Network Security Planning

Gaurab Raj Upadhaya

Network Security Applications

- Network security risks
  - **Open architecture of the Internet Protocol (IP)**
  - **Common security breaches and attacks**
  - **Mistakes People Make that Lead to Security Breaches**
- Best security practices
  - **Benefits**
  - **Network security best practices**
  - **Host security best practices**

Q & A

Network Security Applications

Network security risks

- **Open architecture of TCP/IP (the protocol of the Internet) :**
- highly efficient, cost-effective, and flexible communications protocol for local and global communications
- widely adopted on the global Internet and in the internal networks of large corporations
- was designed twenty years ago when the Internet consisted of a few hundred closely controlled hosts with limited security
- now connects millions of computers, controlled by millions of individuals and organizations
- core network is administered by thousands of competing operators
- this complex network spans the whole globe, connected by fibers, leased lines, dial-up modems, and mobile phones
- while very tolerant of random errors, TCP/IP is vulnerable to a number of malicious attacks

Network Security Applications

Network security risks …contd.

- **Most common types of threats & attacks include:**
  - Unauthorized access – insecure hosts, cracking
  - Eavesdropping a transmission – access to the medium
    - looking for passwords, credit card numbers, or business secrets
  - Hijacking, or taking over a communication
    - inspect and modify any data being transmitted
  - IP spoofing, or faking network addresses
    - Impersonate to fool access control mechanisms
    - redirect connections to a fake server
  - DOS attacks
    - interruption of service due to system destruction or using up all available system resources for the service – CPU, memory, bandwidth

Network Security Applications

Mistakes People Make that Lead to Security Breaches

- Technological holes account for a great number of the successful break-ins, but people do their share, as well:

- **The Five Worst Security Mistakes End Users Make**

1. Failing to install anti-virus, keep its signatures up to date, and perform full system scans regularly.
2. Opening unsolicited e-mail attachments without verifying their source and checking their content first, or executing games or screen savers or other programs from untrusted sources.
3. Failing to install security patches-especially for Microsoft Office, Microsoft Internet Explorer, Outlook, Windows OS.
4. Not making and testing backups.
5. Using a modem while connected through a local area network.

Network Security Applications

Mistakes People Make that Lead to Security Breaches

The Seven Worst Security Mistakes Senior Executives Make

1. Assigning untrained people to maintain security and providing neither the training nor the time to make it possible to learn and do the job.
2. Failing to understand the relationship of information security to the business problem-they understand physical security but do not see the consequences of poor information security.
3. Failing to deal with the operational aspects of security: making a few fixes and then not allowing the follow through necessary to ensure the problems stay fixed
4. Relying primarily on a firewall
5. Failing to realize how much money their information and organizational reputations are worth
6. Authorizing reactive, short-term fixes so problems re-emerge rapidly.
7. Pretending the problem will go away if they ignore it.

Network Security Applications

Mistakes People Make that Lead to Security Breaches

The Ten Worst Security Mistakes IT People Make

1. Connecting systems to the Internet before hardening them.
2. Connecting test systems to the Internet with default accounts/passwords
3. Failing to update systems when security holes are found
4. Using telnet and other unencrypted protocols for managing systems, routers, firewalls, and PKI.
5. Giving users passwords over the phone or changing user passwords in response to telephone or personal requests when the requester is not authenticated.
6. Failing to maintain and test backups.
7. Running unnecessary services : ftpd, telnetd, finger, rpc, mail, rservices
8. Implementing firewalls with rules that don't stop malicious or dangerous traffic - incoming and outgoing.
9. Failing to implement or update virus detection software
10. Failing to educate users on what to look for and what to do when they see a potential security problem.

Network Security Applications

- **Security Best Practices**
  - Some set a goal to fully and completely secure a system
  - But this is impractical and usually an impossible goal to make a system full-proof
  - A realistic goal is to set up a regular routine where you identify/correct as many vulnerabilities as practical

Network Security Applications

Security Best Practices

- **Benefits of implementing best security practices:**

- To make it so difficult for an attacker to gain access that he gives up before he gets in
- Many sites have minimal or no security - attackers usually gain access relatively quickly and with a low level of expertise
- With some security, chances of an attacker exploiting its systems are decreased significantly - the intruder will probably move on to a more vulnerable site
- “The idea is not that you should protect a system to the point it cannot be compromised, but to secure it at least enough so that most intruders will not be able to break in, 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

Network Security Applications

Security Best Practices

- **Benefits of implementing best security practices: …contd.**

- ROI aspect to implementing effective Best Security Practices
- Rather than directing our efforts at protecting against the thousands of specific threats (this exploit, that Trojan virus, these mis-configurations)
- Focus our energies into tasks that provide the most comprehensive protection against the majority of threats
- Best Security Practices are very dynamic, constantly changing and evolving
- Administrators should include their own Best Security Practices and modify those mentioned here to best fit their environment

Network Security Applications

Security Best Practices

- **Points to ponder:**

- Take into consideration your needs risks, resources, and then apply to your systems to most effectively protect them from intrusion or disruption
- Information systems are unavoidably complex and fluid, so the most effective way to apply security is in layers
- You should place security measures at different points in your network, allowing each to do what it does best
- From an attacker's perspective, you have constructed a series of obstacles of varying difficulty between the attacker and your systems
- Secure each component in your system (firewalls, routers, servers, hosts, and appliances) so that even if an attacker works their way through your obstacle-course, at the end they will find systems that are resistant to attack

Network Security Applications

Security Best Practices

Backup

- Maintain full and reliable backups of all data, log files
- Archive all software (purchased or freeware), upgrades, and patches off-line so that it can be reloaded when necessary
- Backup configurations, such as the Windows registry and text/binary configuration files, used by the operating systems or applications
- Consider the media, retention requirements, storage, rotation, methods (incremental, differential, full) and the scheduling
- Keep copy of a full backup in a secure off-site location for disaster recovery

Network Security Applications

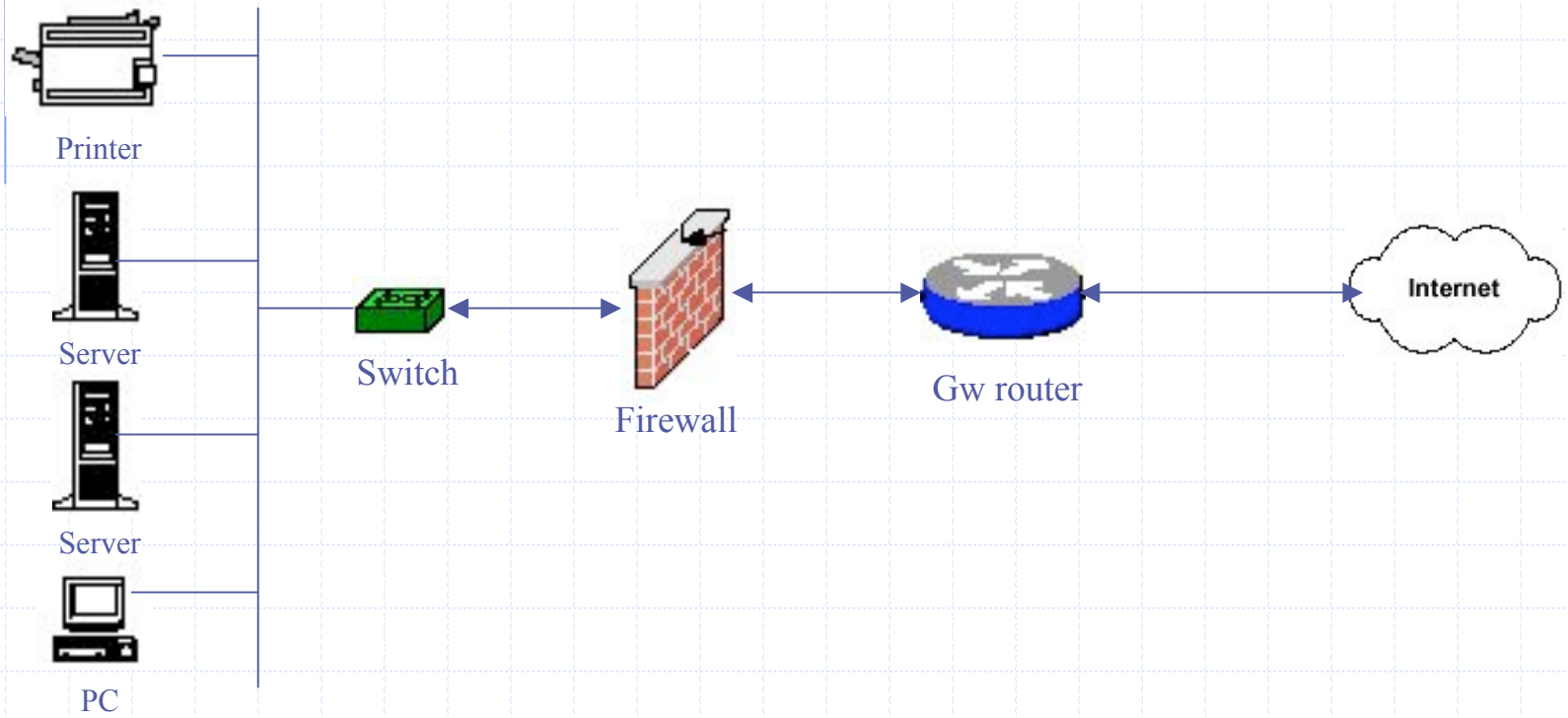
Security Best Practices

- **Secure your network and hosts properly**
- **Firewall**
  - 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

Network Security Applications

Security Best Practices

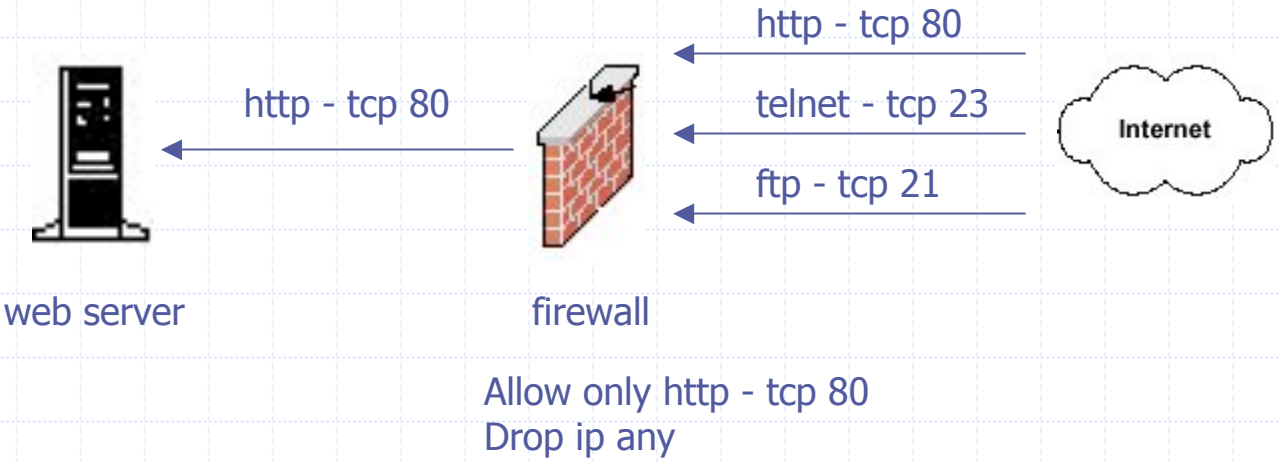
- **A typical firewall setup**



Network Security Applications

Security Best Practices

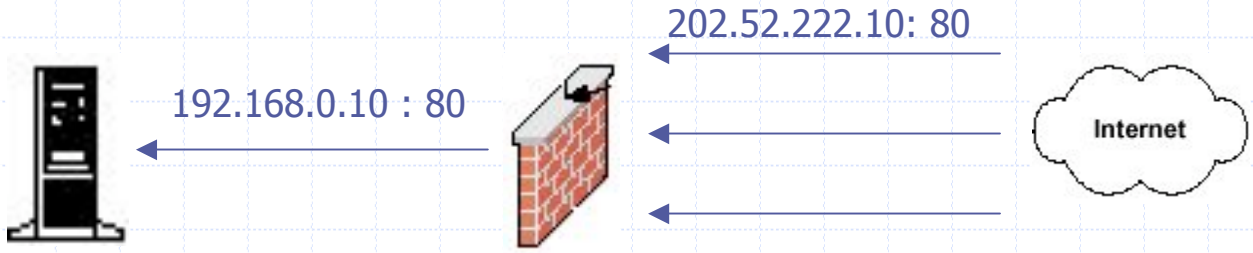
- **Types of firewalls:**
- **Packet filtering firewalls**
  - examines the source and destination address of the data packet and either allows or denies the packet from traveling the network
  - blocks access through the firewall to any packets, which try to access ports which have been declared "off-limits"



Network Security Applications

Security Best Practices

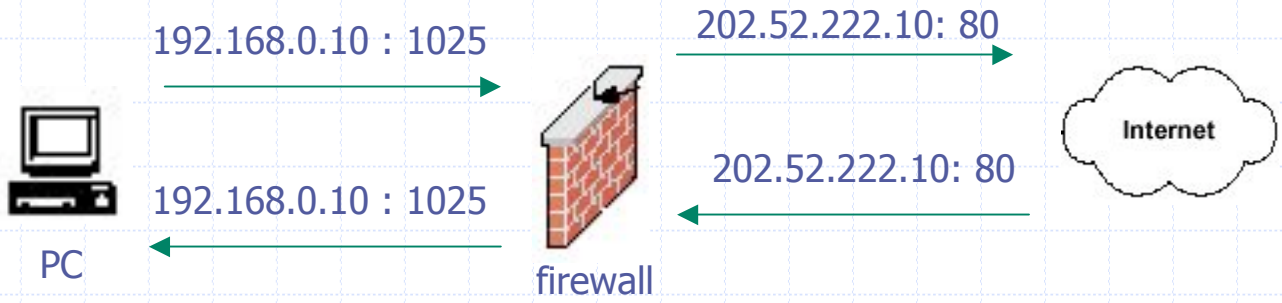
- **Types of firewalls:**
- **Application layer firewalls**
  - Also known proxy firewalls, application gateway
  - attempts to hide the configuration of the network behind the firewall by acting on behalf of that network/servers
  - All requests for access are translated at the firewall so that all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



Network Security Applications

Security Best Practices

- **Types of firewalls:**
- **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

Network Security Applications

Security Best Practices

Firewall Best Practices

- Regardless of which type of firewall, someone has to configure the firewall to make it work properly
- The rules for access must be defined and entered into the firewall for enforcement
- A security manager is usually responsible for the firewall configuration

Network Security Applications

Security Best Practices

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
- 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

Network Security Applications

Security Best Practices

Firewall Best Practices

- Disable or uninstall any unnecessary services and software on the firewall
- Limit the number of applications that run on the firewall
- Consider running antivirus, content filtering, VPN, DHCP on other systems
- Let the firewall do what it's best at doing
- Do not rely on packet filtering alone. Use stateful inspection and application proxies if possible
- Ensure that you're filtering packets for illegal/incorrect addresses – to avoid IP spoofing
- Ensure that physical access to the firewall is controlled
- Use firewalls internally to segment networks between different departments and permit access control based upon business needs
- Remember that firewalls won't prevent attacks that originate from inside your network
- Consider outsourcing your firewall management to leverage the managed security service providers' expertise, network trending analysis and intelligence, and to save time and money

Network Security Applications

Security Best Practices

- Firewall products:

Iptables	www.iptables.org
Ipchains	netfilter.samba.org/ipchains
Cisco PIX	www.cisco.com
Checkpoint	www.checkpoint.com
Border Manager	www.novell.com
Winroute	www.winroute.com

Network Security Applications

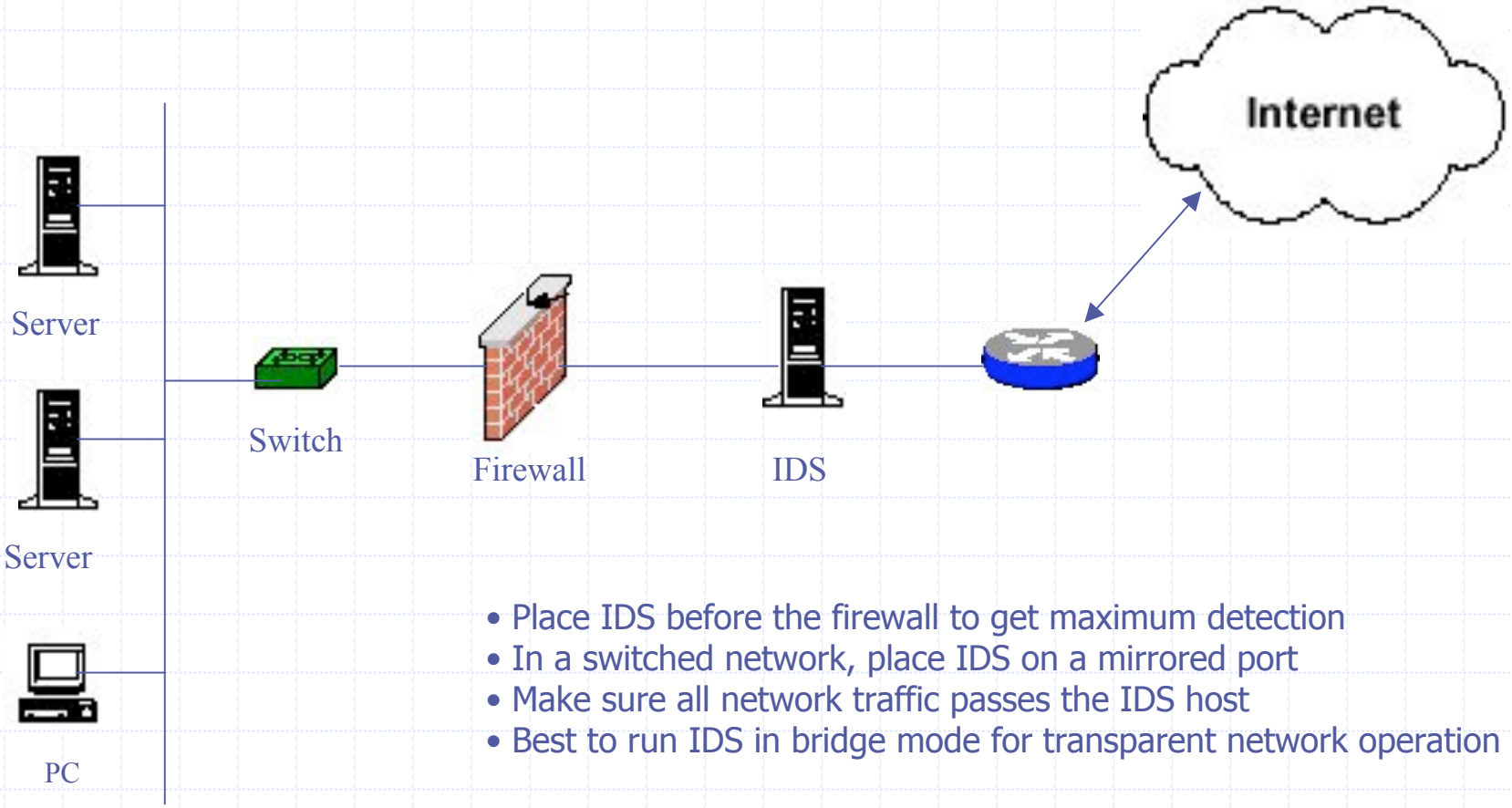
Security Best Practices

- **Consider using the following in conjunction with a firewall:**
- **Intrusion Detection System (IDS)**
  - Intrusion Detection is the art of detecting inappropriate, incorrect, or anomalous activity
  - Inspects/sniffs all network traffic passing thru it for any abnormal content
  - 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
  - Similar to a security "camera" or a "burglar alarm“
  - Alert security personnel that someone is picking the "lock“
  - Alerts security personnel that a Network Invasion maybe in progress

Network Security Applications

Security Best Practices

IDS 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

Network Security Applications

- **Security Best Practices**
  - **IDS products**
  - **Snort** **www.snort.org**
  - **ISS RealSecure** **www.iss.net**
  - **NFR** **www.nfr.com**
  - **PortSentry** **www.psionic.com**

Network Security Applications

Security Best Practices

Hosted-based personal firewall/intrusion-prevention

- A few years ago a user surfing the Internet at home had no worries
- With the increasing use of always-connected cable modems and DSL, the home or small business PC user needs to be aware of security
- Users surfing the Internet without a personal firewall are exposing themselves to serious disaster
- Securing a home / personal computer from Internet hackers has become just as important as securing the corporate workstation
- Home user can be protected from Internet hackers through the use of a personal firewall
- Serious need to protect workstations from malicious traffic

Network Security Applications

Security Best Practices

- **Types of personal firewalls:**
  - Application-based firewall – packet filters block incoming traffic to well-known TCP and UDP ports, while enabling outgoing traffic
  - Another one that performs IP level monitoring; reading data contained in the TCP/IP header for approved protocols and suspicious packet contents - Can trace the source of the attack
  - **Personal firewall products:**

- ZoneAlarm www.zonealarm.com
- Kerio Personal Firewall www.kerio.com
- Norton Internet Security www.symantec.com

Network Security Applications

Security Best Practices

PC

Workstation

Dialup PC

Host security best practices

Although a personal firewall helps in protecting the user against attacks, the following are guidelines that can apply even if there is no firewall installed:

- Have the latest service packs for the Internet browser installed on the PC
- Never run any executables or scripts via e-mail unless the user is sure
- Have the latest service updates for e-mail client software
- Set the file permissions of "normal.dot" in Microsoft Word to read only to prevent viruses or Trojans from affecting the Word setup
- Use a good Antivirus software and make sure to regularly update it
- Regularly scan your PC with Adaware to detect any spyware/trojans/malicious programs

Network Security Applications

Security Best Practices

Server security best practices

WWW

MAIL

DNS

- Run the server on a hardened and routinely patched operating system
- Keep current on software / application updates
- make sure you test these updates in a controlled, non-production environment whenever possible
- one server patch may undo a correction a previous patch applied
- scan the server after the patching up to make sure
- hackers usually attack servers with security bugs that are well known and around for a long time
- Disable file sharing an all critical machines – as it makes them vulnerable to bother information theft and certain types of quick-moving viruses
- Improper sharing configuration can expose critical systems files or give full file system access to any hostile party

Network Security Applications

Security Best Practices

WWW

MAIL

DNS

- **Regularly Scan Systems**
  - Scans will help determine that only the required ports are open
  - Services running on the open ports are not vulnerable to known security bugs/holes
  - Will help you determine if your systems have been compromised – if new open ports are found
  - Perform full port scans using a tool like nmap/ndiff, nessus, fscan on a regular basis
  - Port scans should cover all ports (1-65,535), both UDP and TCP, on all systems:
    - both clients and servers
    - devices such as routers, switches, printers
    - and anything else connected (physically through wire or wireless) to your network

Network Security Applications

- **Security Best Practices**
  - **Host / Network scanning software**

Nmap/Ndiff	**www.nmap.org**
Nessus	**www.nessus.org**
Fscan	**www.foundstone.com**
Satan	**www.fish.com/satan/**

Network Security Applications

Security Best Practices

IDS

FW

Logger

- **Effective/secure user accounts management**
  - Remove all unnecessary accounts
  - Simply disabling an account is not sufficient to guard against an intruder abusing it
  - Privileged accounts (administrators, power users, executive staff) are very dangerous
  - Rename Default Administrative Accounts
  - It is trivial to identify the actual Administrator account, but then why make it easy for them?
  - Renaming the default Administrator accounts may not slow down a moderately skilled attacker
    - will defeat most of the automated tools and techniques used by less skilled attackers
    - who make the assumption your system is using default account names
    - Purpose is to keep the intruders guessing, at least!

Network Security Applications

Security Best Practices

- **Password Policies**
  - While there are promising technologies on the horizon that could replace passwords as a method of authenticating clients, at present we are reliant on passwords
  - Use secure authentication like PKI, digital certificates, ssh, etc.
  - A password policy should define the required characteristics of accepted passwords for each system:
    - Minimum length
    - Composition; alpha, upper or lower case, numeric, special
    - Effective life
    - Uniqueness (how often a password can be reused)
    - Lockout properties; under what conditions, and for how long

    - These characteristics differ from system to system because each has different capabilities

Network Security Applications

Security Best Practices

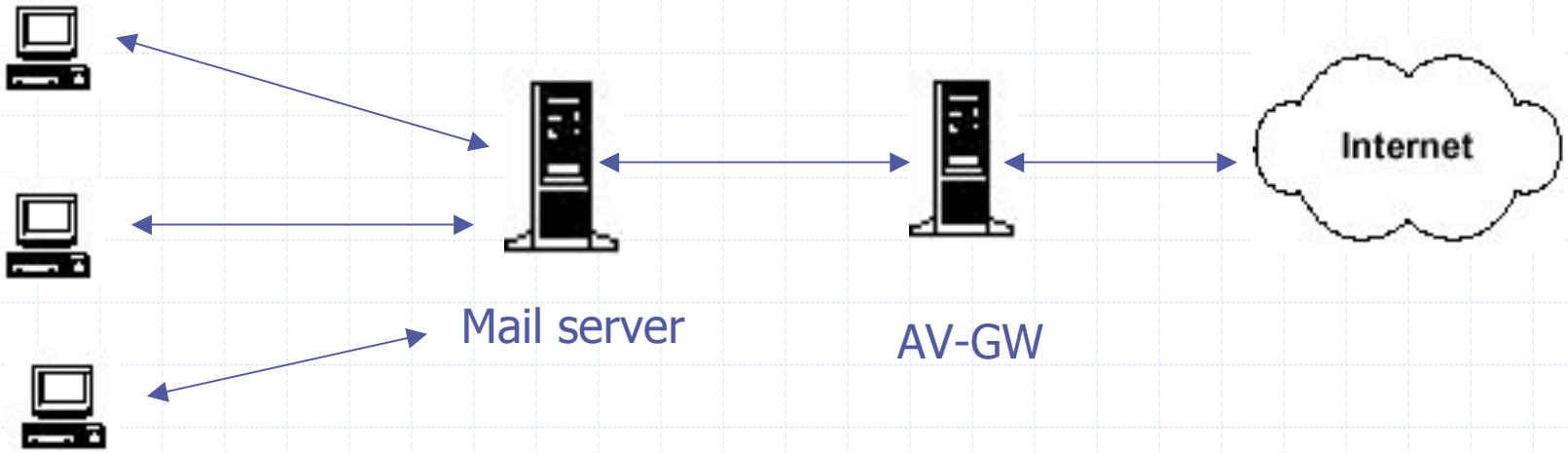
Name Servers and Workstations Securely

- Host name alone can advertise to a potential attacker a host's primary service or purpose and how important *you* consider the host to be
- Database servers are named db1, db2, sql.xyz.com
- Mail servers are named mail.xyz.com, smtp.abc.com, mx.klm.com
- DNS servers have names like ns.abc.com, ns2.xyz.com
- Follow a very generic naming conventions – name of mountains
- Do not to reveal any host related services from the host name that lessens the guess work for possible intruders
- Do not name boxes for the people who primarily use them
  - provides a "directory" of executives, administrators, and other users likely to have privileged rights on the network
  - executives are people who demand excessive privilege, user-friendliness and convenience over security

Network Security Applications

Security Best Practices

- Anti-Virus Systems
  - Install anti-virus protection systems at key points – file servers, post offices (inbound/outbound email and attachments), end-user workstations
  - Of critical importance, keep them current!
  - Viruses that quietly, skillfully, and effectively alters the victim system, allowing an intruder privileged backdoor access are of greater concern



Network Security Applications

Security Best Practices

Enable and Monitor Logging and Auditing on a 24x7 basis

IDS

FW

Logger

- "Prevention is ideal, but detection is a must"
- We must realize that “No prevention technique is full-proof”
- New vulnerabilities are discovered every week that you may not be aware of
- Constant vigilance is required to detect new unknown attacks
- Once you are attacked, without logs, you have little chance of finding what the attackers did
- You can not detect an attack if you do not know what is occurring on your network
- Logs provide the details of what is occurring, what systems are being attacked, and what systems have been compromised
- If any log entries that don't look right, and investigate them immediately

Q & A

Exercise

- Create a Security Policy template for your organization
  - Hints:
    - Document your existing Network, Insert Security policies in the current network
    - Look at it from a planning perspective and put additional features/policies in place

Introduction to Linux/BSD environment

Gaurab Raj Upadhaya

Open Source Infrastructure for Networks

Why Linux?

- Linux is now the official GNU OS
- All the benefits of open source
- One of the most popular and increasing user base
- Many resources on the web specially made for Networks using Linux
- You may choose to use NetBSD, FreeBSD if they suit your needs.

Open Source Infrastructure for Networks

What is Linux ?

- Linux Distributions : Pre- Packaged Linux Kernel with host of other useful software and utilities, package managers, installation routine etc.

RedHat - A Major Linux Distribution

Some other distributions are SuSe, Mandrake, Debian, Caldera, Slackware.

We choose RedHat (RH for short), because we are most familiar with it.

Open Source Infrastructure for Networks

Useful Web sites:

- www.linuxdoc.org
- www.redhat.com
- www.freshmeat.net
- www.linuxisp.net
- www.linux.org
- www.chuvakin.org/ispdocs
- www.unesco.org/webworld/portal_freesoft/index.shtml
- www.sdnp.undp.org/observatory/ -- click “open source” theme

Open Source Infrastructure for Networks

Linux Essentials

- Linux Kernel
- File System
- Directory Structure
  - Filesystem Hierarchy Standard
  - The root filesystem
  - the /var, /usr, /home, filesystems
  - Mounting
- Device Files
- Run Levels
- Shell
- Operating System Structure in Linux
- Important Files

Open Source Infrastructure for Networks

Linux Kernel

- Linux Kernel is the most important part of the operating system. The current linux kernel version is 2.4.xx
- The official kernel website is www.kernel.org

- Linux kernel provides the basis for the operating system i.e, services like memory management, networking code, i/o management, etc.

- Networks may need to re-compile the kernel based on the use they are putting it to.

Open Source Infrastructure for Networks

File System

- In Unix “every thing is a file”
- Files form the basis of all operation
- The current file system (for RH 7.2) is ext3 (extended file system 3). With RH 7.1 it is ext2
- ext3 provides a journaling file system, which vastly improves recovery of corrupt file systems

Open Source Infrastructure for Networks

Directory Structure

- The file system hierarchy standard.
- The root file system (and the root directory) is the most important
- /usr is used for installing programs
- /home is used for home directories
- /var is used for variable data - I.e, DNS files, web files, mail spool, printer spool, ftp server etc..

Open Source Infrastructure for Networks

File System or Directory

- Basically, for the system it doesn't matter.
- A directory may reside on the same filesystem or may be on a different file system
- for e.g, a CD-ROM in ISO 9660 may reside within the /var/ftp, so that the ftp service is provided directly off the CD-ROM.
- In this case, the CD-ROM was mounted.

Open Source Infrastructure for Networks

Device Files

- Device file are in the special directory /dev
- device files are block devices or serial (character) devices.

- Common devices

IDE - Primary Master /dev/had
Secondary Master /dev/hdc

Serial Port /dev/ttyS0, and ttyS1
Ethernet Card /dev/eth0, eth1

Open Source Infrastructure for Networks

Run Levels

- The run level species what programs are being run at a particular time.. This is fully customizable, but there are best practices (and defaults)

0	Halt the system
1	Single user (or maintenance) mode
2-5	Customizable
	3 Normal Default Mode
	5 Normal with Graphics Mode
6	Reboot

Open Source Infrastructure for Networks

Shell

- The shell provides a gateway between the users and the linux kernel.
- The default shell for each user is specified in the /etc/passwd file.
- The most common shell on Linux systems is Bourne shell again (bash)

- The shell can be changed by giving the command chsh

Open Source Infrastructure for Networks

Operating System structure

System Software User Applications

Shell

Linux Kernel

Hardware

In Linux it is important to understand WHY and WHAT before HOW !!!

Open Source Infrastructure for Networks

Important Files

- /etc -- has all the configuration files
  - /etc/rc.d/ startup scripts
  - /etc/sysconfig/ system configuration
  - /etc/inittab init file
- /var/ -- has all the logs and data files for web, ftp, mail and dns servers
- /home has all the user data

Open Source Infrastructure for Networks

Installation

- The best and easiest is to use a CD-ROM/DVD
- You can also start install using a diskette
- Network install is recommended if you have to do frequent installs / upgrades
- Latest edition may always not be the best edition

Open Source Infrastructure for Networks

Pre-Installation Checklist

- Model and Make of display card, monitor and network card
- IRQ / IO of the network card
- Plan for directory hierarchy and filesystem (including swap)
- Hostname, domain name and IP address
- Gateway IP address
- DNS address (if using external DNS)
- List of services needed

- It is actually easier to install in the beginning rather then adding it later

Open Source Infrastructure for Networks

Booting up

- The boot up process follows the following order
  - The BIOS of the computer hands over the bootup to LILO (Linux Loader), which is the most common Boot Loader
  - LILO loads the Kernel Image
  - The Kernel Image loads and detects all the hardware, loads the file systems as specified in /etc/fstab. Creates the virtual file system /proc
  - The init process takes over, and runs the default run-level scripts
  - Most common run-level is 3, but if Xdm is used, it's 5
  - The login prompt is provided after all scripts are run.

Open Source Infrastructure for Networks

Moving Around in Linux

- Using Virtual Consoles in command Mode
  - (using alt+F1…F6)
- Using multiple terminals in X and multiple Desktops in the WM

Login Process

- Username and passwords are verified against /etc/passwd
- user shell is created and shell scripts run

Open Source Infrastructure for Networks

Simple Commands

- Not possible to list all of them here
- Most commands are in
  - /bin
  - /usr/bin
  - /usr/local/bin
- Super user commands are in
  - /sbin
  - /usr/sbin
  - /usr/local/sbin

Open Source Infrastructure for Networks

Basic ISP Operation using Linux Networking services

- Xinetd and inetd -- the Internet super server
- TCP Wrappers and basic Host security
- Changing IP address
  - Configuration of multiple NIC cards.
  - Assigning multiple IP addresses to NIC cards
- Using Modems
  - Configuring Dial-in Services
  - Multi-port Serial Cards
- User Management

Open Source Infrastructure for Networks

Xinetd and inetd - the internet super server

inetd has been replaced by xinetd (RH 7.1 and 2 come with xinetd)

/etc/inetd.conf has been replaced by /etc/xinetd.d

All TCP/IP network services are controlled through this server.

The startup script is /etc/rc.d/init.d/xinetd

Open Source Infrastructure for Networks

Sample xinetd service configuration file

- service smtp

```
{
        flags           = REUSE NAMEINARGS
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = qmaild
        server          = /usr/sbin/tcpd
        server_args     = ../qmail/bin/tcp-env -R ../bin/qmail-smtpd
}
```

Open Source Infrastructure for Networks

TCP Wrappers and basic Host security

- Xinetd also provides a wrapper service, commonly called as TCP wrappers
- Host security implemented through
  - /etc/host.allow
  - /etc/host.deny
- Sample of host.allow file

Service name	IP address / hostname	
telnetd:	192.168.1.	#denies (or allows) telnet access to all hosts in the 192.168.1 network
ftpd:	ALL except 192.168.1.	

Open Source Infrastructure for Networks

Setting up and changing IP address

- Using Ifconfig
- Editing the scripts and restarting service
  - /etc/sysconfig/network
  - /etc/sysconfig/network-scripts/ifcfg-eth0
  - /etc/hosts
  - You may have to edit /etc/modules.conf in order to get your network card detected

Open Source Infrastructure for Networks

Assigning Multiple Address to NIC cards

Examples

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
IPADDR=192.168.0.1
NETMASK=255.255.255.0
NETWORK=192.168.0.0
BROADCAST=255.255.255.255
GATEWAY=192.168.0.1
ONBOOT=yes
```

/etc/sysconfig/network-scripts/ifcfg-eth0:0

```
DEVICE=eth0:0                (assuming your first network card is eth0)
IPADDR=192.168.0.20
NETMASK=255.255.255.0
NETWORK=192.168.0.0
BROADCAST=255.255.255.255
GATEWAY=192.168.0.1
ONBOOT=yes
```

Open Source Infrastructure for Networks

Modems

- Modems do not require drivers
- Directly access the ttyS0 and ttyS1 for modem access
- Win-modems (most internal modems these days) are a problem, but can still be used

- Mutliport serial cards, when installed, simply show additional number of serial ports (e.g, ttyX0… X15)

Open Source Infrastructure for Networks

User Management

- adduser, passwd and finger are the most common command
- deluser is for deleting users
- default configuration is stored at /etc/skel
- Most Networks use customized scripts to add users

Open Source Infrastructure for Networks

Questions and Answers

Applications :SSH

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

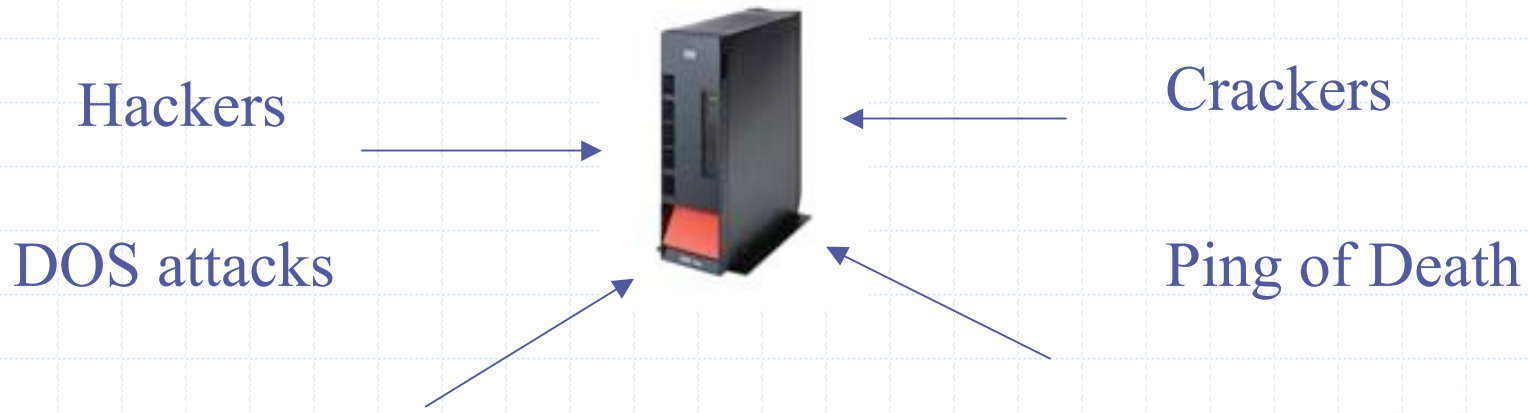
- What kinds of attacks does SSH protect against ?
  - Eavesdropping a transmission - looking for passwords, credit card numbers, or business secrets
  - Hijacking - taking over a communication and redirect
  - Interception of communication between two systems - inspect or modify any data being transmitted 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

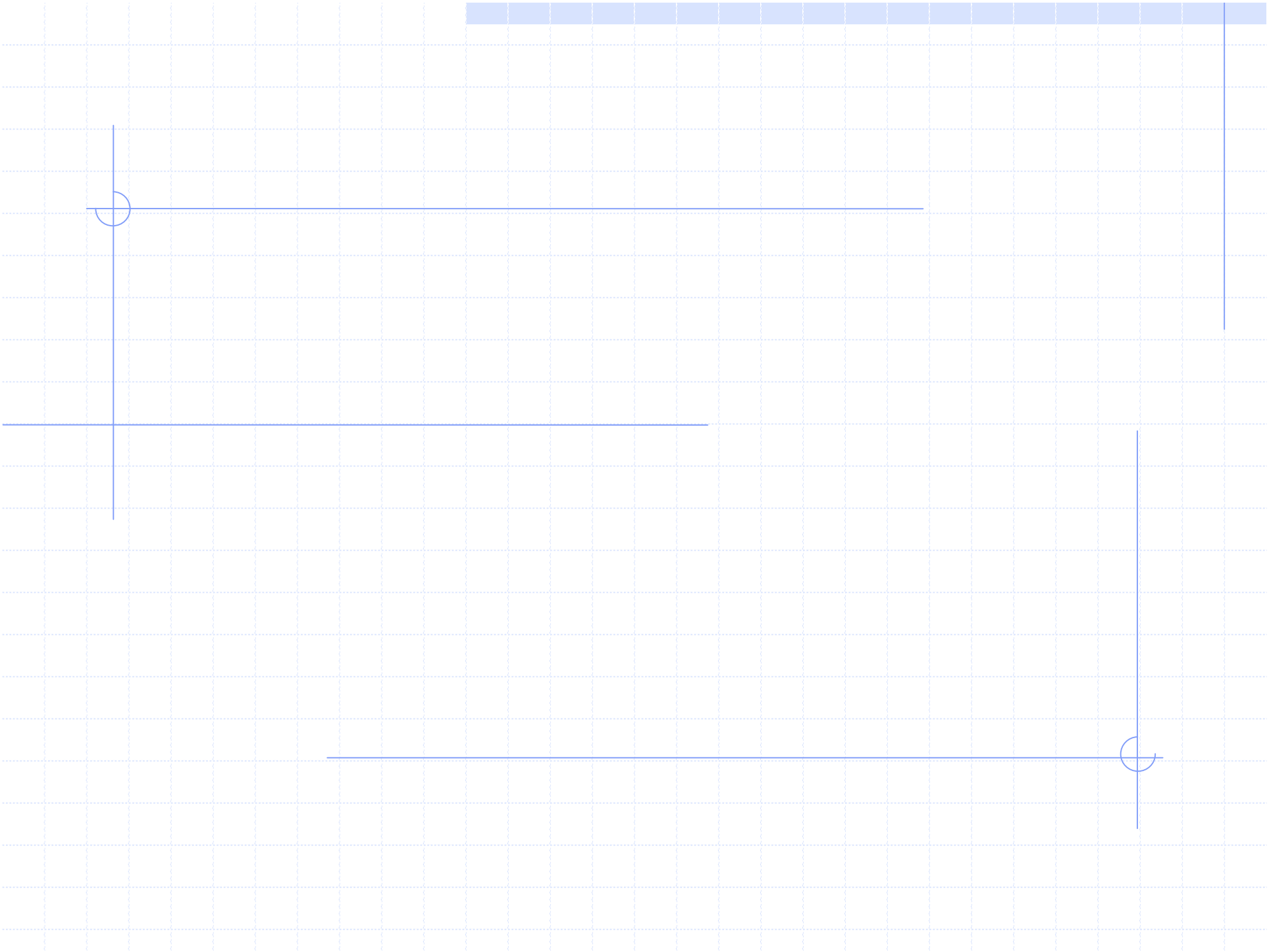
- Configuration data is parsed as follows:
  - 1. command line options
  - 2. user-specific file
  - 3. system-wide file

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 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



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

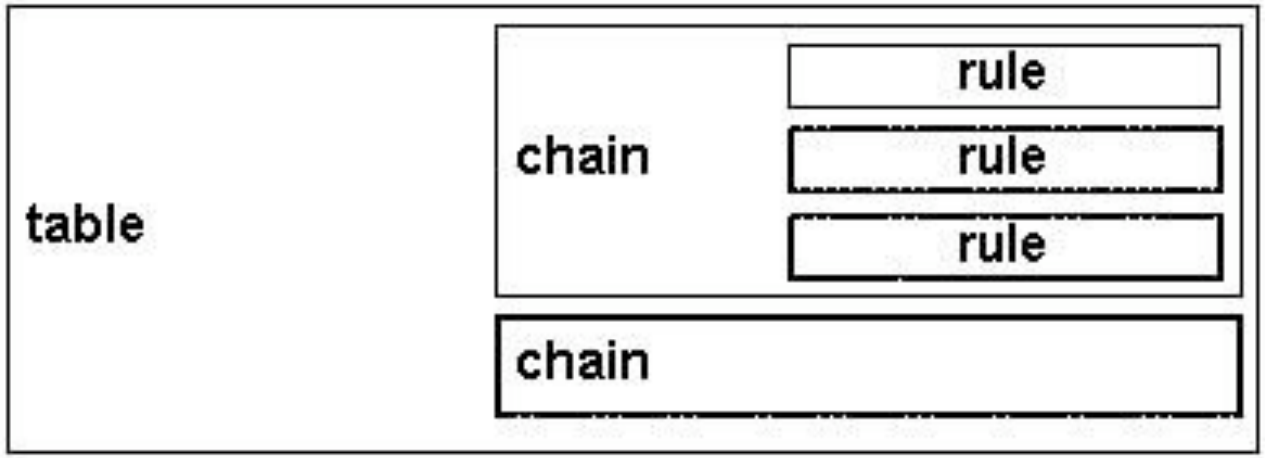
So What's A Packet Filter?

- 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
- Chains – collection of rules
- Table – collection of chains

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

IPTables “tables”

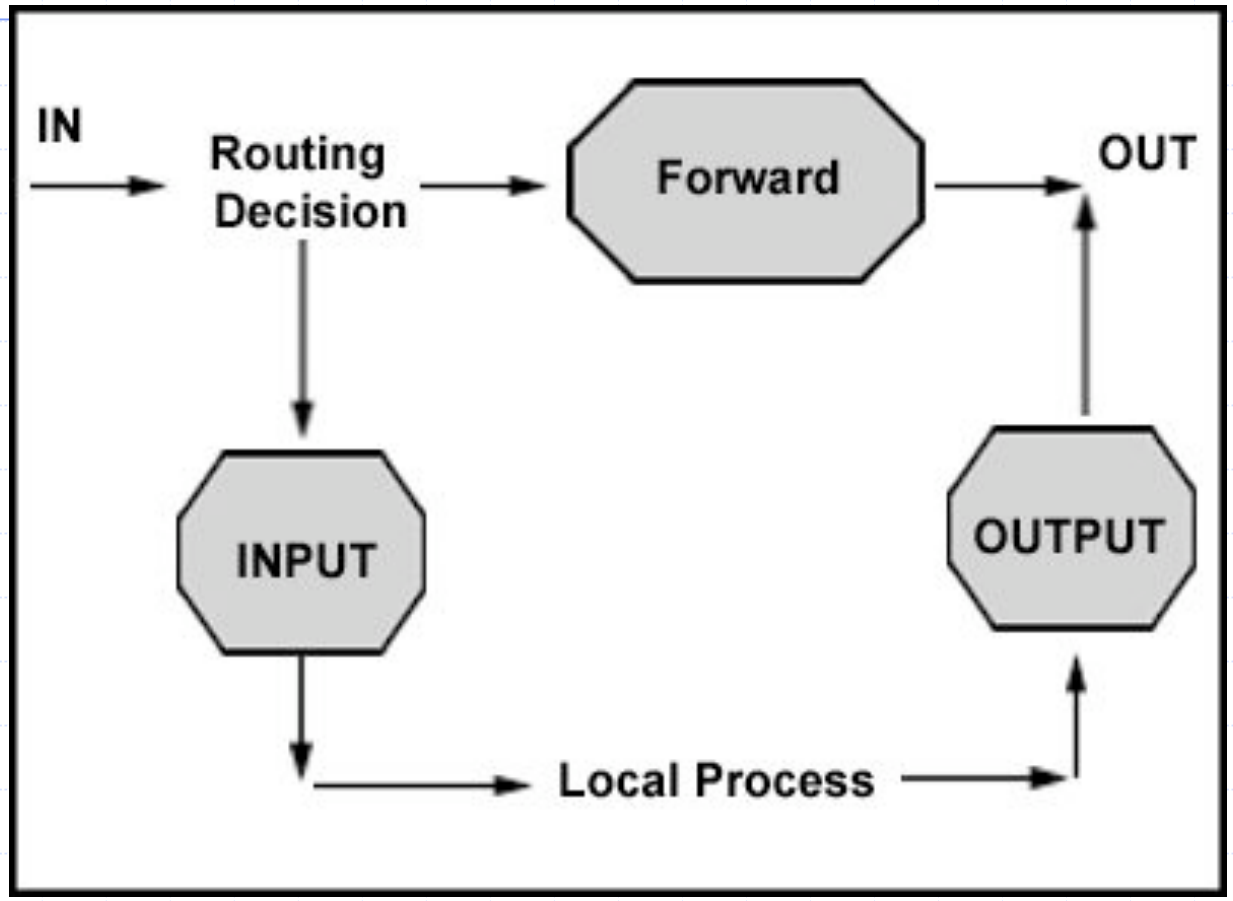
IPTables has three tables:

- 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

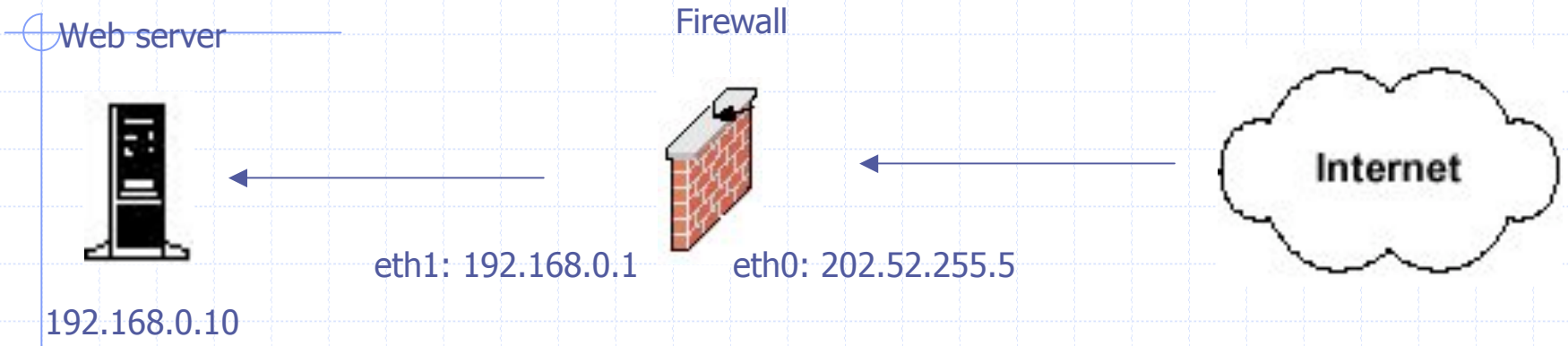
To NAT all outbound traffic with the IP of eth0:

```
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)

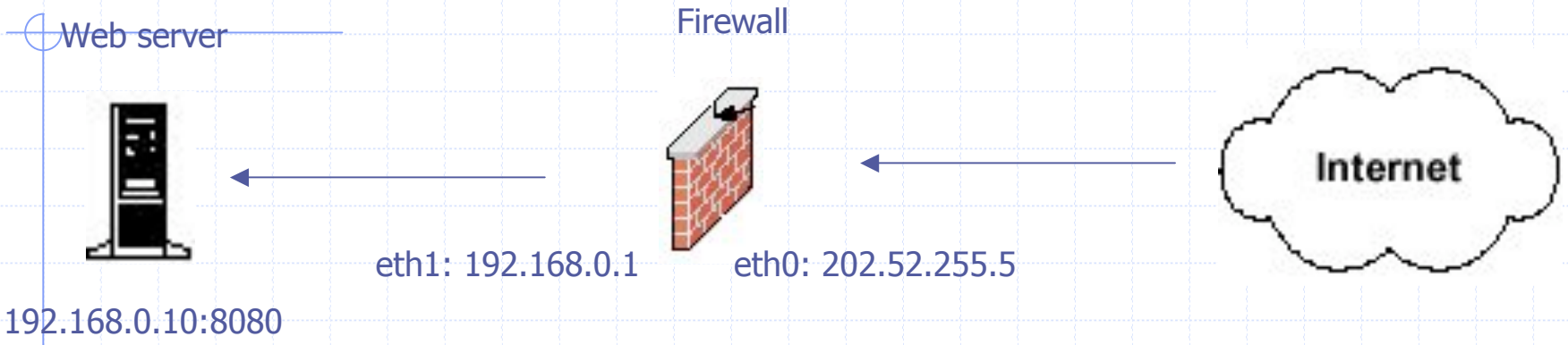


Makes it possible to hide internal server IP from the Internet

```
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)

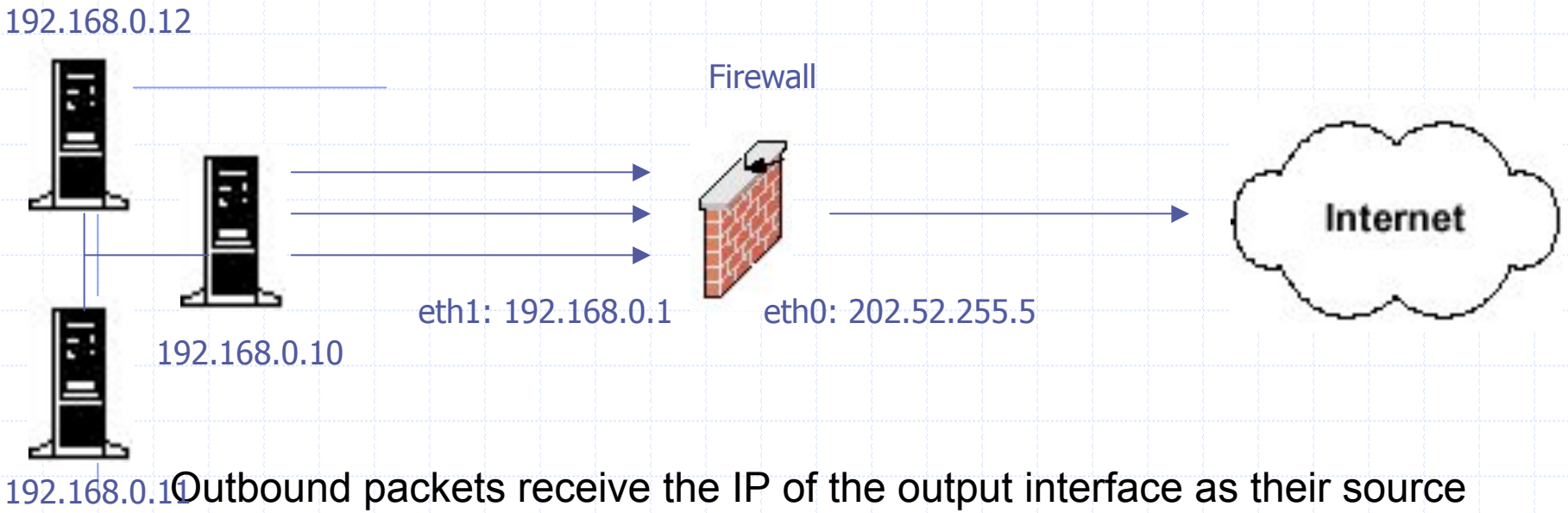


Entails the modification of the destination port and enables clients to access a service via destination port other than that on which service listens.

```
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)

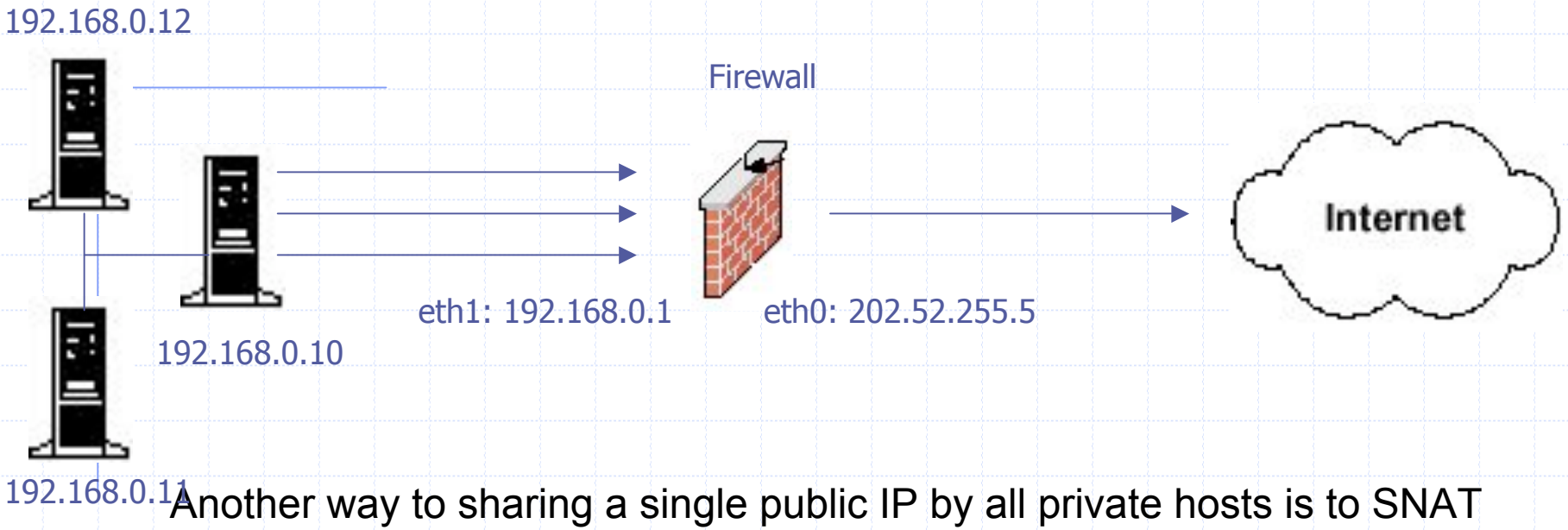


Outbound packets receive the IP of the output interface as their source address. It is useful when there is no fixed IP addresses of output interface.

```
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)

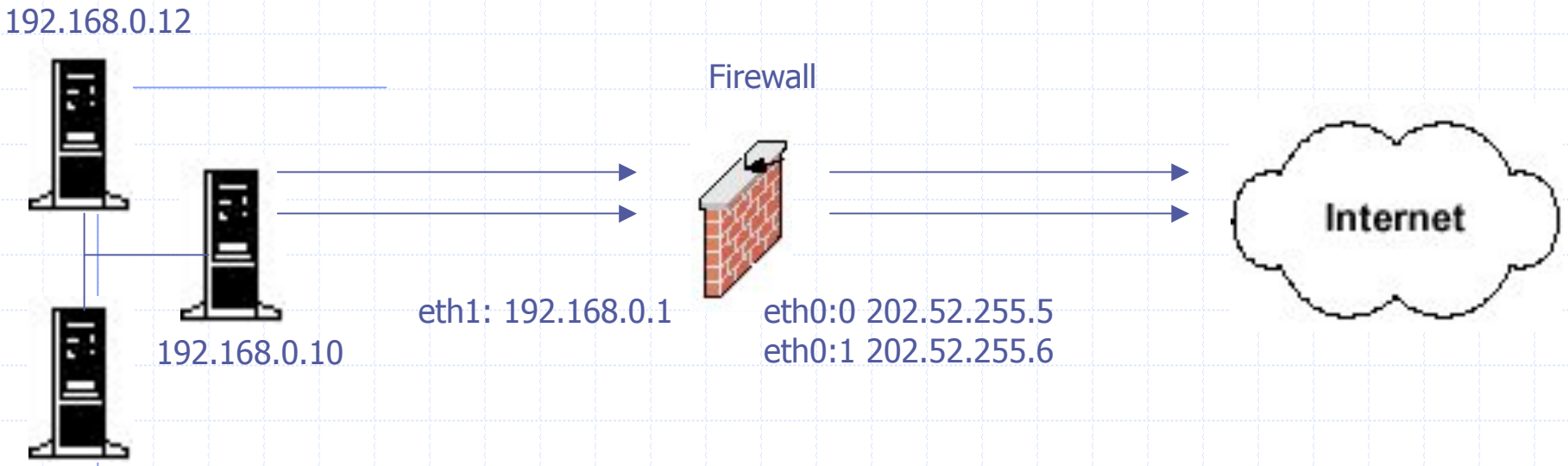


Another way to sharing a single public IP by all private hosts is to SNAT (Source NAT) it.

```
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)



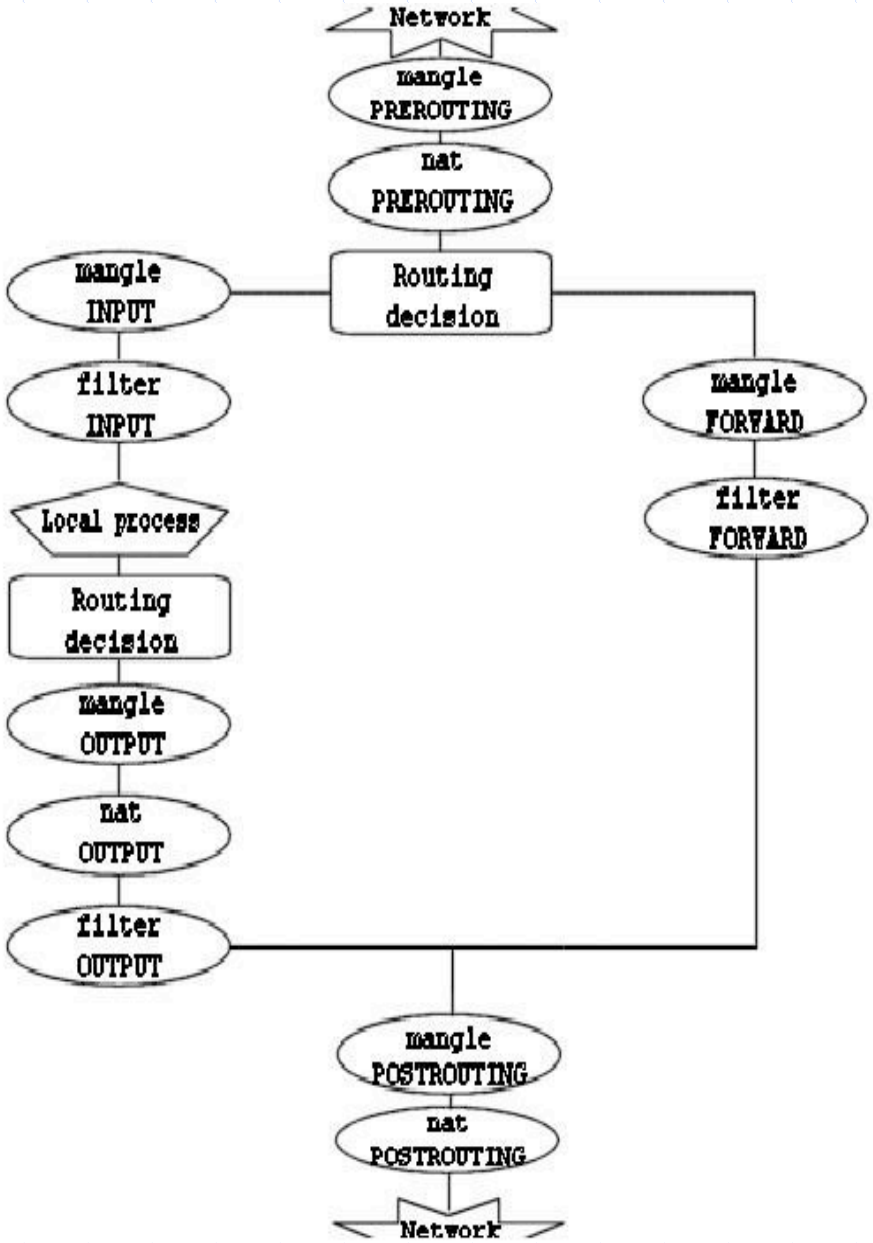

```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.11
-j SNAT --to-source 202.52.255.5

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

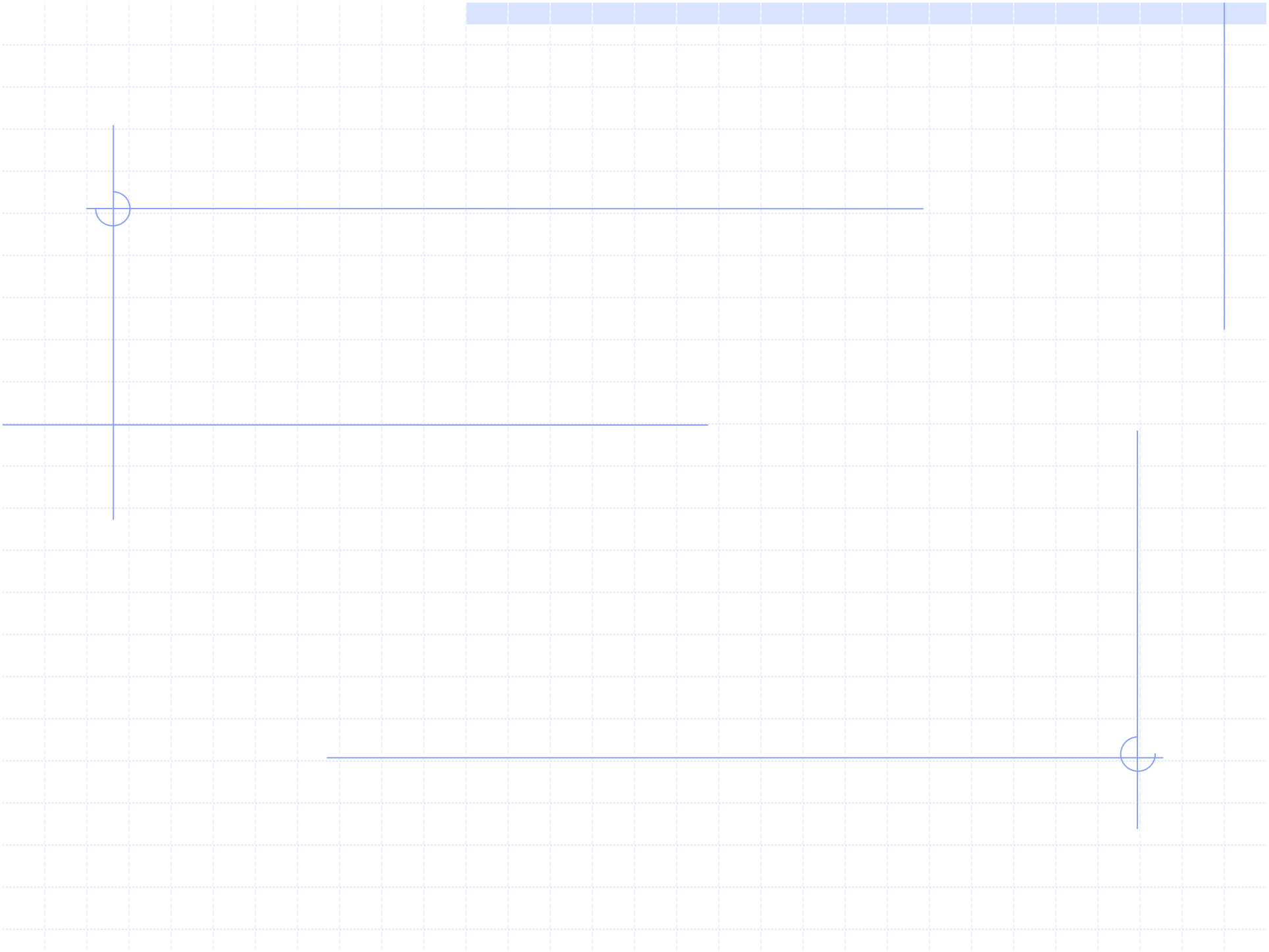
The source IP of 192.168.10.12 will be converted to 202.52.255.6

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



Security Workshop

- Nmap www.insecure.org/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

Security Workshop

- 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 code, GNU GPL
- Well 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

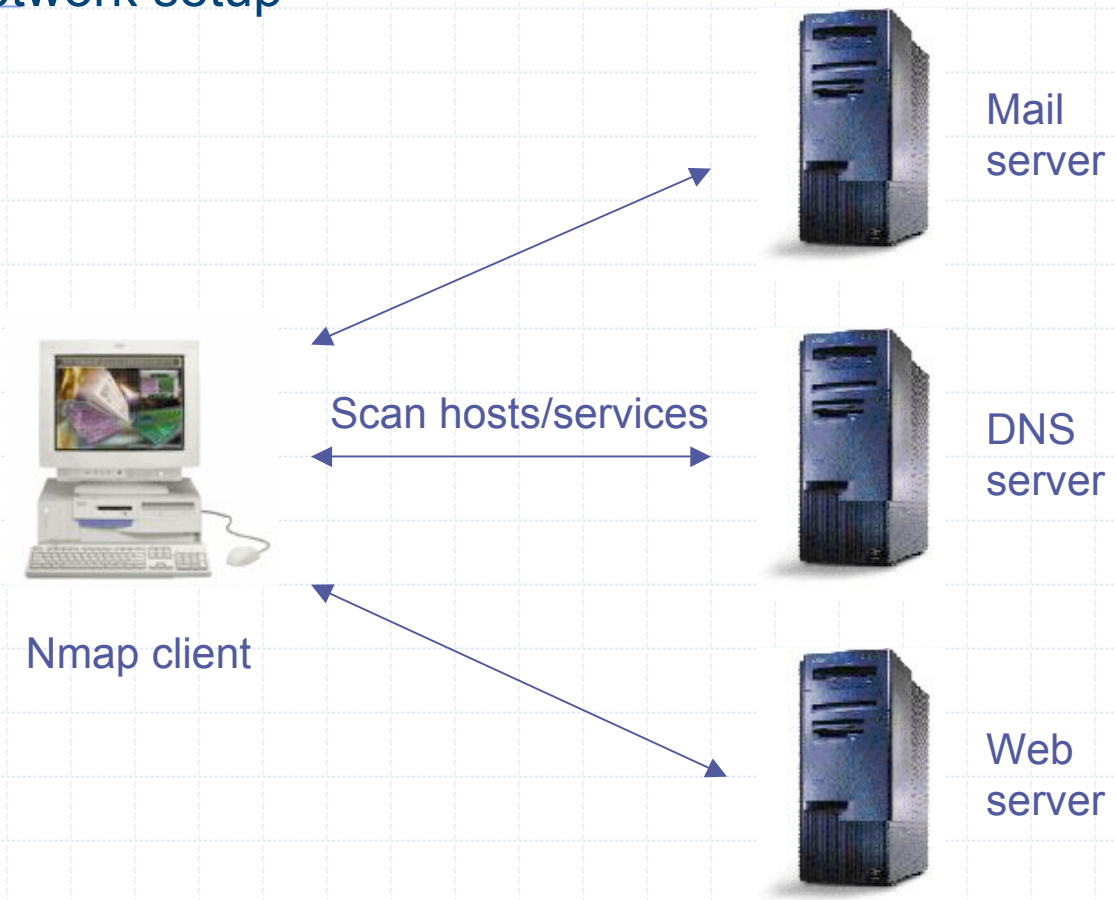
2 - portscan hosts to determine what services are listening

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

Security Workshop

- Nmap network setup



Security Workshop

- 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…

Using 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

Using 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

Security Workshop

- 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 www.vinecorp.com/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

Security Workshop

- 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

Security Workshop

- Nessus www.nessus.org
  - 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

Security Workshop

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

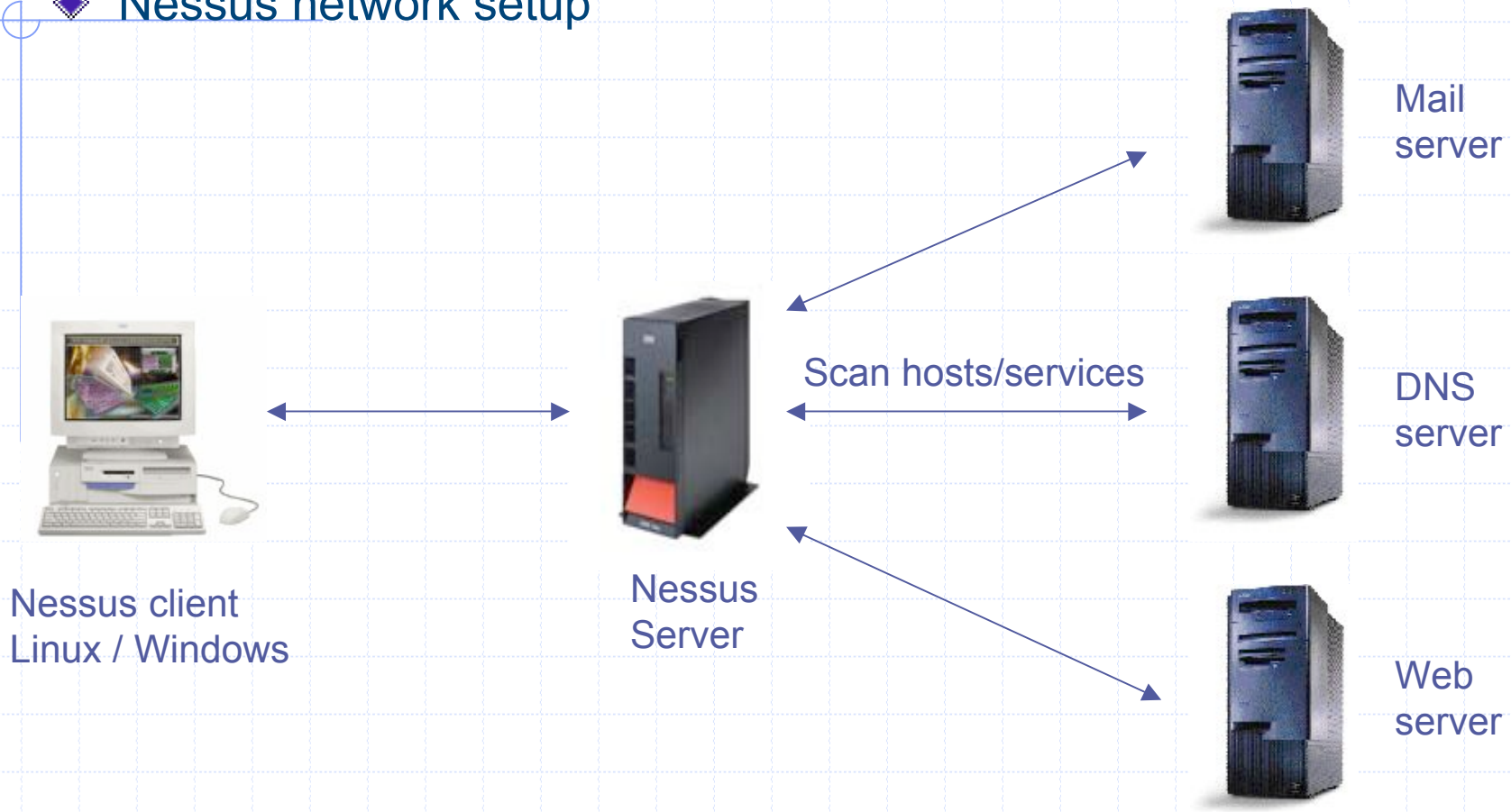
Security Workshop

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

Security Workshop

- Nessus network setup



Security Workshop

- Snort www.snort.org

  - **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

Security Workshop

Snort Features

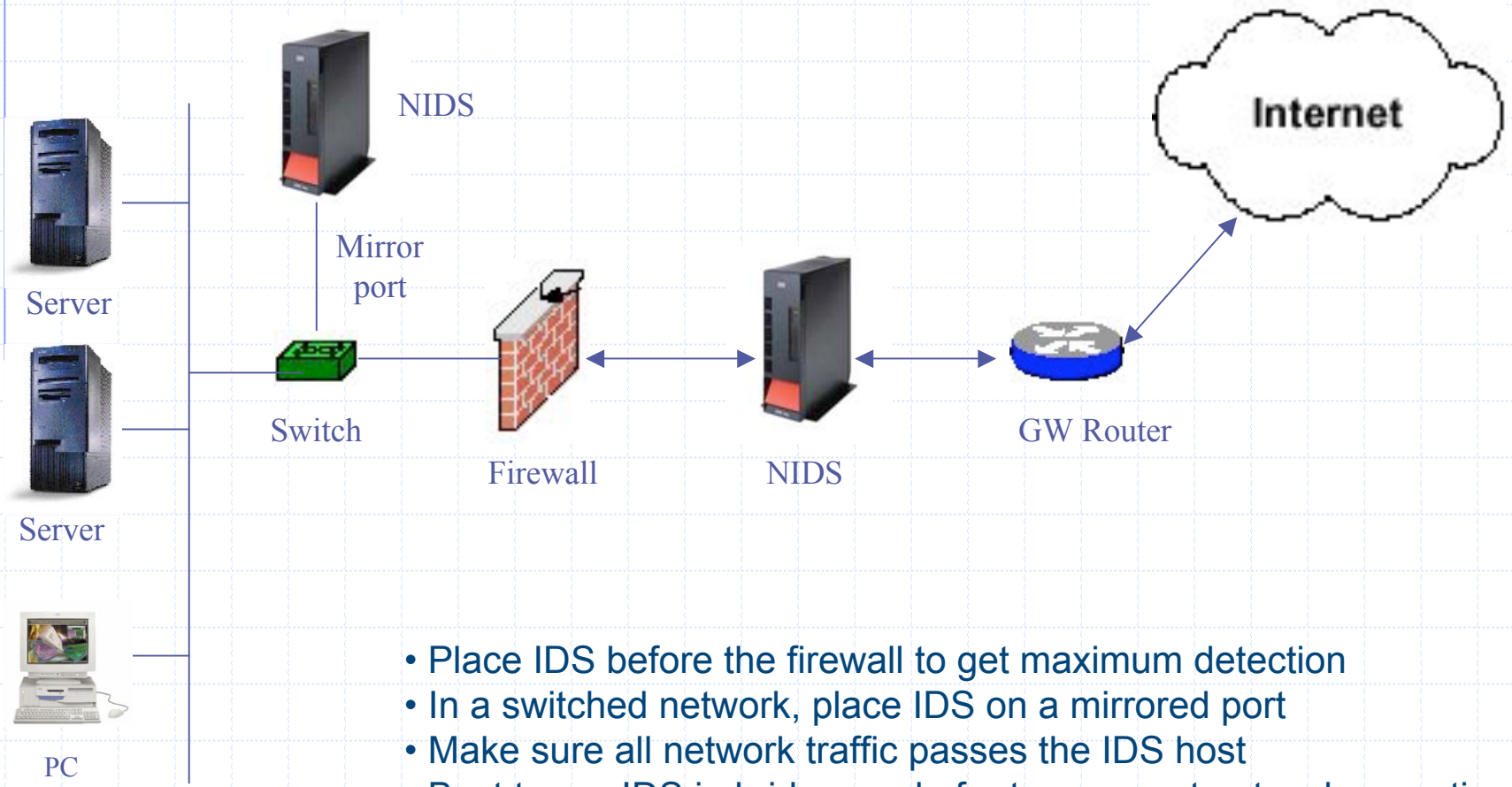
- a cross-platform, lightweight network intrusion detection tool
- rules based logging to perform content pattern matching
- detect a variety of attacks and probes
- buffer overflows [ALE96], stealth port scans, CGI attacks, SMB probes, and much more
- has real-time alerting capability - syslog, SMB "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

Security Workshop

- 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

Security Workshop

- 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

Security Workshop

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

Security Workshop

- 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

Security Workshop

- 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

Security Workshop

- 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

Security Workshop

Snort Architecture

3. The logging/alerting subsystem

- 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

Security Workshop

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

Security Workshop

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.