAP is very configurable, and any of these steps may be omitted
- Although portscanning is necessary in order to do an OS scan
- There are multiple ways to accomplish most of these
- Many command line switches to tweak the way that NMAP operates

Nmap network setup



Using Nmap

Target Selection

- Specify targets on the command line or in a filename with the -i option
- Range of hosts - cert.org/24, 192.88.209.5/24, 192.88.209.0-255

Ping Scans

- Default behavior - ICMP ping sweep and TCP port 80 ACK ping sweep
- ICMP ping sweep - the usual kind of ping, -PI
- TCP port ACK ping sweep - sends an ACK to port, expects a RST, -PT
- random high-numbered port may work *much* better thru firewalls
- both an ICMP ping scan and an ACK scan to a high port, -PB32523
- intelligent firewall may block your “illegal” ACK packet
- then you may do a TCP SYN sweep with -PS
- Try ICMP pings, if not TCP ACK pings, if not TCP SYN pings…

Nmap Port Scanning

The vanilla scan is a TCP connect() scan (-sT) - loggable – don’t use this

- **SYN** scans (-sS) - workhorse of scanning methods
  also called "half-open" scans -
  send a SYN packet, look for the return SYN|ACK (open) or RST (closed) packet and then you tear down the connection before sending the ACK that would normally finish the TCP 3-way handshake
  They are also harder to detect, packet filters like ipfwadm, firewall can

- **FIN** (-sF), **NULL** (-sN) and **XMAS** (-sX) scans are all similar
  work by getting a RST back (closed) or a dropped packet (open)

- **UDP** scanning (-sU) - packet-filtered ports turn up as being open ports
  runs extremely slowly against machines with UDP packet filters

Nmap Lab

- Scan all reserved TCP ports on target.example.com in verbose mode

```
nmap -v target.example.com
```

- Launches a stealth SYN scan against 255 hosts in target's network with OS detection - requires root privileges

```
nmap -sS -O target.example.com/24
```

- Launch a stealth scan with OS detection on specified ports against 255 hosts in the network, in verbose mode

```
nmap –sS –O –v 192.168.10.0/24 –p '1-1024,1080,3128'
```

- Launch a stealth scan with OS detection on all privileged ports against 255 hosts in the network, output the results into the file /root/nmap.scan

```
nmap –sS –O 192.168.10.0/24 –oN /root/nmap.scan
```

Ndiff

- compares two nmap scans and outputs the differences
- allows monitoring of your network(s) for interesting changes in port states and visible hosts
- eliminates the need to examine voluminous raw scan output in search of the few noteworthy differences
- useful to network administrators to monitor large networks in an organized fashion
- known to work on Linux/x86, other POSIX/UNIX platforms
- requires perl 5.005_03 or later and nmap 2.53 or later
- supports HTML output for viewing results

Ndiff usage

Use the machine-parseable output of two nmap runs on the same net:

```
nmap -m first_scan.nm  10.0.0.0/24
```

later…

```
nmap -m second_scan.nm 10.0.0.0/24
```

OK, now we have two scans of the same net at different moments in time.
Now to see the changes:

```
ndiff -baseline first_scan.nm -observed second_scan.nm
```

We designate **first_scan** as the ``**baseline**'' for comparison.
Changes are reported as differences from **first_scan**.

Ndiff results

`... ndiff outputs: ...`

`missing hosts:`

< hosts present in first_scan, but missing in second_scan >

`new hosts:`

< hosts present in second_scan, but missing from first_scan >

`changed hosts:`

< hosts present in both scans, but whose port states have changed>

`[ for each host, a list of changes in port states]`

Ndiff has additional options, features for controlling output detail & format

Ngen

- synthetically create baseline nmap results
- Using the results of a previous scan as your baseline for comparison is fine for many purposes
- but if you never knew or liked the state of the scanned net
- the previous scan probably didn't yield a satisfactory baseline
- you really want as your baseline is a description of your ideal net
- one which reflects your firewall rules and/or security policy

- Ngen accepts host and port specifications, and outputs an equivalent nmap scan result to be used as an ndiff baseline
- Comparisons with this output then will show how your net varies from your ideal net

```
ngen -o baseline.nm -h 10.0.2.128/25:80 -h
  10.0.2.144-150:22,53,53u
```

Nessus

- A security scanner
- Software to remotely audit a given network or servers
- Determine whether bad guys (aka 'crackers') may break into it, or misuse it in some way
- Unlike others, Nessus does not take anything for granted
- Will *not* consider that a service is running on a fixed port
- if you run your web server on port 1234, Nessus will detect it and test its security
- will not make its security tests by the version number, but will really attempt to exploit the vulnerability
- very fast, reliable and has a modular architecture that allows you to fit it to your needs

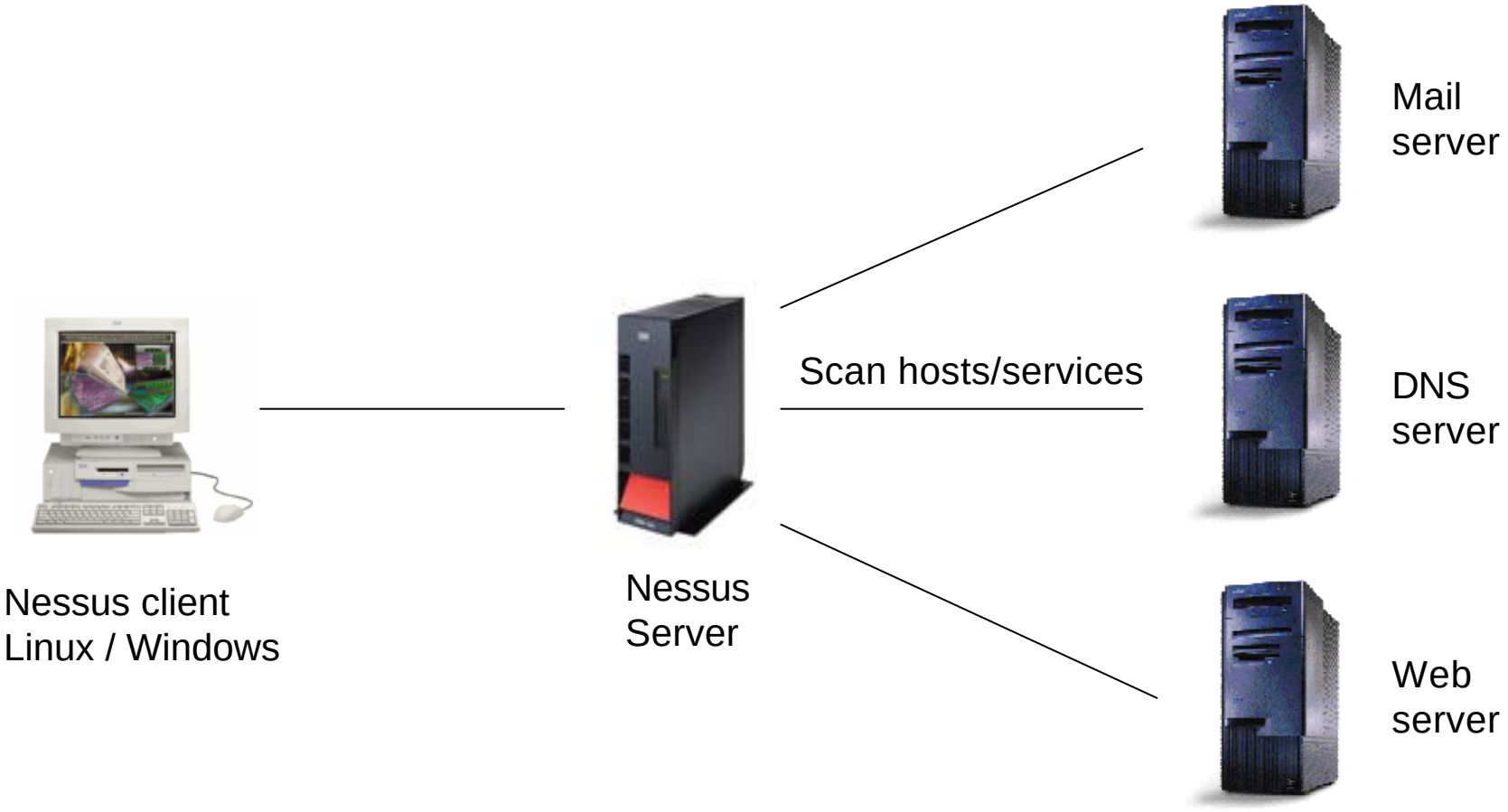
Nessus Features

- **Plug-in architecture** - Each security test is written as an ext plugin
- **Up-to-date security vulnerability database** - updated on a *daily* basis with recent security holes/bugs and available on ftp servers
- **Client-server architecture** - a server, which performs the attacks, and a client which is the front-end, can be different systems
- **Can test an unlimited amount of hosts at the same time**
- **Smart service recognition** – services on non-standard ports
- **Test multiples services** - **two** web servers (or more) on same host
- **Tests cooperation** - so that no useless tests is made
- **Complete reports** – problems and their solutions, risk levels
- **Exportable reports** - as ASCII text, HTML, HTML (pies, graphs)
- **Full SSL support** – can test https, smtps, imaps services
- **Smart plugins** - determine the right plugins for the remote service
- **Non-destructive** - can enable the "safe checks" option
- **Independent developers** - not hide any security vulnerability

Using Nessus

- Nessus is made up of two parts: a client and a server
- Server: a Unix-like system required : Linux will do
- Client: Unix-like system, Windows
- Comes as a standalone package that auto-installs itself
- download the script *nessus-installer.sh* and run it
- Create a nessusd account – to connect to server, run the scans
- Each user has a a set of restrictions to scan the network
- Configure your nessus daemon – standard file will work
- Start nessusd
- Fire up *nessus* client

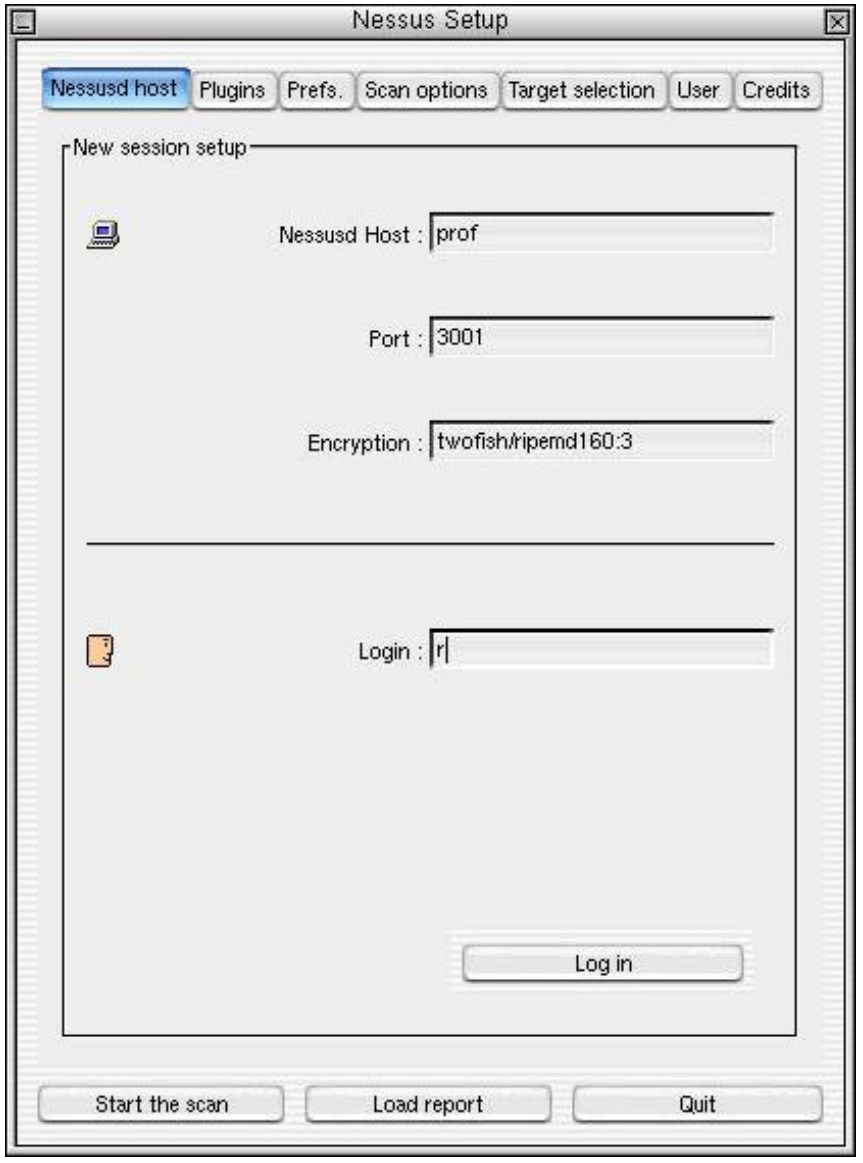
Nessus network setup



Nessus client login

Click on **Login**, since this setup is correct. Since this is the first time connecting to this server, it will ask the password. The next time you connect to it, the public key will be enough.

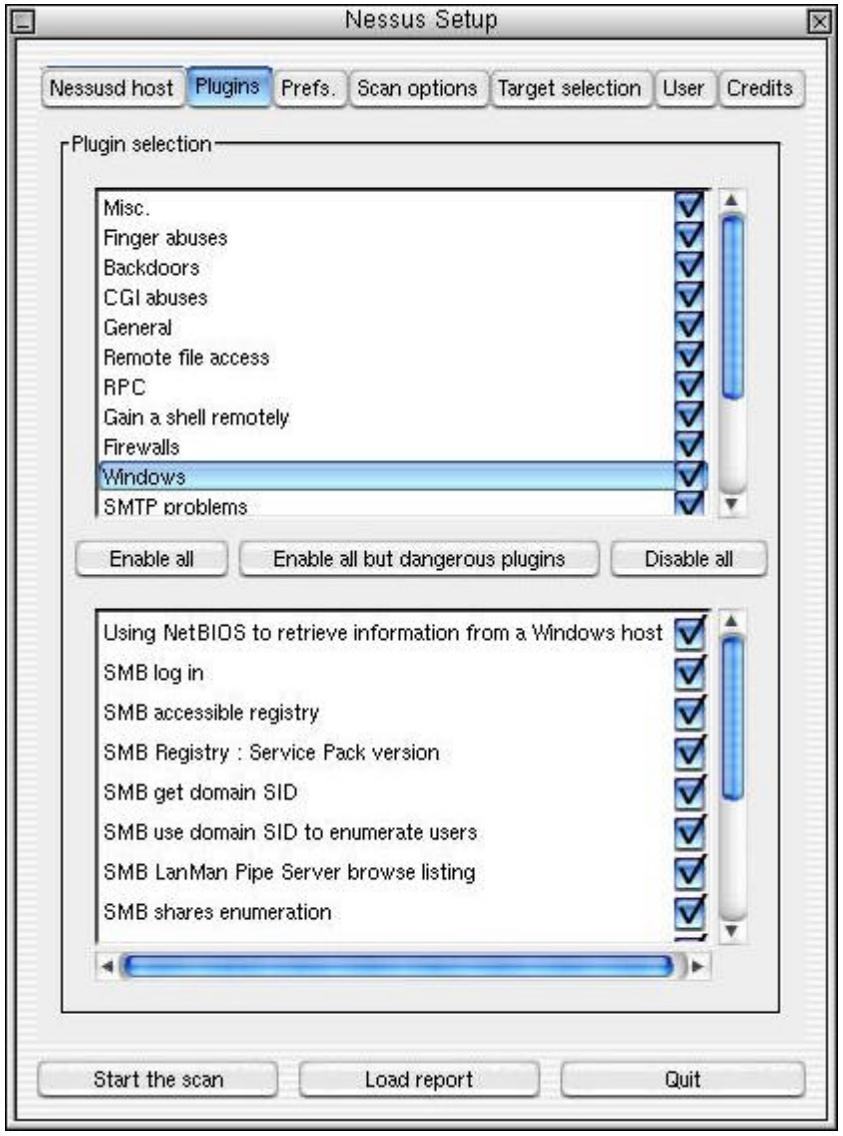
Once connected, the **Log in** button changes to **Log out**, and a **Connected** label appears at its left.



Nessus security checks configuration

Let all the security checks to be performed, except the **Denial of Service attacks**, because you do not want hosts to **crash**.

Clicking on a **plugin** name will pop up a window explaining what the plugin does.



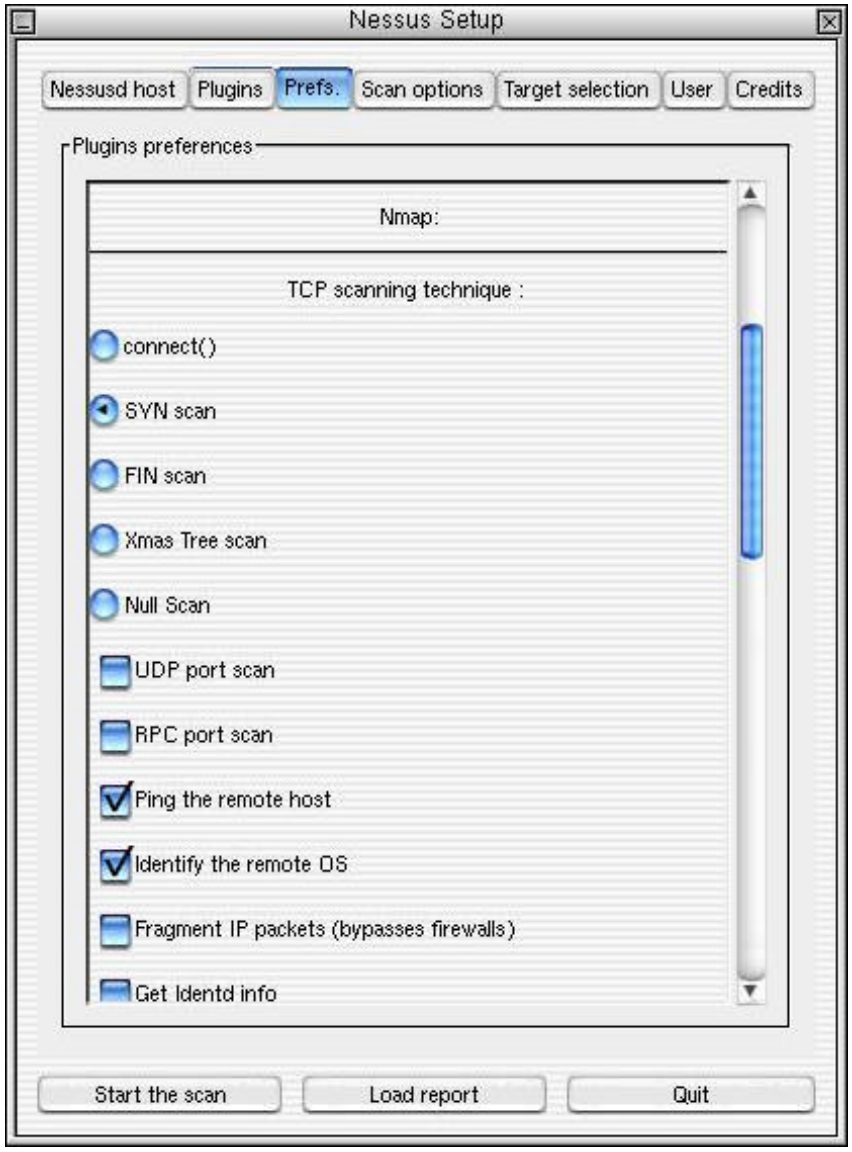
Nessus plugins preferences

Some security checks will require extra arguments.

For instance, the **pop2 overflow** security test needs a **valid pop account**. The plugin which tests whether a FTP directory is **writeable** or not asks if it should just trust the permissions or really attempt to store a file.

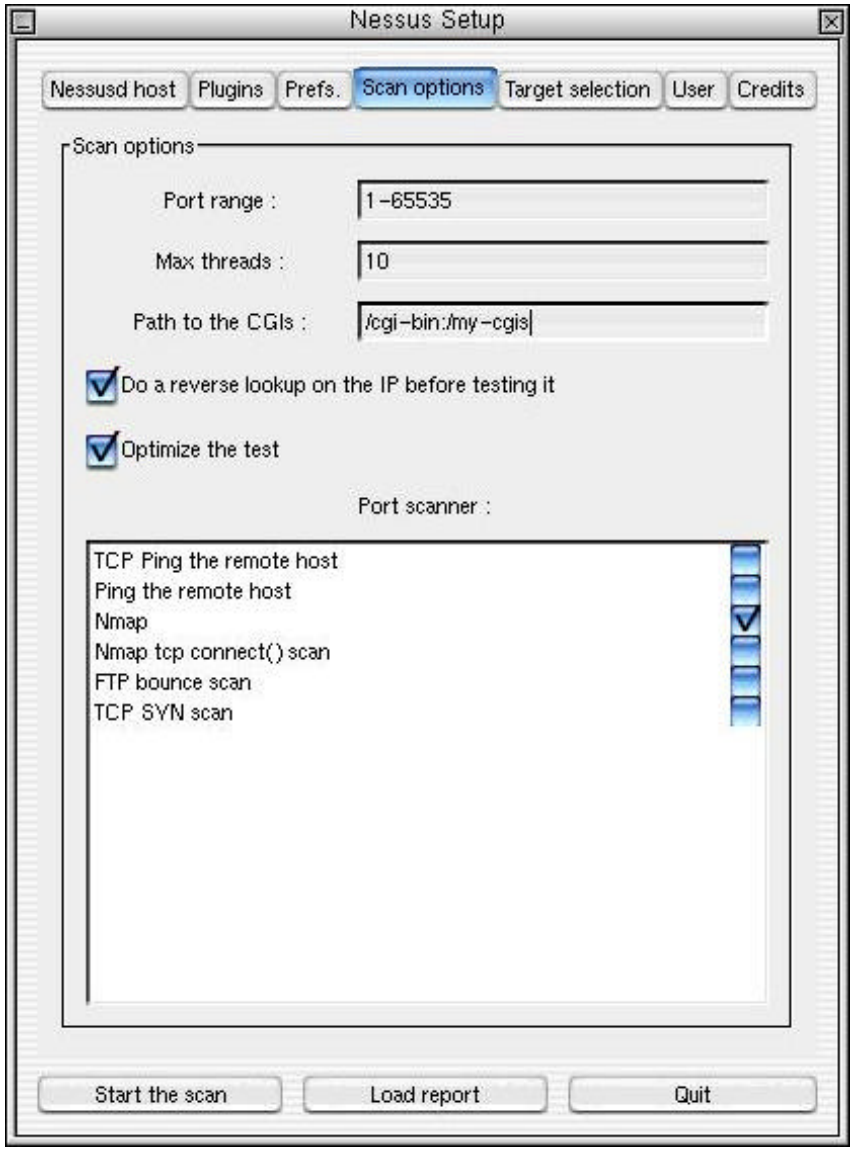
And so on...

This screen shot shows the configuration of **Nmap**



Nessus scan options

Here you choose which **port scanner** you want to use. Prefer to use the **Nmap tcp connect** scanner, since it's the **fastest**.



Nessus targets

Uncheck the 'Perform a DNS transfer zone' option.

Options to define the targets:

192.168.1.1

192.168.1.1-7

192.168.2.1-192.168.2.50

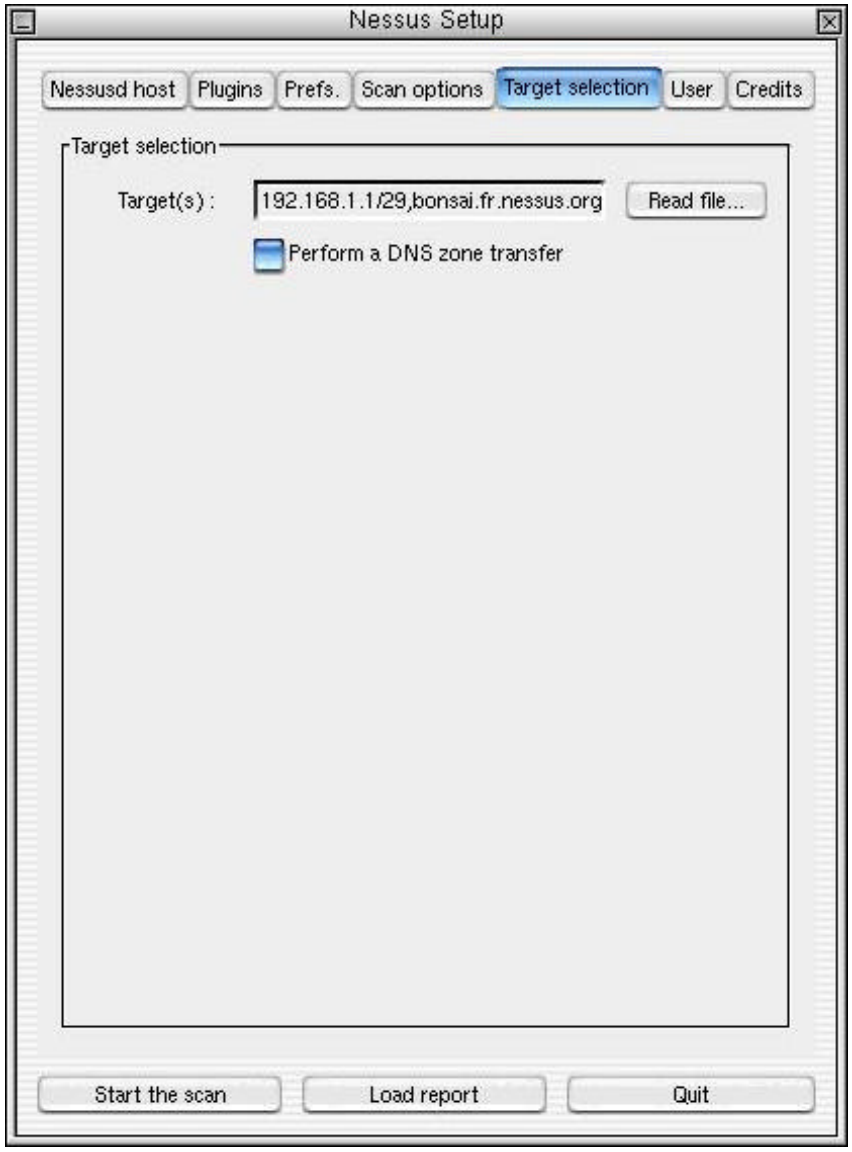
192.168.1.1/29

prof

prof.fr.nessus.org

prof, 192.168.1.1/29, ...

Any combination of the above mentioned forms separated by a comma.



Nessus rules section

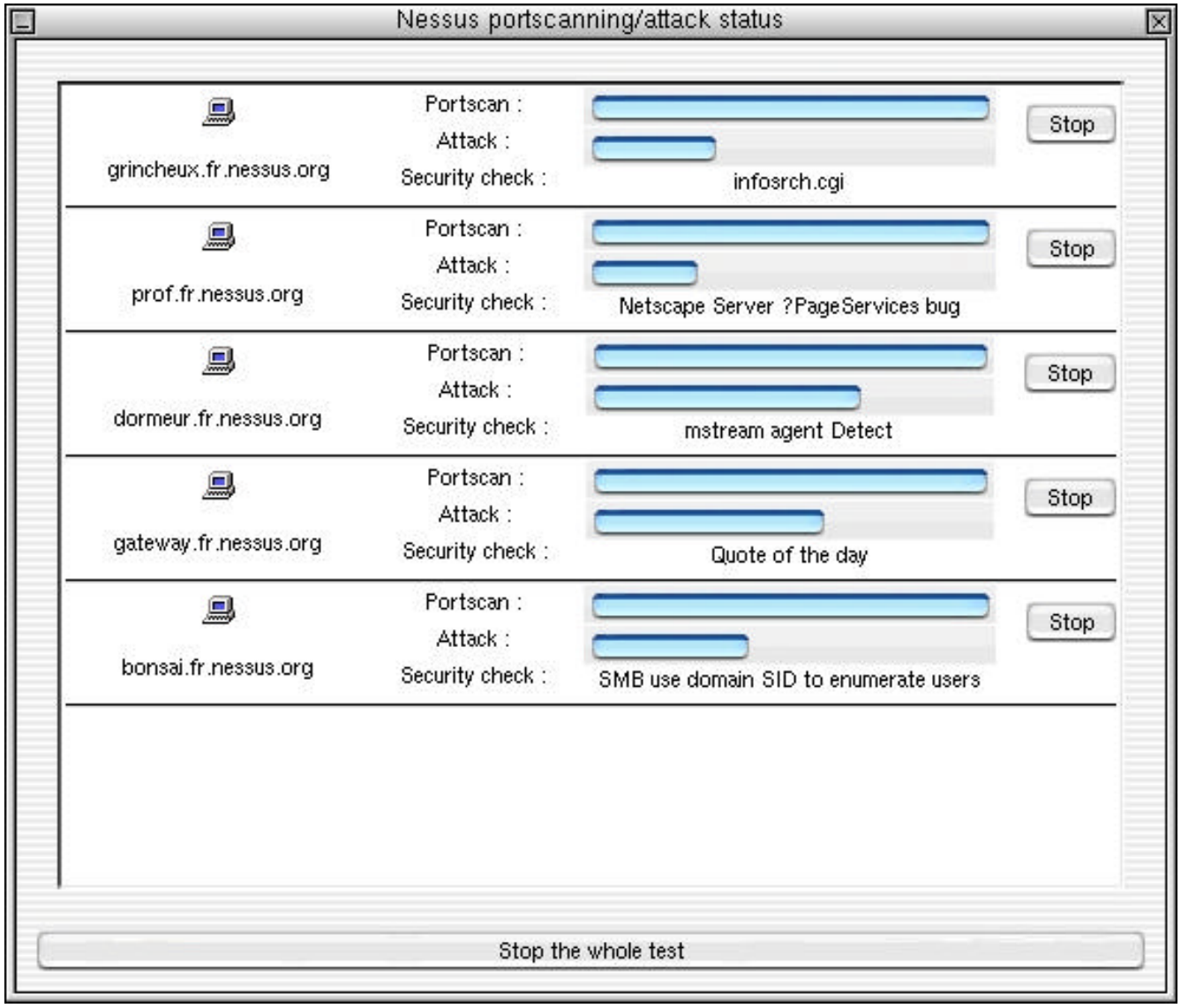
The rules allow a user to **restrict** his test.

For instance, if you want to test 192.168.1.0/29, except 192.168.1.2. The rule set entered allows you to do that.

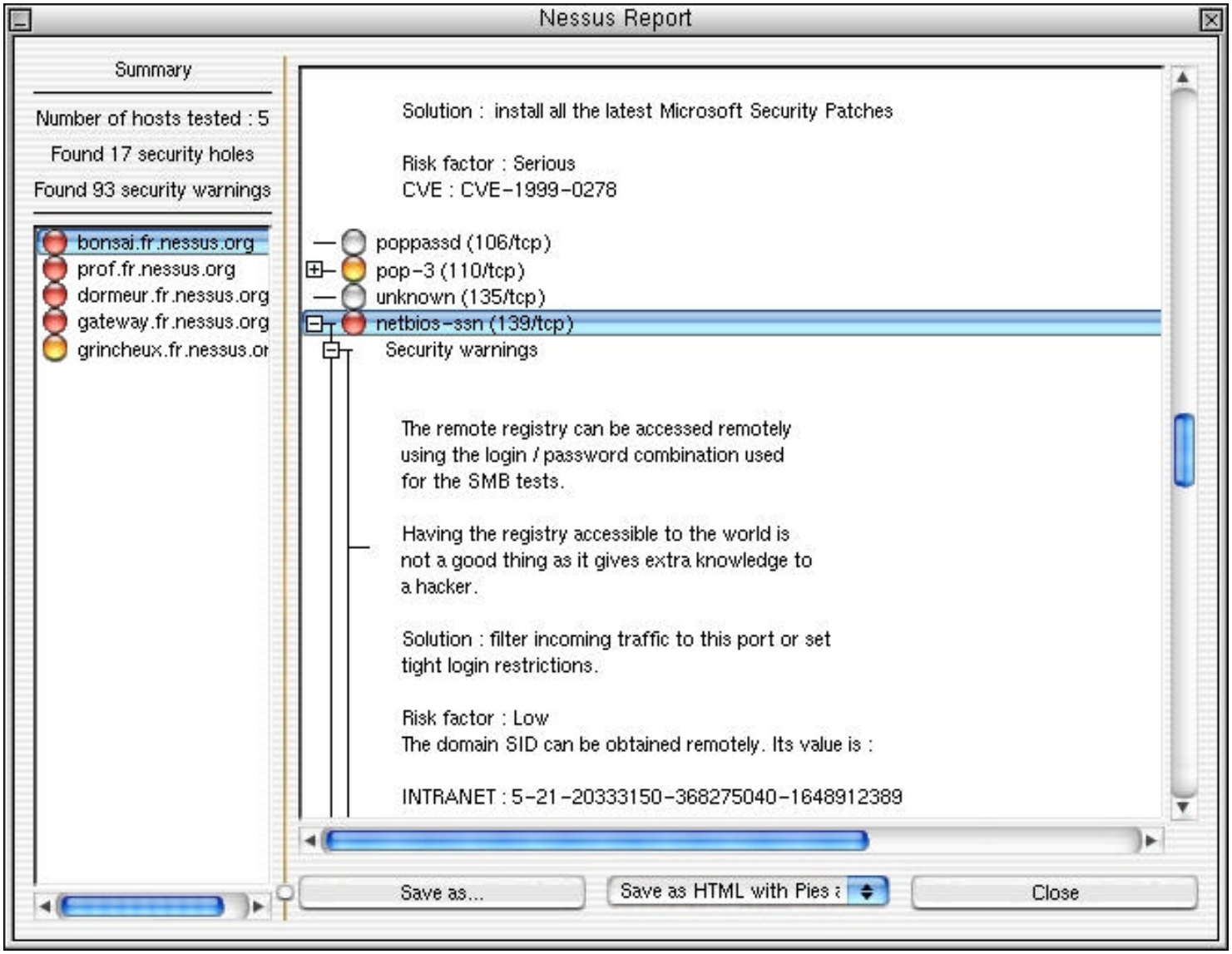
Once all of this is done,

Start the scan…

Nessus scan in progress



Nessus report window



Nessus Lab

- Installation
- Using Nessus
- Configuring Nessus
- Scanning with Nessus
- Viewing Nessus report

Content Filtering

- Viruses in web-based e-mails & files
- Transmission of confidential data
- Malicious code from websites can crash network
- Recreational surfing and e-mail
- Downloading multimedia files
- Valuable bandwidth wastage
- Spam that wastes bandwidth, time, and disk space
- Pornography and offensive content
- Instant message and peer-to-peer traffic
- Leaked trade secrets

Content Filtering

- Monitoring and controlling the electronic data entering or leaving an organization
- Any content you read, send or receive carries a risk
- Firewalls, passwords, encryption can't protect you
- Results could be lost business, productivity, lawsuits
- Content violations cost billions of dollars each year
- Filters help companies stop unwanted content

Content Filtering

- Enforce Internet access policies that prevent users and employees from accessing inappropriate or unproductive content on the network
- Manage all Internet content, Web and e-mail
- Reduce recreational Internet surfing
- Secure your network from web-based threats
- Control IM traffic and service protocols
- Restrict non-work related activities including online games, Internet shopping, stock trading, Internet radio, streaming media and MP3 downloads
- Strip and replace potentially dangerous active content from web page, while still serving remainder of the web content
- Increase employee productivity
- Limit the legal liability associated with pornographic or illegal file downloads

Content Filtering

Manage Internet traffic to optimize network bandwidth

Apply any web usage criteria:

content type, file type (downloaded or streamed), bandwidth allocated, time-of-day, time spent online, groups or employees affected

Dynamic filtering:

- neural network technology to block access to questionable sites not yet classified

Content Filtering

Products:

- SurfControl www.surfcontrol.com
- Blue Coat Systems www.bluecoat.com
- Websense www.websense.com
- NetIQMarshal www.marshalsoftware.com

Buffer Overflow

- Also known as "stack smashing" technique
- Is the most common way used in exploits to break the security of programs

Buffer Overflow – how?

- Send an unexpected amount of input data to a program
- Cause a buffer overflow that allows the attacker to make the program execute arbitrary assembler code
- Grant him the access to the system, destroy the system files or do anything else.

Buffer Overflow Protection

- Stack Shield

www.angelfire.com/sk/stackshield/index.html

- StackGuard

www.immunix.org/stackguard.html

- Libsafe

www.research.avayalabs.com/project/libsafe

Libsafe

- New method to detect and handle buffer overflow exploitation
- Does not require any modification to the operating system and works with existing binary programs
- Does not require access to the source code of defective programs
- Does not require recompilation or off-line processing of binaries
- Can be implemented on a system-wide basis transparently
- Is based on a middleware software layer that intercepts all function calls made to library functions that are known to be vulnerable

Libsafe

- Ensures that any buffer overflows are contained within the current stack frame
- Prevents attackers from 'smashing' (overwriting) the return address and hijacking the control flow of a running program
- Has demonstrated its ability to detect and prevent several known attacks
- Real benefit is its ability to prevent yet unknown attacks
- The performance overhead of libsafe is negligible.
- libsafe is extremely easy to install and use

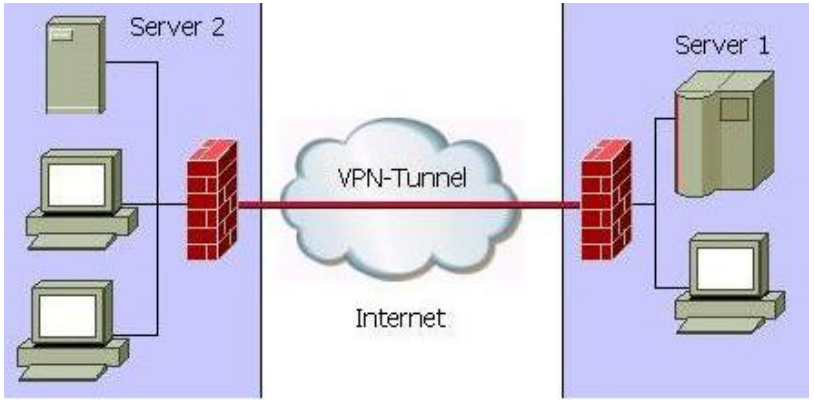
Buffer Overflow Protection Limitations

- All the methods/tools described are limited
- No tool can solve completely the problem of buffer overflow
- Can decrease the probability of stack smashing attacks
- Code scrutiny (writing secure code) is still the best possible solution to these attacks
- Programmers should be educated to prevent & minimize the use of standard unsafe functions

IPSEC

IPSEC

- IPSEC stands for Internet Protocol Security. It uses strong cryptography for both authentication and encryption services. With IPSEC you can build secure tunnels through unsecured networks such as the Internet. The data passing by the tunnel is encrypted by the originating gateway and decrypted by the receiving gateway thereby creating a VPN (Virtual Private Network).



IPSEC Software for Linux

- **FreeS/WAN**
  - Freeswan is a free Implementation of IPSEC in Linux.
  - You can download Freeswan from http://www.freeswan.org/download.html
- **Freeswan Installation**
  - You should have IPSEC support in your kernel to use FreeSwan. Most versions of Linux have IPSEC support compiled into the kernel. If you have other kernels you

FreeSwan Installation

- **Download freeswan**

- **Verify the package**

```
$ md5sum freeswan-2.05.tar.gz
20b6ef266d29b25b9a2581a0a33afeec  freeswan-
  2.05.tar.gz
$ md5sum freeswan-2.05.tar.gz >md5sum
$ diff md5sum freeswan-2.05.tar.gz.md5
```

- **Install**

```
$ tar xzvf freeswan-2.05.tar.gz
$ cd freeswan-2.05
$su
# make oldmod
# make minstall
```

FreeSwan Installation

- **Starting IPSEC**

```
# service ipsec start
```

- **Verify IPSEC**

```
# /usr/local/sbin/ipsec verify
Checking your system to see if IPsec got installed and started
   correctly:
Version check and ipsec on-path
    [OK]
Linux FreeS/WAN 2.05
Checking for IPsec kernel support: found KLIPS
   [OK]
Checking that pluto is running
            [OK]

Opportunistic Encryption DNS checks:
Looking for TXT in forward map: latitude.wlink.com.np
    [MISSING]
Does the machine have at least one non-private address?
    [OK]
Looking for TXT in reverse map: 1.1.1.1.in-addr.arpa.
    [MISSING]
```

FreeSwan Cofiguration

- **Firewall Consideration**
  - You need to allow UDP port 500 and protocol ESP(50) for IPSEC to work.

```
# /sbin/iptables -A INPUT -p udp --dport 500 -j
  ACCEPT
# /sbin/iptables -A INPUT -p 50 -j ACCEPT
```

FreeSwan Configuration

- **Configuration**
  - You can edit the IPSEC configuration to add new configuration. I have added a test connection below:

```
# vi /etc/ipsec.conf
# basic configuration
config setup
        # Debug-logging controls:  "none" for (almost) none, "all" for
   lots.
        # klipsdebug=all
        # plutodebug=dns
        interfaces="ipsec0=eth0"
# Add connections here.
conn test
        left=1.1.1.1
        leftsubnet=192.168.10.0/24
        leftid=@server1
        leftrsasigkey=left servers public key
        right=2.2.2.2
        rightsubnet=192.168.0.0/24
```

FreeSwan Configuration

- **Getting Public Keys**
  - You can get the public key by:

```
# ipsec showhostkey --left        (on left server)
# ipsec showhostkey --right       (on right server)
```

  - Note: Both VPN gateways much have the same config in their “/etc/ipsec.conf”

  - You can start the IPSEC connection by issuing:

```
# /usr/local/sbin/ipsec auto --up test
```

  - **Test**

```
ping -I 192.168.10.1 192.168.0.1        (from right
  server)
```

FreeSwan Installation Verification and Monitor

- ***ifconfig***
- You can verify that IPSEC is up by doing "/sbin/ifconfig". If you can see an Interface named "ipsecX" the installation is correct and working.

```
# ifconfig
ipsec0    Link encap:Ethernet  HWaddr 01:04:63:42:2D:1A
          inet addr:1.1.1.1 Mask:255.255.255.0
          UP RUNNING NOARP  MTU:16260  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

- ***route***
  - You can also use "/sbin/route -n" command to verify that all the network you want to protect are being sent via ipsec Interface.

```
# route -n
Kernel IP routing table
Destination       Gateway      Genmask   Flags Metric Ref
   Use Iface
10.0.0.0      1.1.1.2           255.0.0.0     U     0      0
   0 ipsec0
```

Questions and Answers

SPAM

- UCE (Unsolicited Commercial Email) is the leading complaint of Internet users
- Junk e-mail is more than just annoying
- It costs Internet users and Internet-based businesses millions per year
- Junk e-mailers can get into the business very cheaply
- The costs of email spam are borne by recipients
- Consumes extra unnecessary resources: CPU, BW
- Causes slower service, server crashes, time wasted downloading unwanted mail

Antivirus For MailServers

Compiled from various sources on the Internet

Need For Antivirus

We should use Anti virus in Mail servers to protect from malicious viruses from reaching the User's Mailbox. There are different kinds of Anti Virus Engines available for Linux, both commercial and Open source. Some of the Commercial ones are as follows:

- Rav Anti virus
- Panda Anti virus
- F-prot Anti virus

We will be discussing the installation of Open source Antivirus ClamAV and qmail-scanner it this section.

Installation of ClamAV

You can download the Latest Version of ClamAV from http://www.clamav.net

```
# groupadd clamav
# useradd -g clamav -s /bin/false -c "Clam AntiVirus" clamav

$ tar xzvf clamav-x.yz.tar.gz
$ cd clamav-x.yz

$ ./configure --sysconfdir=/etc

$ make
$ su
# make install
```

Installation of ClamAV .. contd

Configuration

If you are going to use the daemon you have to configure it because it won't run with default settings:

```
$ clamd
ERROR: Please edit the example config file
/etc/clamav.conf.
```

Remove the "Example" directive and make other changes suitable for your system.

Installation of qmail-scanner

You can download the Latest version from :

http://qmail-scanner.sourceforge.net/

Prerequisites:

Maildrop 1.3.8+ http://download.sourceforge.net/courier/

TNEF unpacker http://sourceforge.net/projects/tnef/

Perl Modules:

Time::HiRes
http://search.cpan.org/search?module=Time::HiRes

DB_File
http://search.cpan.org/search?module=DB_File

Sys:Syslog
http://search.cpan.org/search?module=Sys::Syslog

Installation of qmail-scanner .. contd

Patch qmail

You should also patch your qmail program with the patch available from http://www.qmail.org/qmailqueue-patch. This patch will cause any qmail program that would run qmail-queue to look for an environment variable QMAILQUEUE. If it is present, it is used in place of the string "bin/qmail-queue" when running qmail-queue.

Installation of qmail-scanner .. contd

Add the Qmail Scanner Users

```
# groupadd qscand
# useradd -g qscand -s /bin/false qscand
```

Extract the Installation File

```
$ tar xzvf qmail-scanner.tar.gz
$ cd qmail-scanner
```

Installation of qmail-scanner .. contd

```
 ./configure \
--spooldir /var/spool/qmailscan \
--qmaildir /var/qmail \
--bindir /var/qmail/bin \
--qmail-queue-binary /var/qmail/bin/qmail-queue \
--admin root \
--domain qmail-scanner.domain.com \
--notify sender,admin \
--local-domains qmail-scanner.domain.com \
--silent-viruses auto \
--lang en_GB \
--debug 1 \
--unzip 0 \
--add-dscr-hdrs 0 \
--archive 0 \
--redundant no \
--log-details syslog \
--log-crypto 0 \
--fix-mime 1  \
--scanners "auto" \
```

Installation of qmail-scanner .. contd

```
$ chmod 4755 /var/qmail/bin/qmail-sacnner.pl
$ /var/qmail/bin/qmail-scanner-queue.pl -h (help)
$ /var/qmail/bin/qmail-scanner-queue.pl -g (need to run once)
```

Add the following lines

```
# vi /etc/tcp.smtp
127.:allow,RELAYCLIENT="",QMAILQUEUE="/var/qmail/bin/qmail-scanner-queue.pl"
1.1.1.:allow,RELAYCLIENT="",QMAILQUEUE="/var/qmail/bin/qmail-scanner-queue.pl"
```

```
Rebuild cdb
# tcprules /etc/tcp.smtp.cdb /etc/tcp.smtp.tmp < /etc/tcp.smtp
```

Installation of qmail-scanner .. contd

The qmail-scanner installation should automatically detect the installation of Clamav. After installation you should start qmail as you normally do. However this time , before the delivery of mails the qmail program will run /var/qmail/bin/qmail-scanner.pl as instructed by the QMAILQUEUE variable.

The Mail is then parsed by qmail-scanner and sent to ClamAV for Virus Checking. If a virus is found the mail is bounced back to the Sender informing him about the presence of the Virus

Questions & Answers

Security Sites

- CERT – www.cert.org
- SANS – www.sans.org
- Help Net Security - www.net-security.org
- ISS Xforce Alerts - xforce.iss.net/xforce/alerts
- Hacker Whacker - www.hackerwhacker.com
- Linux Security - www.linuxsecurity.com
- Internet Storm Center - isc.incidents.org
- Security Stats - www.securitystats.com
- Cisco Advisories - http://www.cisco.com/warp/public/707/advisory.html
- Microsoft TechNet - http://www.microsoft.com/technet/security/default.asp