Hacking and Securing Linux

- **Craig Ozancin**
- **Senior Security Analyst**
- **Symantec Corporation**
- **cozancin@symantec.com**

August 27, 2002

Agenda

- **Who is who**
- **The threat**
- **The solution**
- **Where can I find more information**
- **Conclusion**
- **Questions?**

I: Who Is Who?

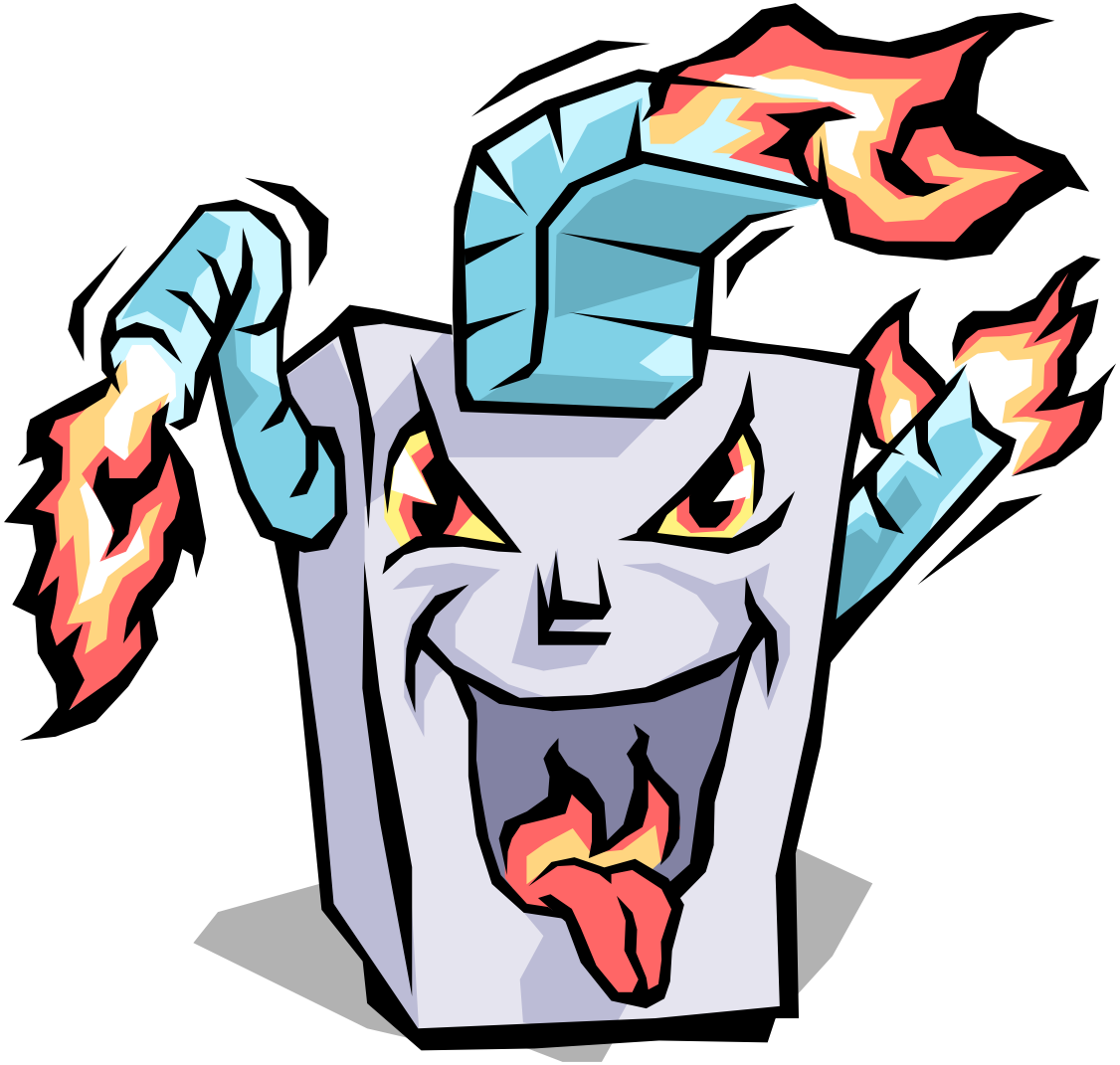
Who Is Who?

- **Hackers**
- **Crackers**
- **Script kiddies**
- **Social engineer**
- **Phone Phreaks**
- **Packet monkeys**

- **White hat hacker**
- **Black hat hacker**
- **Criminal**
- **The kid next door?**

Attackers

II: The Threat



The Threat

- **Steps to breaking in**
- **Scanning**
- **Getting and keeping control**
- **Covering your tracks**
- **Extend the attack**
- **Denial-of-service**
- **Worms**

Steps to breaking in

Common Steps of an Attack

- **Identify target**
  - **Pick one**
  - **Scan**
  - **Random**
  - **Link from another location**
- **Find more information**
  - **Research / footprint**
  - **Scan**
- **Identify way in and use it**
  - **Identify vulnerability**
    - Password cracker
    - Buffer overflow
    - Configuration flaw
    - Many others
  - **Exploit it**
- **Elevate privilege (if necessary)**
- **Remove evidence of exploit**
  - **Logs**
  - **Intrusion detection systems**
- **Explore, look for new targets or abuse**
  - **Network sniffing**
  - **Steal content**
  - **Deface website**
  - **Backdoor**
  - **Destroy system**
  - **Others**

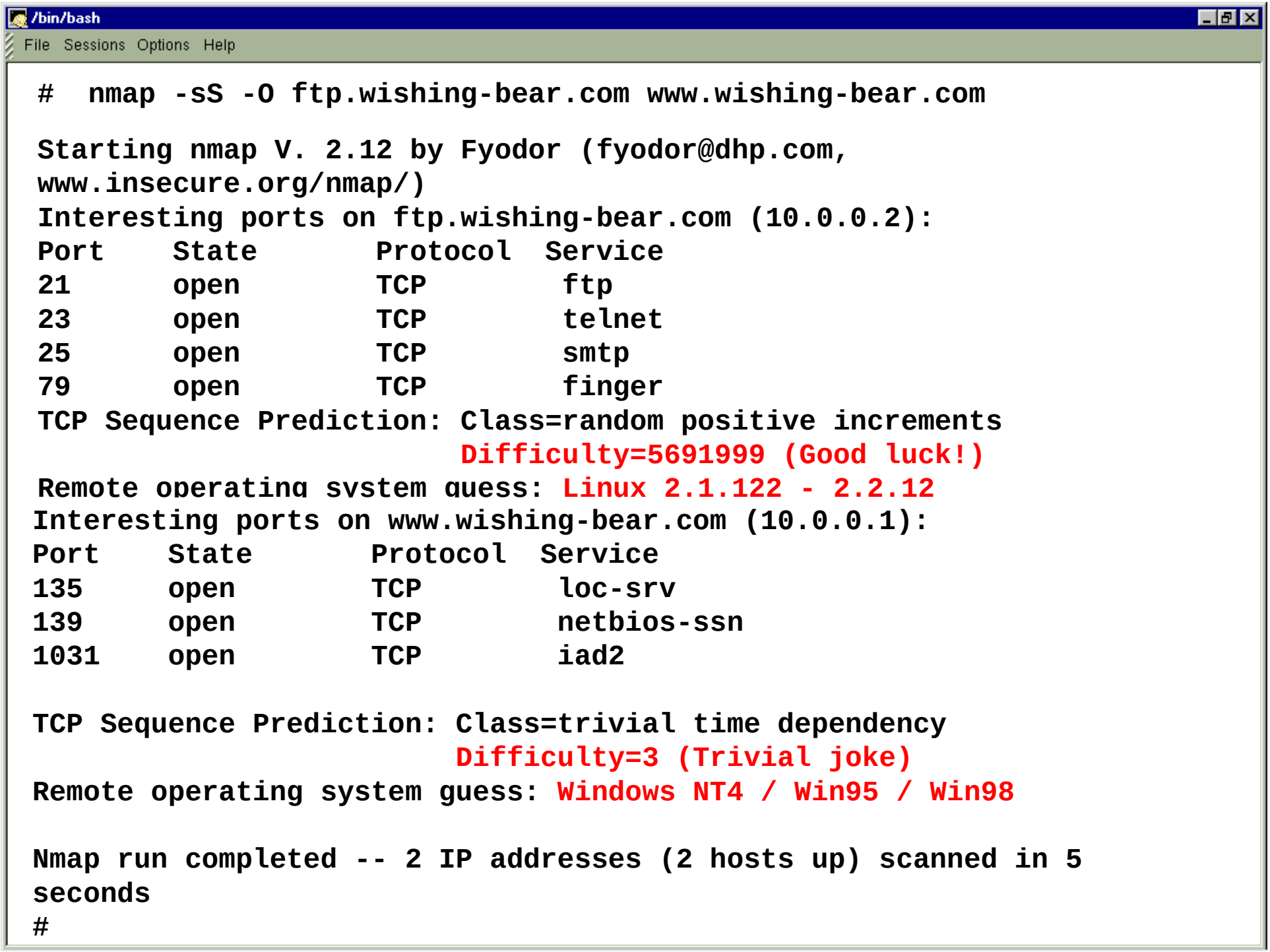
Scanning

Scanning

- **Port scanning**
  - **Acquires accessible port information from remote systems**
  - **Operating system discovery**
- **Look for specific vulnerable services**
- **Dialup modems (war dialing)**
- **Wireless networks (war driving)**
- **Firewall rule discovery**

Port Scanning

- **Acquires accessible port information from remote systems**
- **This information can be used to identify potentially vulnerable services**
- **Some popular port scanners are:**
  - **Strobe**
    - Attempts to open ports and report success
  - **Nmap**
    - Can be used to gather extensive network mapping of a network
    - Adds the concept of stealth scanning
    - Operating system type and version discovery
    - Identifies both open TCP and UDP ports
  - **Cheops**
    - Similar to strobe and nmap but creates graphical network maps
    - Also identifies SNMP services and allows user to send requests

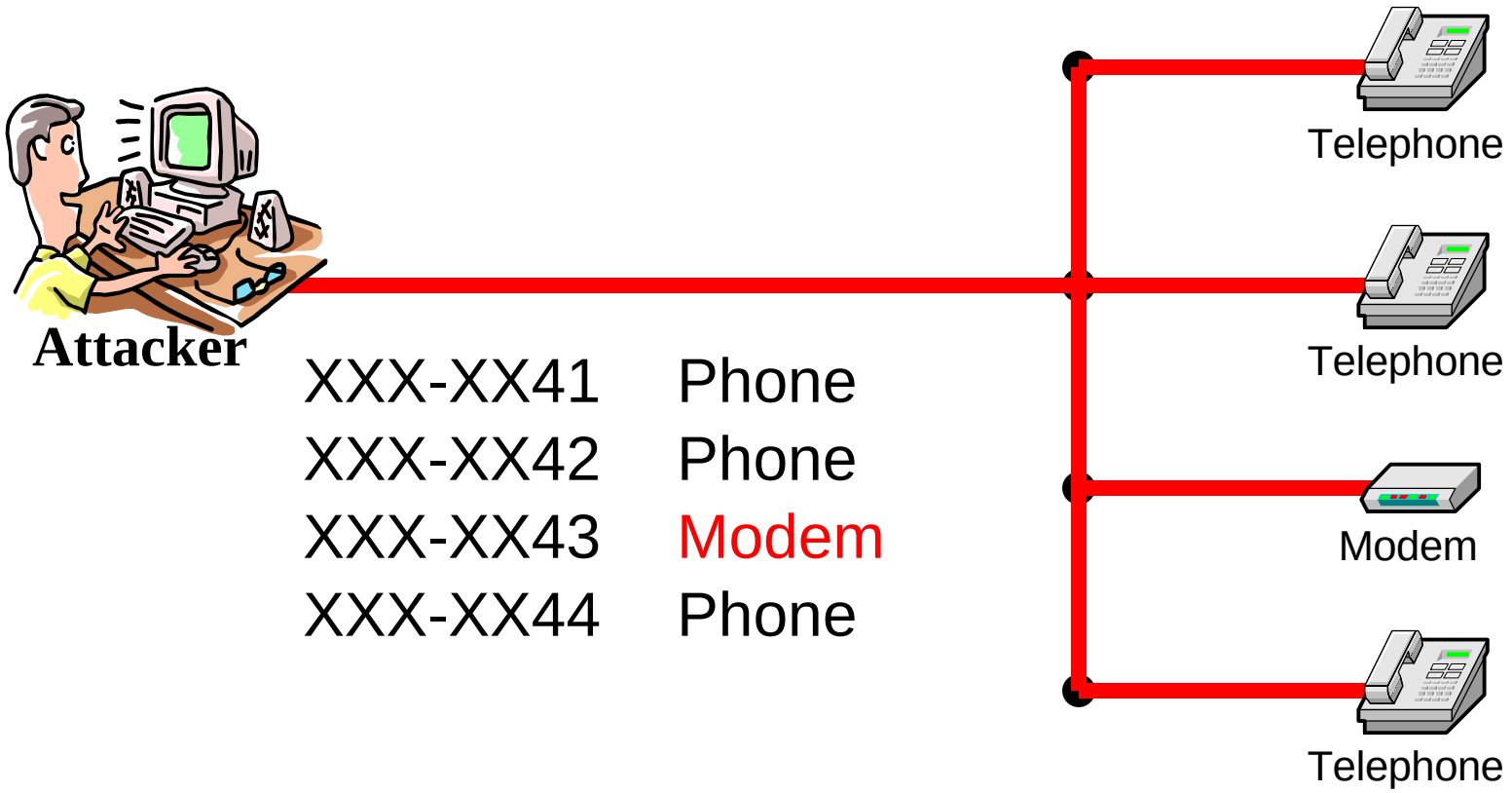

/bin/bash
File Sessions Options Help
nmap -sS -O ftp.wishing-bear.com www.wishing-bear.com
Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Interesting ports on ftp.wishing-bear.com (10.0.0.2):
Port State Protocol Service
21 open TCP ftp
23 open TCP telnet
25 open TCP smtp
79 open TCP finger
TCP Sequence Prediction: Class=random positive increments
Difficulty=5691999 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.12
Interesting ports on www.wishing-bear.com (10.0.0.1):
Port State Protocol Service
135 open TCP loc-srv
139 open TCP netbios-ssn
1031 open TCP iad2
TCP Sequence Prediction: Class=trivial time dependency
Difficulty=3 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98
Nmap run completed -- 2 IP addresses (2 hosts up) scanned in 5 seconds
#

Single Port Scans

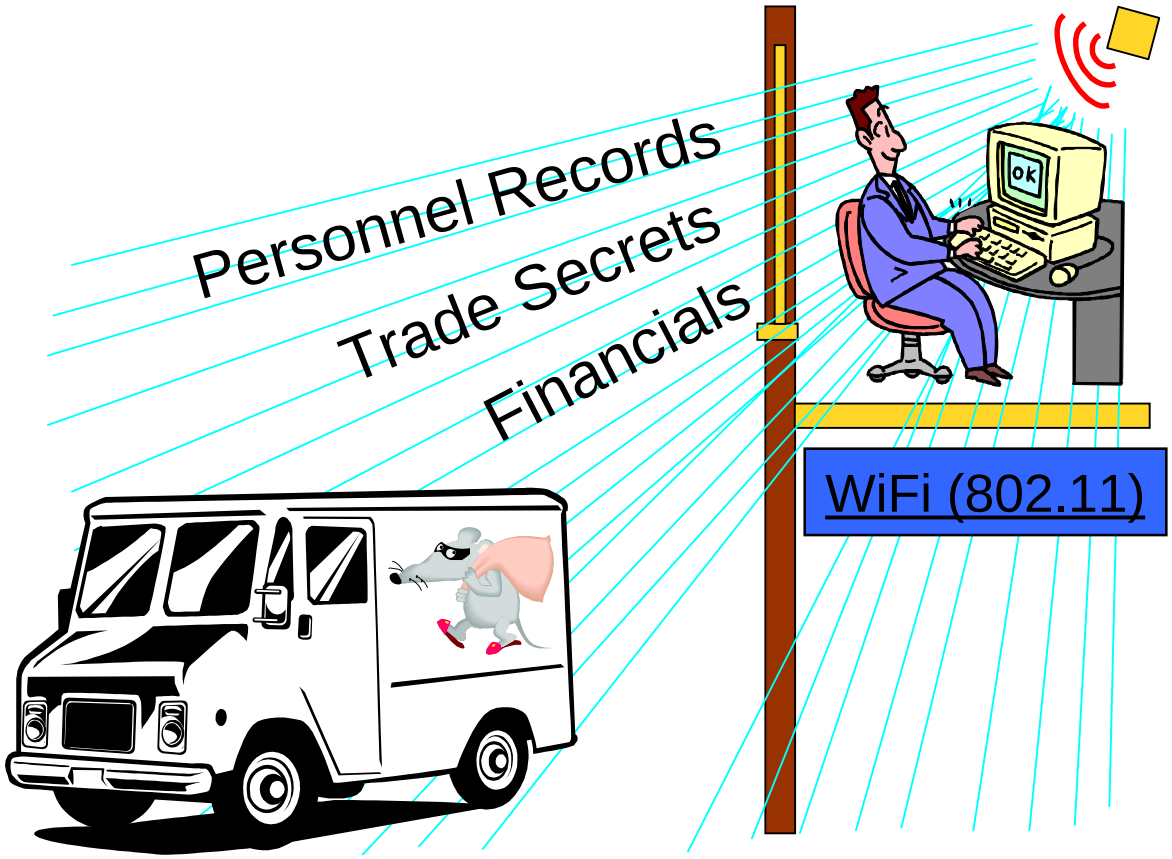
- **Scan a range of IP addresses looking for a single accessible** port
- **Target a specific vulnerable service**
- **Often seen shortly after a new vulnerability is discovered**
- **Can also be a sign of worm activity**
  - **Worm is looking for other systems with a specific vulnerable network service that can be exploited and used to spread itself**
- **Far more difficult to identify with scanner detection tools**

War Dialing

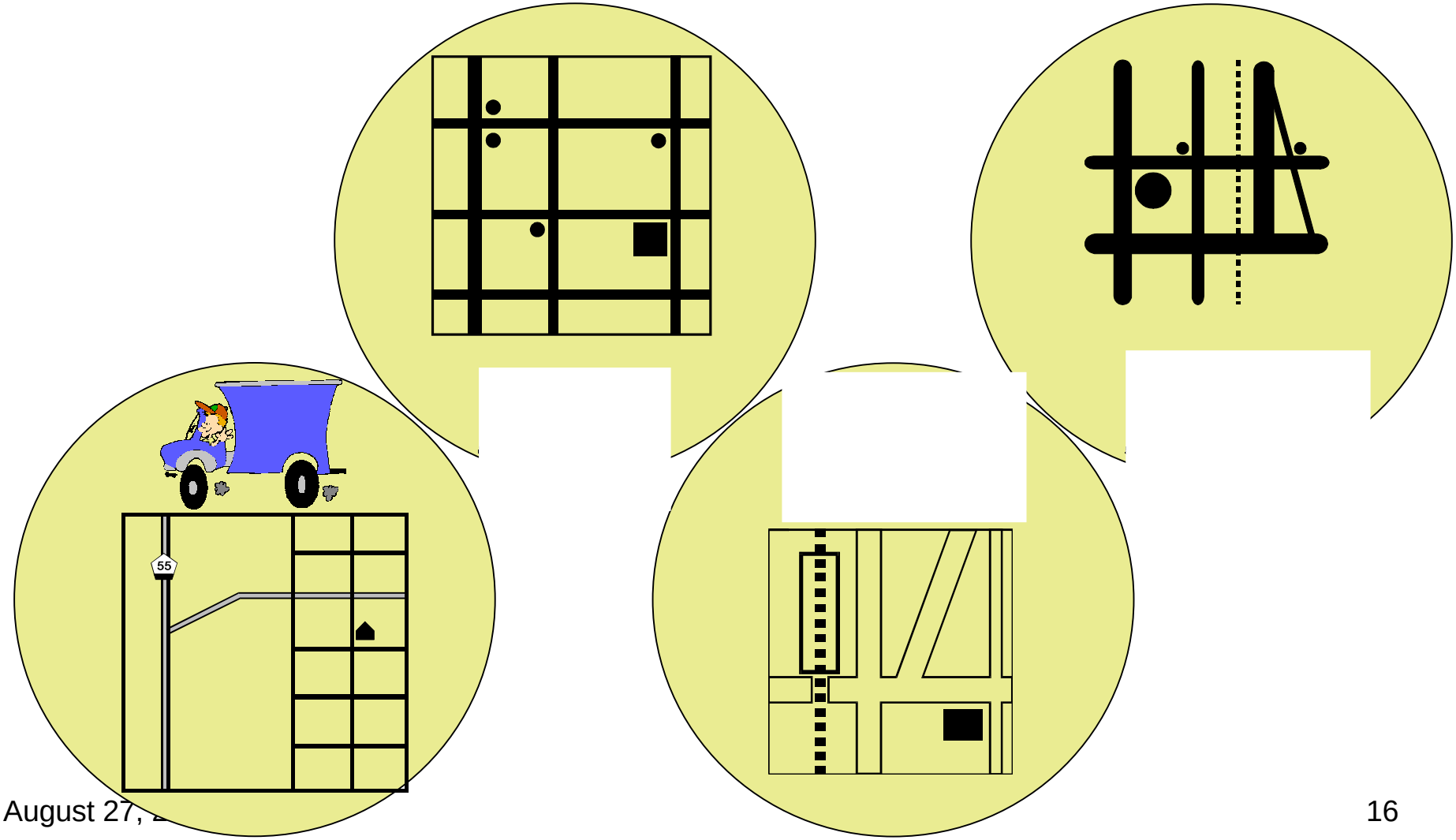
Scan a range or list of phone numbers searching for modems



Wireless networks



War Driving



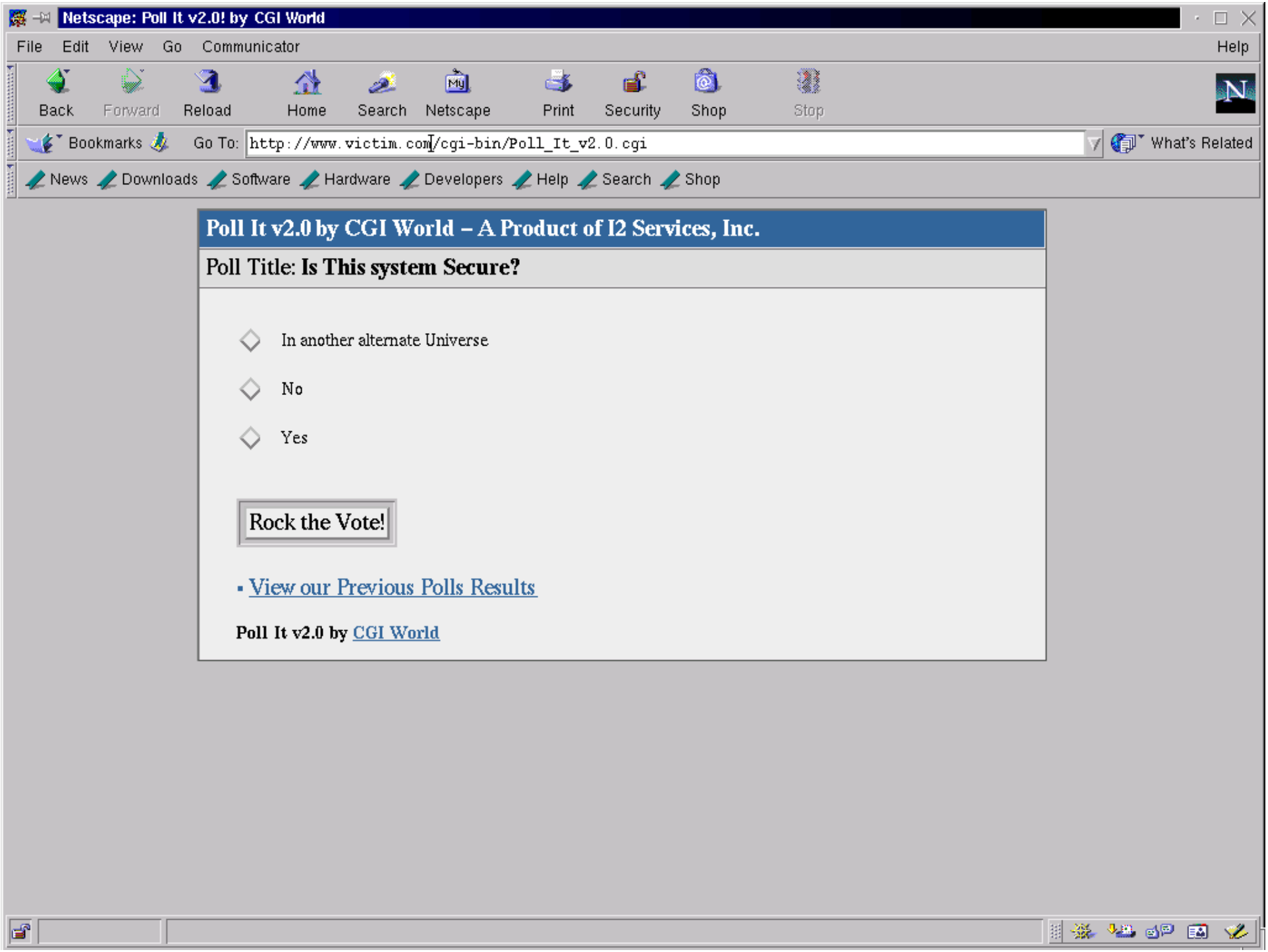
Password stealing / Cracking

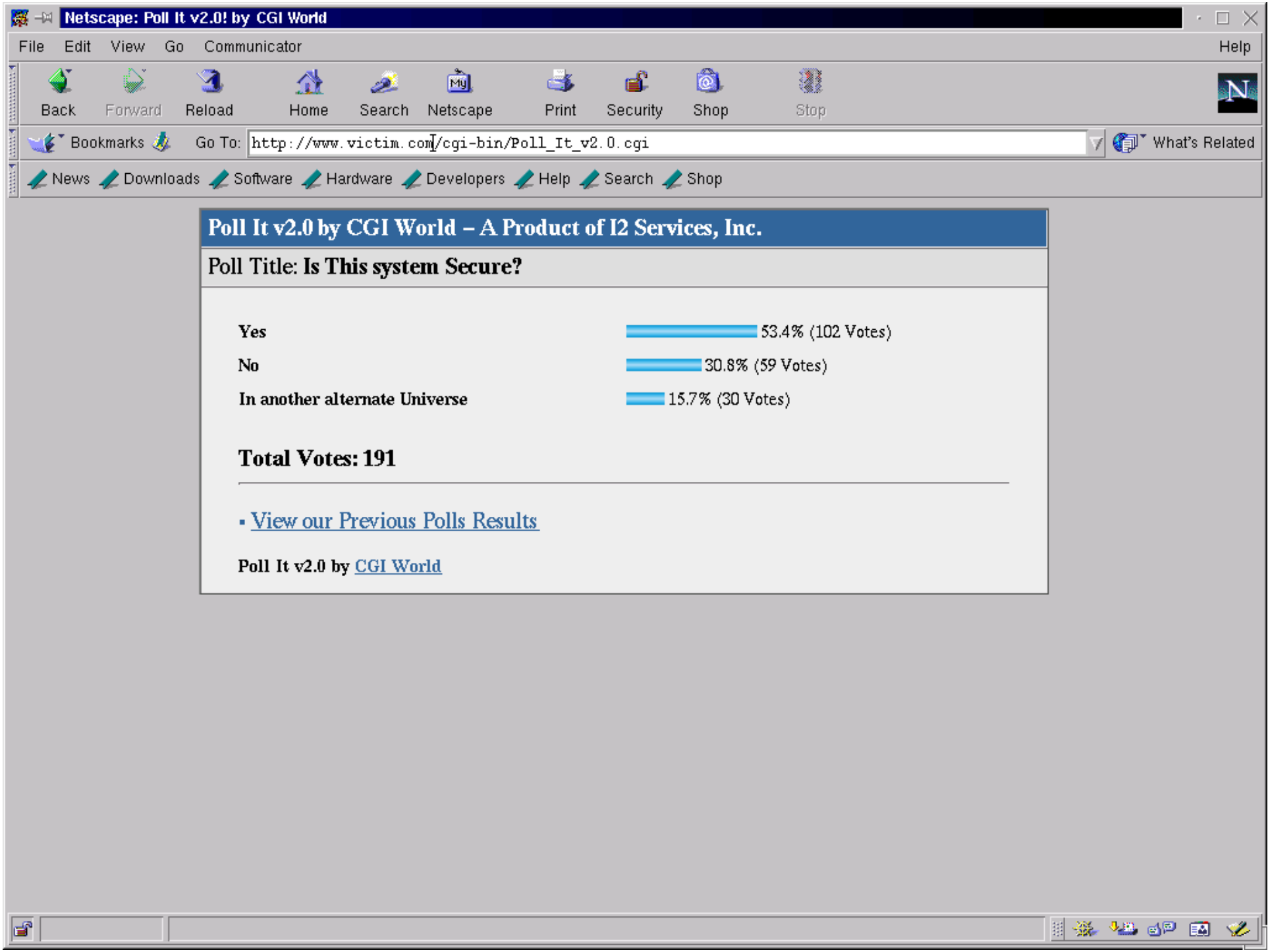
Passwords Abuse

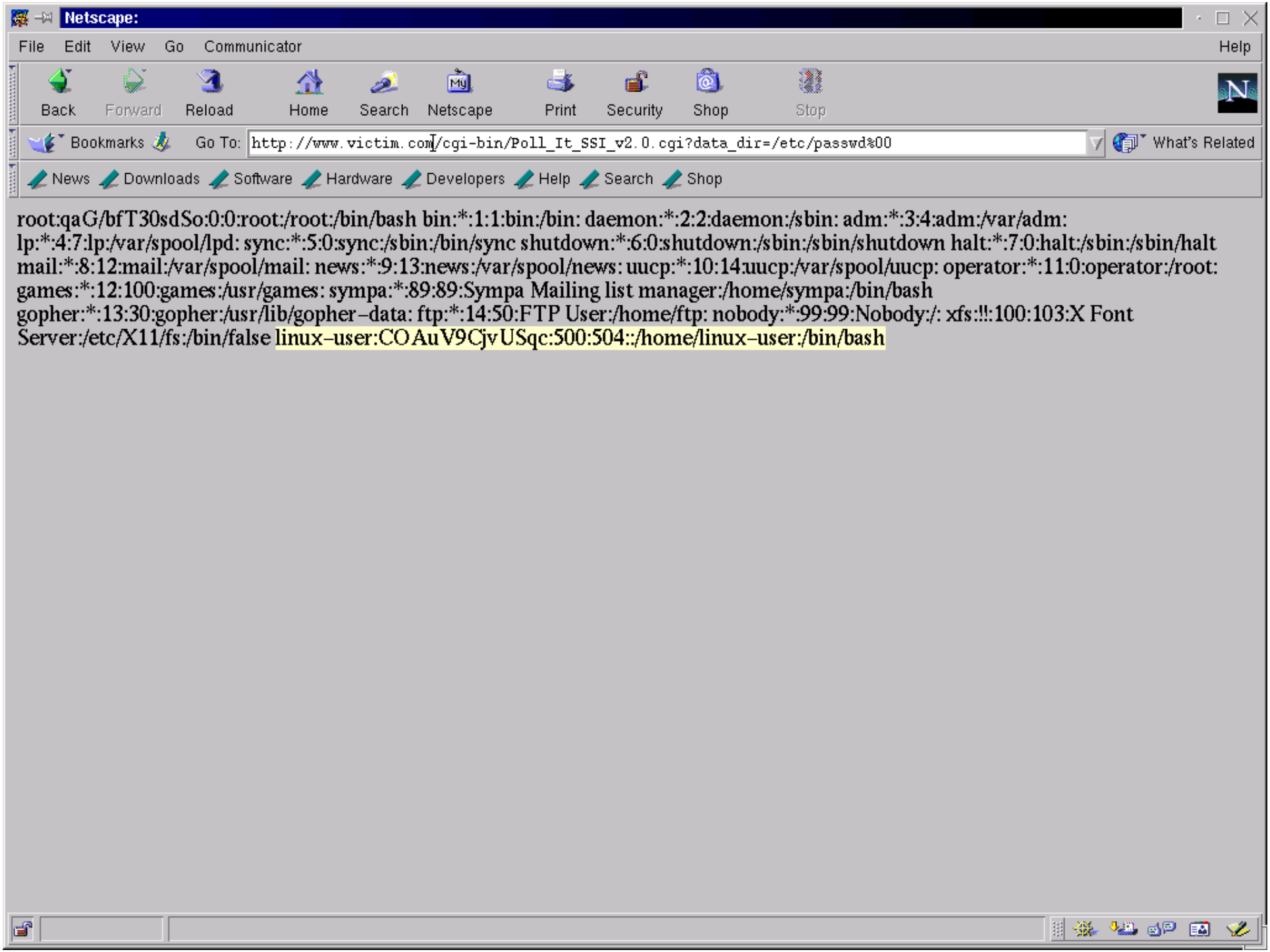
- **Password stealing (CGI script exploits, shoulder surfing, …)**
- **Network sniffing (reading the password directly from network traffic)**
- **Password guessing**
  - **Predictable passwords (blank, “guest”, user name, family name, …)**
  - **Dictionary attack (earth1 is an example of a password that is susceptible to dictionary attack)**
  - **Brute force**

CGI-bin Exploits to Steal Passwords

- **Exploits design or coding flaws in CGI-bin code**
- **Three types of exploits possible**
  - **Execute commands on web server**
  - **Read system files from web server**
  - **Modify files on web server**
- **One of the most common types of attacks for web servers**
- **Possible to use web-based search engines to locate vulnerable systems**


Netscape: Poll It v2.0! by CGI World
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Go To: http://www.victim.com/cgi-bin/Poll_It_v2.0.cgi What's Related
News Downloads Software Hardware Developers Help Search Shop
Poll It v2.0 by CGI World – A Product of I2 Services, Inc.
Poll Title: Is This system Secure?
In another alternate Universe
No
Yes
Rock the Vote!
View our Previous Polls Results
Poll It v2.0 by CGI World


Netscape: Poll It v2.0! by CGI World
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Go To: http://www.victim.com/cgi-bin/Poll_It_v2.0.cgi What's Related
News Downloads Software Hardware Developers Help Search Shop
Poll It v2.0 by CGI World – A Product of I2 Services, Inc.
Poll Title: Is This system Secure?
Yes 53.4% (102 Votes)
No 30.8% (59 Votes)
In another alternate Universe 15.7% (30 Votes)
Total Votes: 191
View our Previous Polls Results
Poll It v2.0 by CGI World


Netscape:
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Go To: http://www.victim.com/cgi-bin/Poll_It_SSI_v2.0.cgi?data_dir=/etc/passwd%00 What's Related
News Downloads Software Hardware Developers Help Search Shop
root:qaG/bfT30sdSo:0:0:root:/root:/bin/bash bin:*:1:1:bin:/bin: daemon:*:2:2:daemon:/sbin: adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd: sync:*:5:0:sync:/sbin:/bin/sync shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/spool/mail: news:*:9:13:news:/var/spool/news: uucp:*:10:14:uucp:/var/spool/uucp: operator:*:11:0:operator:/root:
games:*:12:100:games:/usr/games: sympa:*:89:89:Sympa Mailing list manager:/home/sympa:/bin/bash
gopher:*:13:30:gopher:/usr/lib/gopher-data: ftp:*:14:50:FTP User:/home/ftp: nobody:*:99:99:Nobody:/: xfs:!!:100:103:X Font
Server:/etc/X11/fs:/bin/false linux-user:CO AuV9CjvUSqc:500:504::/home/linux-user:/bin/bash

Passwords Crackers

- **Automated tools that attempt to discover passwords**
- **Requires user name and raw password hashes as input**
- **Unix / Linux tools**
  - **Crack**
  - **John the ripper**
  - **Distributed password crackers (shares the load among many systems)**
    - Mio-star
    - Saltine-cracker
    - Slurpie
  - **Many others**

/bin/bash

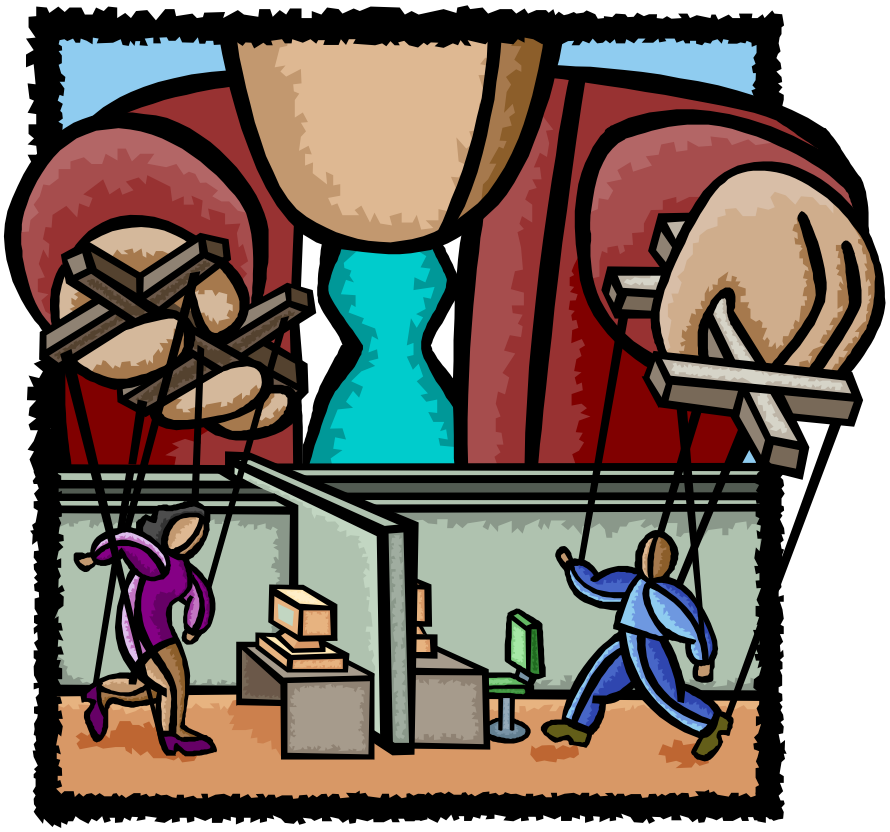
File Sessions Options Help

```
# john passwd

Loaded 5 passwords with 5 different salts (Standard DES
[24/32 4K])

john             (john)
earth1           (dave)
longpass         (rick)
```

Getting And Keeping Control



Privileged Access

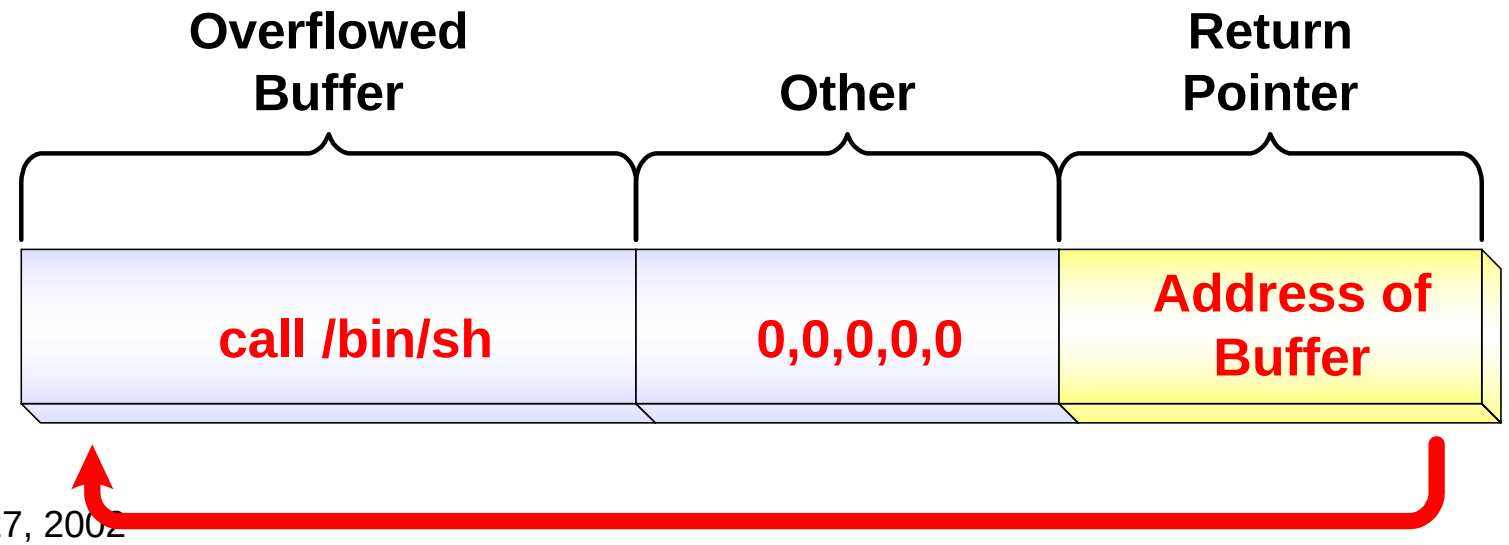
- **Exploit buffer overflow**
- **Exploit configuration errors**
- **Exploit other OS or application bugs**
- **Use a system or application backdoors (this continues to plague the community)**
- **Keep control by inserting backdoor**

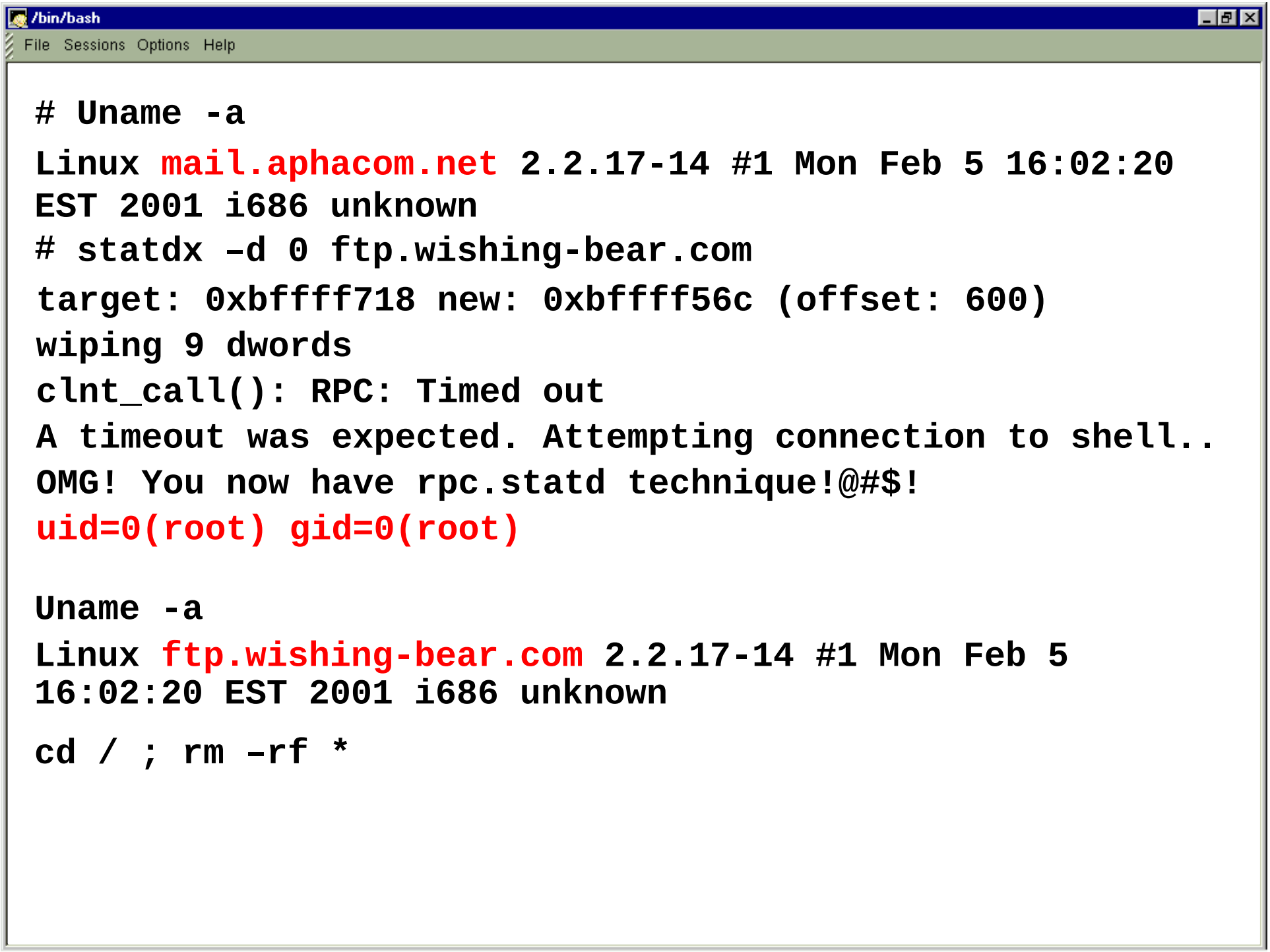
Taking Control – Buffer Overflows

- **Common attack to gain complete access**
- **Buffer overflows exploit software bugs that cause it to overwrite segments of memory**
- **Types of buffer overflows**
  - **Stack smashing**
  - **Heap overflow**
  - **Return into libc overflow**
  - **Others?**
- **New buffer overflows continue to be discovered**

Buffer Overflows

- **Overflow buffer with executable code**
- **Fill space between buffer and return pointer with random or null data**
- **Over write return pointer with address of buffer**
- **When function returns, the exploit coded is executed**




/bin/bash
File Sessions Options Help
Uname -a
Linux mail.aphacom.net 2.2.17-14 #1 Mon Feb 5 16:02:20 EST 2001 i686 unknown
statdx –d 0 ftp.wishing-bear.com
target: 0xbffff718 new: 0xbffff56c (offset: 600)
wiping 9 dwords
clnt_call(): RPC: Timed out
A timeout was expected. Attempting connection to shell..
OMG! You now have rpc.statd technique!@#$!
uid=0(root) gid=0(root)
Uname -a
Linux ftp.wishing-bear.com 2.2.17-14 #1 Mon Feb 5 16:02:20 EST 2001 i686 unknown
cd / ; rm –rf *

Keeping Control

- **Backdoors**
  - **May replace system program**
  - **Allows attackers to gain access without normal authentication process**
  - **Appear to have the same behavior as the program they are replacing**
- **Trojan horses**
  - **May appear to be a normal or reasonable executable**
  - **Are traps that can be used to compromise system**
  - **Appear to have the same behavior as the program they are replacing**

Rootkit

- **New tools**
  - **Bindshell - connects a shell to a network port**
  - **Packet sniffer specialized to look for user names and passwords**
- **Trojan tools**
  - **Ls, ps, crontab, du, find, ifconfig, netstat, pidof and top (hide presence of bindshell, sniffer)**
- **Tools that have backdoors added**
  - **Inetd, login, rshd - allow remote access without authentication**
- **Tools to remove entries from wtmp, utmp and last log**
- **Tools to modify checksum and timestamp to that of the original non-Trojan executable**
- **Other miscellaneous backdoors and tools**

Knark (Kernel Level) Rootkit

- **Knark implemented as a loadable kernel module**
- **Knark means “drugs” in Swedish**
- **Knark contains the following features:**
  - **Hide/unhide files or directories**
  - **Hide TCP or UDP connections**
  - **Execute redirection**
  - **Unauthorized privilege escalation (“rootme”)**
  - **Utility to change UID/GID of running processes**
  - **Unauthenticated, privileged remote execution daemon**
  - **Kill –31 to hide a running process**

Knark (Kernel Level) Rootkit

- **Includes the following remote exploits for:**
  - **LPR**
  - **Wu_ftpd site_exec()**
  - **Bind 8.2.1**
- **These exploits can be used to attack other systems**
- **Written by author as a Prof-of-concept**
- **Author has also written and release a program called knarkfinder.C. This tools does not identify knark specifically, but looks for hidden processes**
- **Since knark is a kernel module, any form of detection could be masked in future versions**

Covering Your Tracks

Covering Your Tracks

- **What logging is active?**
  - **syslogd**
  - **Tripwire**
  - **Event log**
  - **Commercial monitoring and intrusion detection packages**
- **Find logs**
- **Turn them off**
- **Flood them with noise**
- **Remove incriminating audit trail entries**

Stick

- **Read attack signatures from Open Source Network Intrusion Detection tool “snort”**
- **Repeatable sends random pick for list of attack signatures across a target network or directly at IDs system in the order of thousands-per-second**
- **The intent is to:**
  - **Cause Network IDS to become so busy processing signatures that it will start dropping packets and miss any real attack signatures**
  - **Report so many events that the administrator ignores or disables the IDS**
  - **The real signatures are included with thousands of other fake signatures making it very difficult to identify the actual attack**

Extend The Attack

Extend the Attack

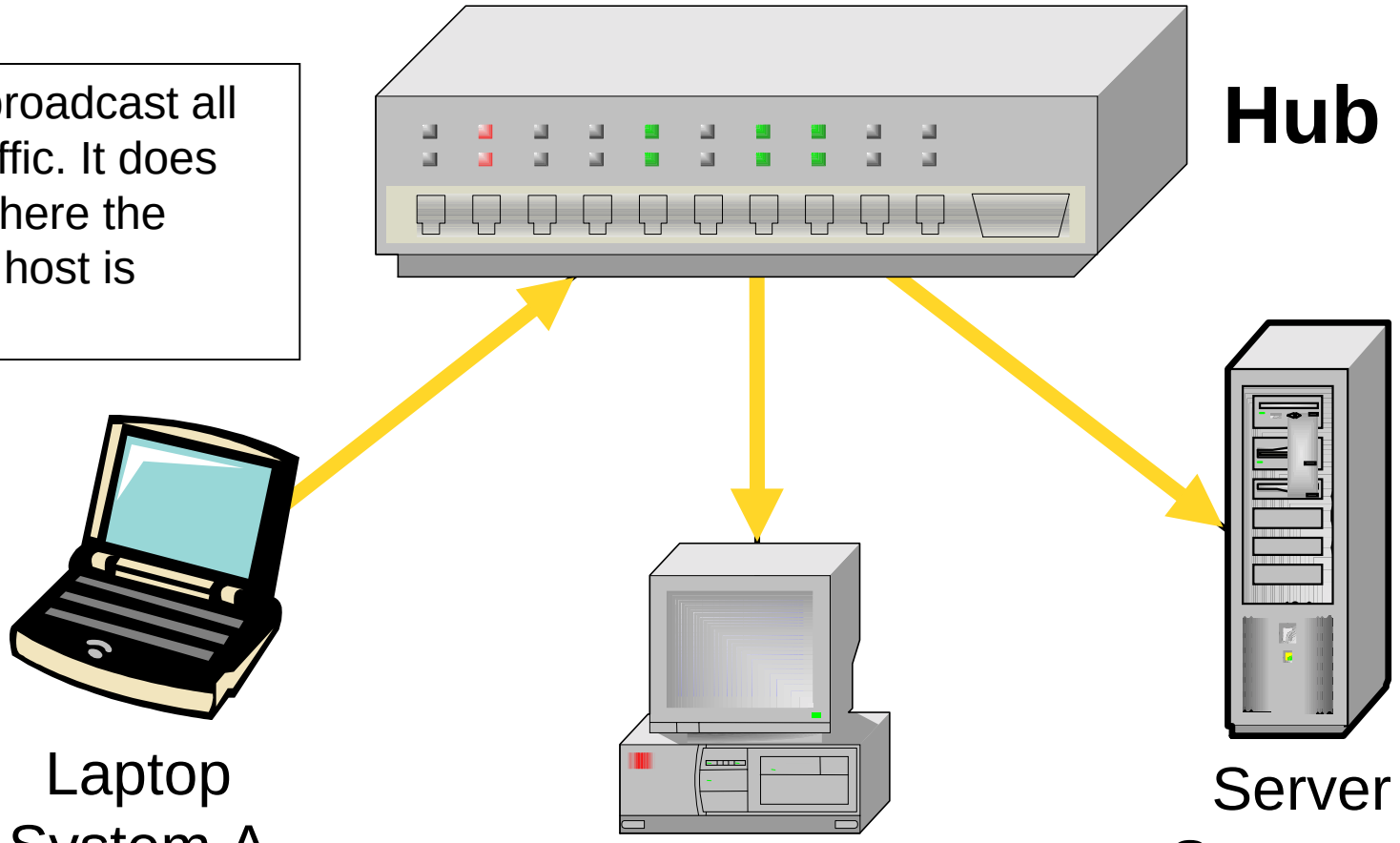
- **Once inside, the attacker can get almost any information they want**
- **Packet sniffers**
- **On-line network maps and management tools**
- **More probing to find new systems**
- **Attack other locations**
  - **Use the current site to hide their tracks**
  - **Denial-of-service**

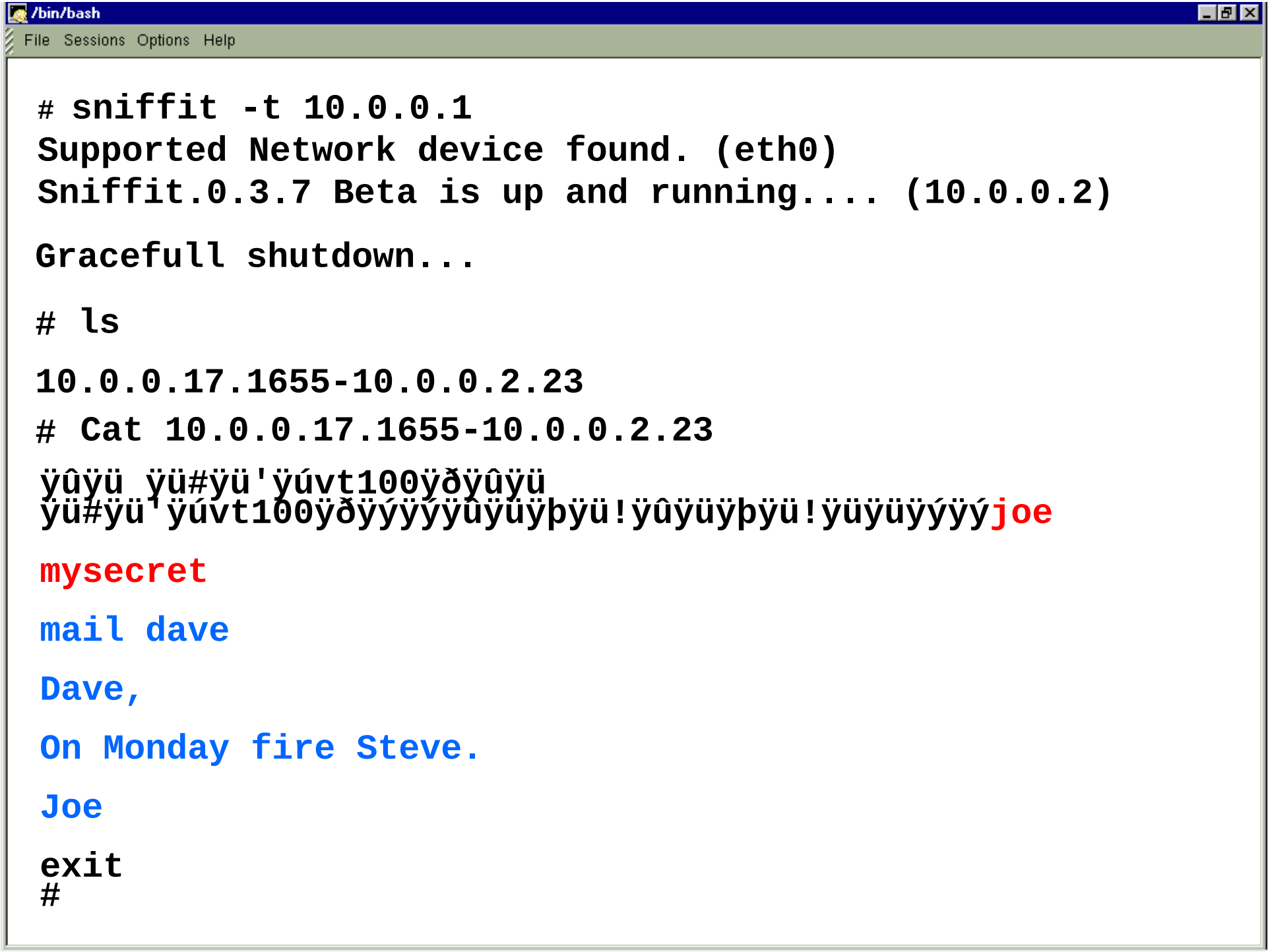
Packet Sniffers

- **Designed as a network diagnostics tool**
  - **User can dissect network packets looking for problems**
- **Places network-interface-card in promiscuous mode**
  - **All network traffic can now be read (not just that sent to the host)**
- **Can also be used to read packet payload**
  - **User name**
  - **Password**
  - **Other private content**
- **Many open source and commercial packet sniffers available (many included with operating system installation media)**
- **Some specialized versions that target just user names and password information and log it for later retrieval**

Packet Sniffers

A hub will broadcast all network traffic. It does not know where the destination host is located.




/bin/bash
File Sessions Options Help
sniffit -t 10.0.0.1
Supported Network device found. (eth0)
Sniffit.0.3.7 Beta is up and running.... (10.0.0.2)
Gracefull shutdown...
ls
10.0.0.17.1655-10.0.0.2.23
Cat 10.0.0.17.1655-10.0.0.2.23
ÿûÿü ÿü#ÿü'ÿúvt100ÿðÿûÿü
ÿü#ÿü'ÿúvt100ÿðÿýýýýûÿüÿþÿü!ÿûÿüÿþÿü!ÿüÿüÿýýýjoe
mysecret
mail dave
Dave,
On Monday fire Steve.
Joe
exit
#

What Is a Denial-of-Service

A Denial-of-Services is when someone or something is prevented from performing a desired task or operation.

Types of Denial-of-Service Attacks

- **Bandwidth Consumption**
  - **Flooding a smaller network with data**
    - flooding a 56-kbps network connection from a T1 connection.
    - This may actually be legitimate network usage
  - **Using multiple sources to flood a network**
- **Resource Starvation (Consuming system resources)**
  - **filling Disk/File system**
  - **memory fully allocated**
  - **CPU at maximum usage**
  - **Filling process table**

Definitions from “Hacking Exposed”

Types of Denial-of-Service Attacks

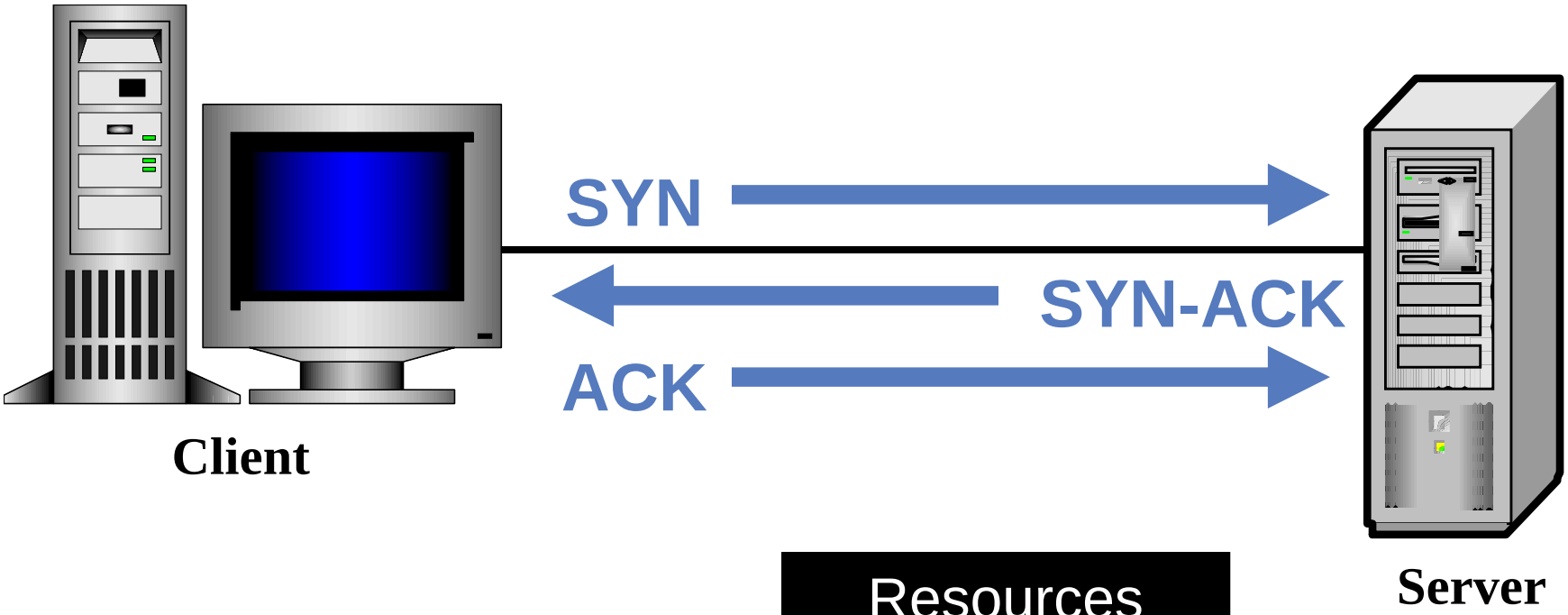
- **Programming Flaws**
  - **Buffer overflows that cause services to terminate prematurely**
  - **Memory leaks that can be used to consume system resources**
  - **Malformed or illegal network packets that cause kernel crashes**
- **Routing and DNS Attacks**
  - **Manipulation of routing tables to prevent legitimate access (breaking into routers)**
  - **Manipulation of DNS tables to point to alternate IP addresses**

Definitions from “Hacking Exposed”

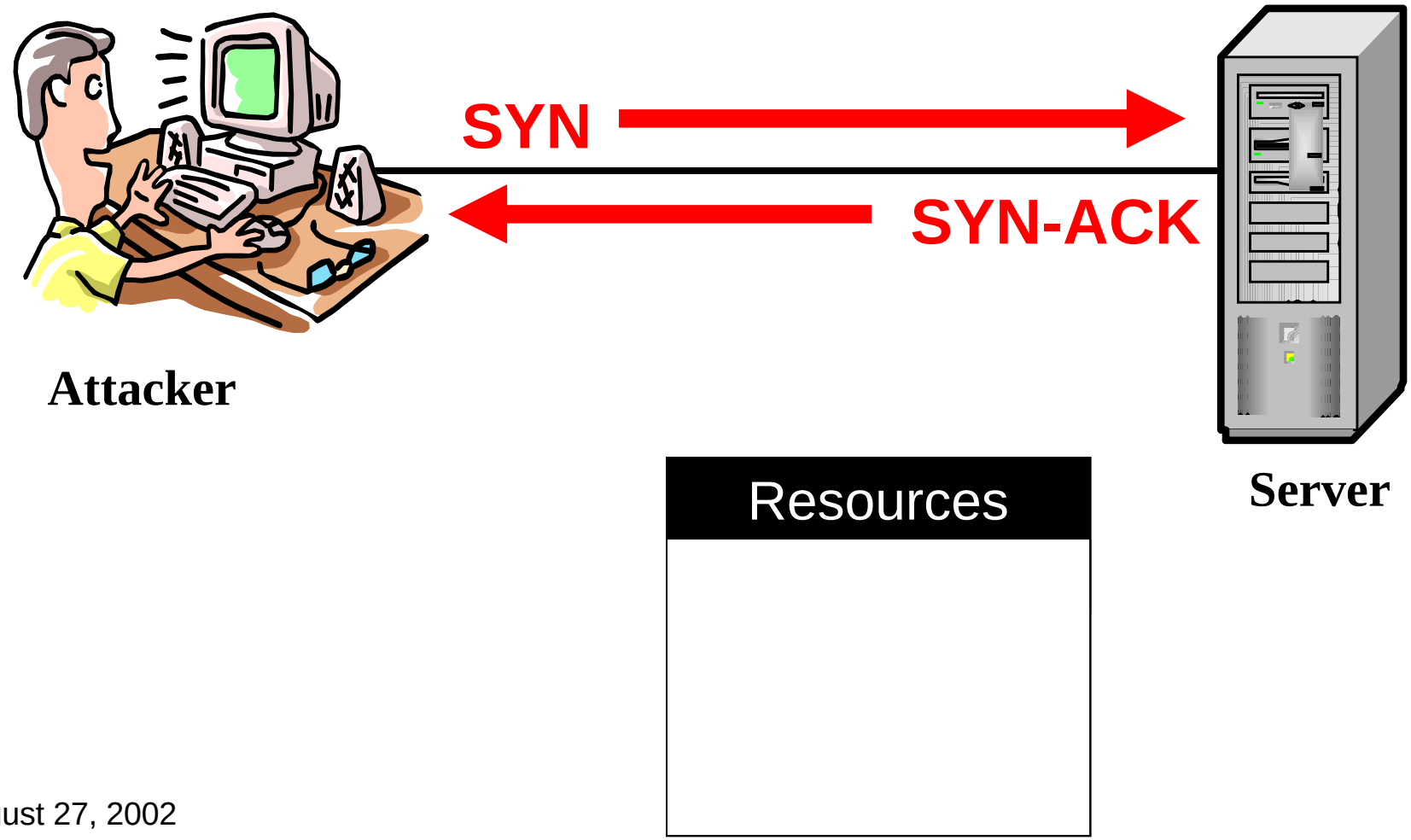
DoS Attacks Can Strike Anywhere

- **Web browsers**
  - **The browser becomes unresponsive**
  - **Continues to open windows (until system resources are exhausted)**
- **Individual Services**
  - **Disable or crash network services (a buffer overflow can cause a service to crash)**
- **The whole system**
  - **Resource attacks (file system, process table, memory, …)**
- **The whole network**
  - **NIS, DNS, …**

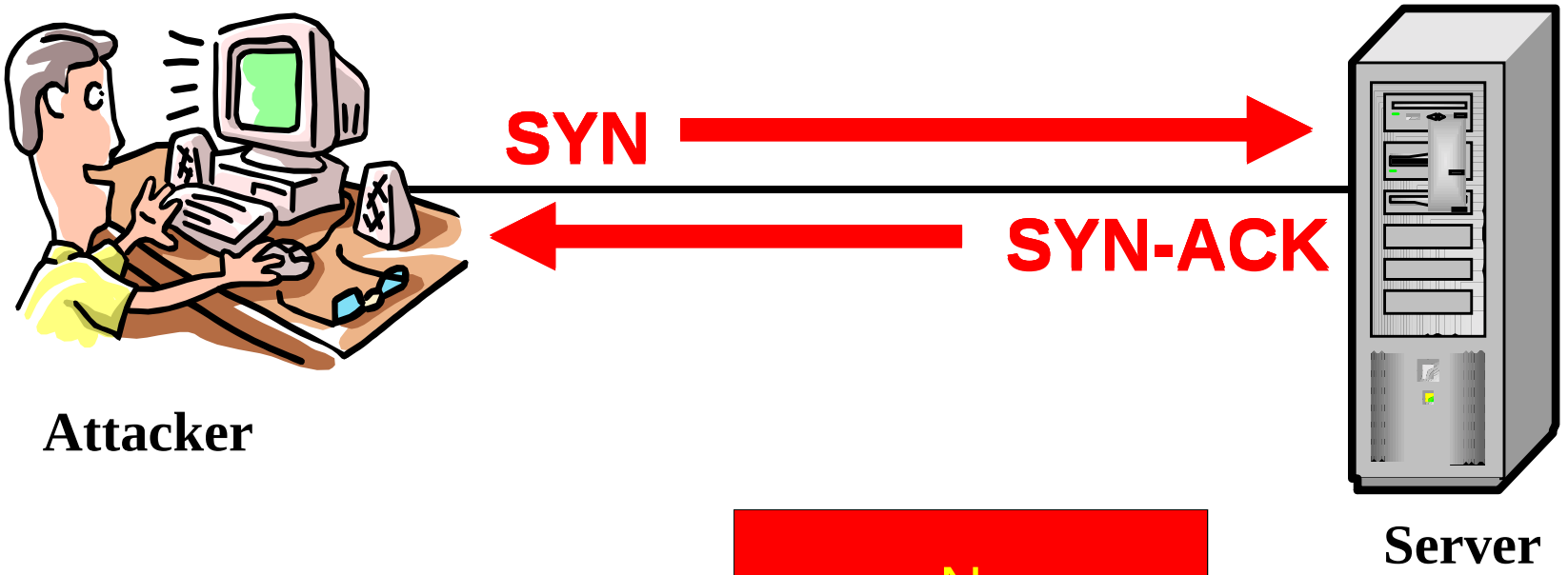
Connection Oriented 3-Way Handshake



Beginning of a Syn-flood Attack



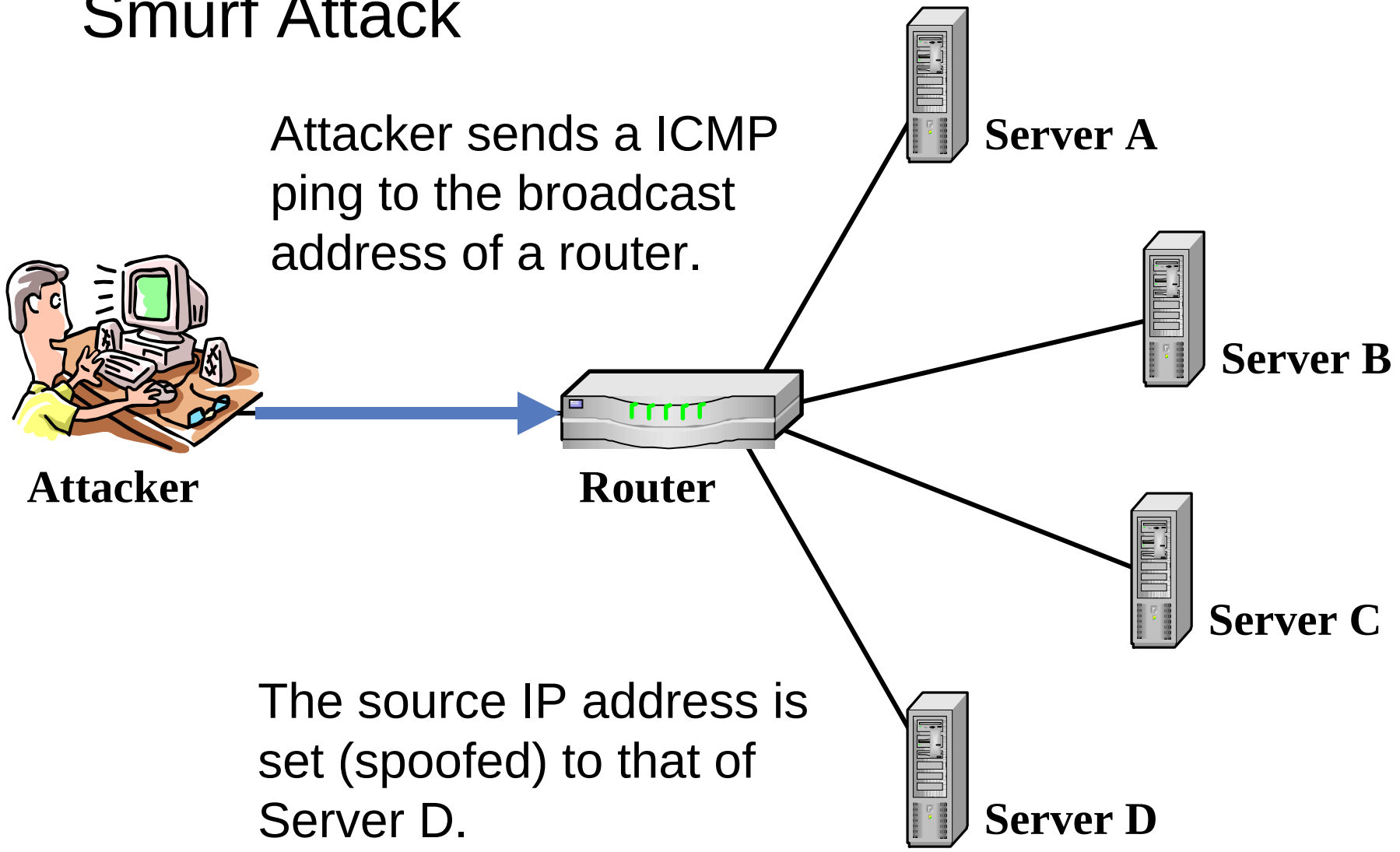
The Complete Syn-flood



No More Resources

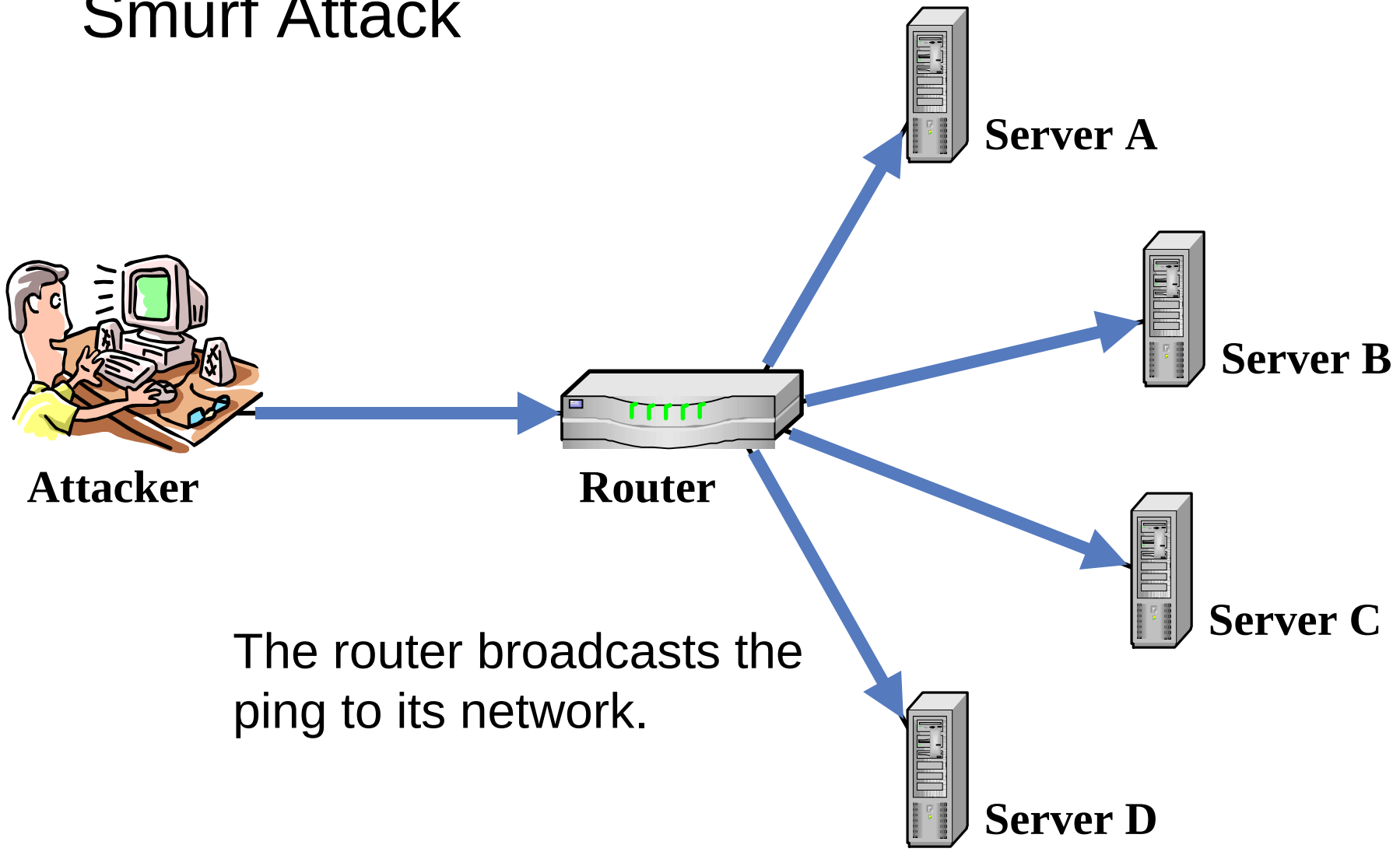
Smurf Attack

Attacker sends a ICMP ping to the broadcast address of a router.



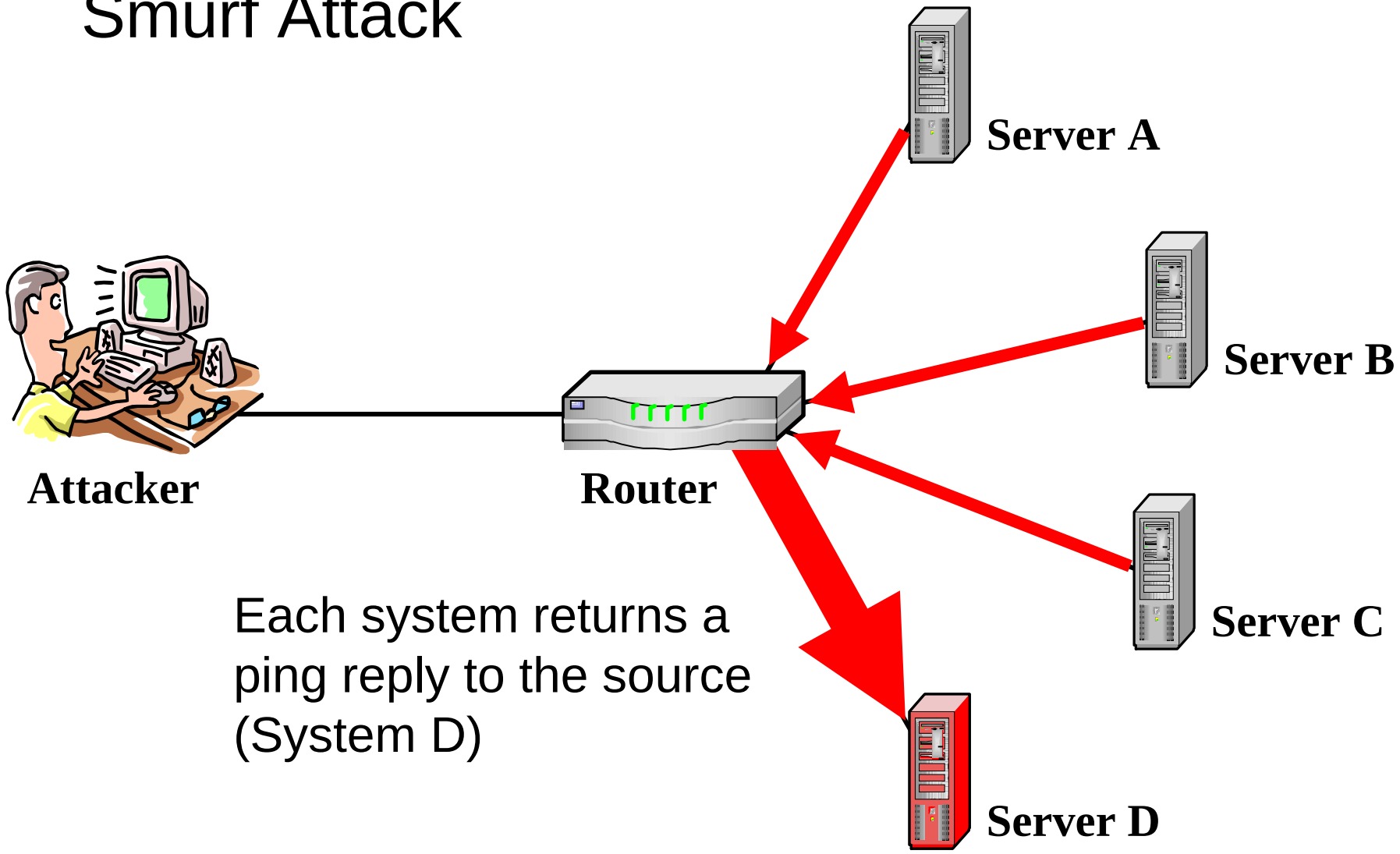
The source IP address is set (spoofed) to that of Server D.

Smurf Attack



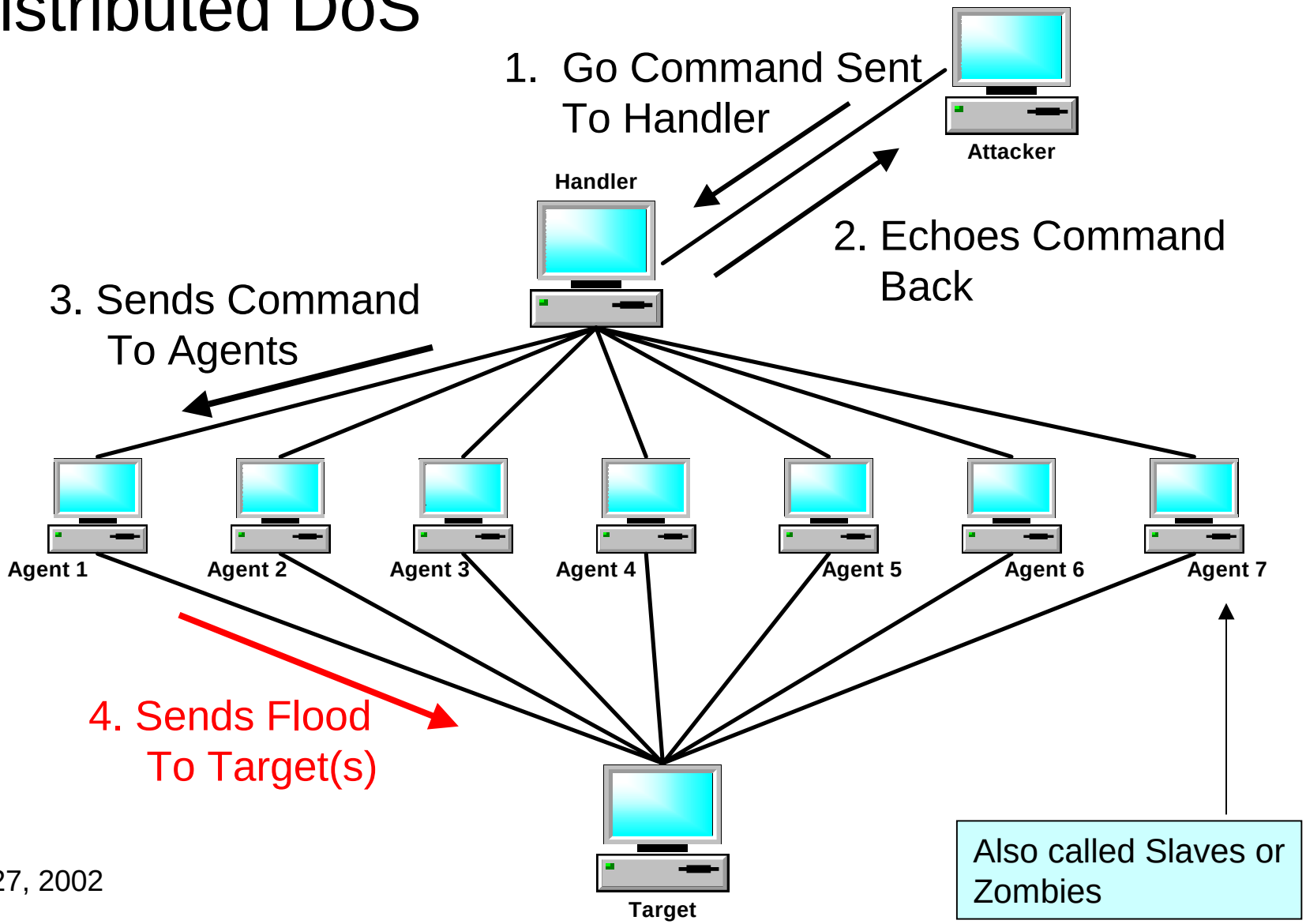
The router broadcasts the ping to its network.

Smurf Attack

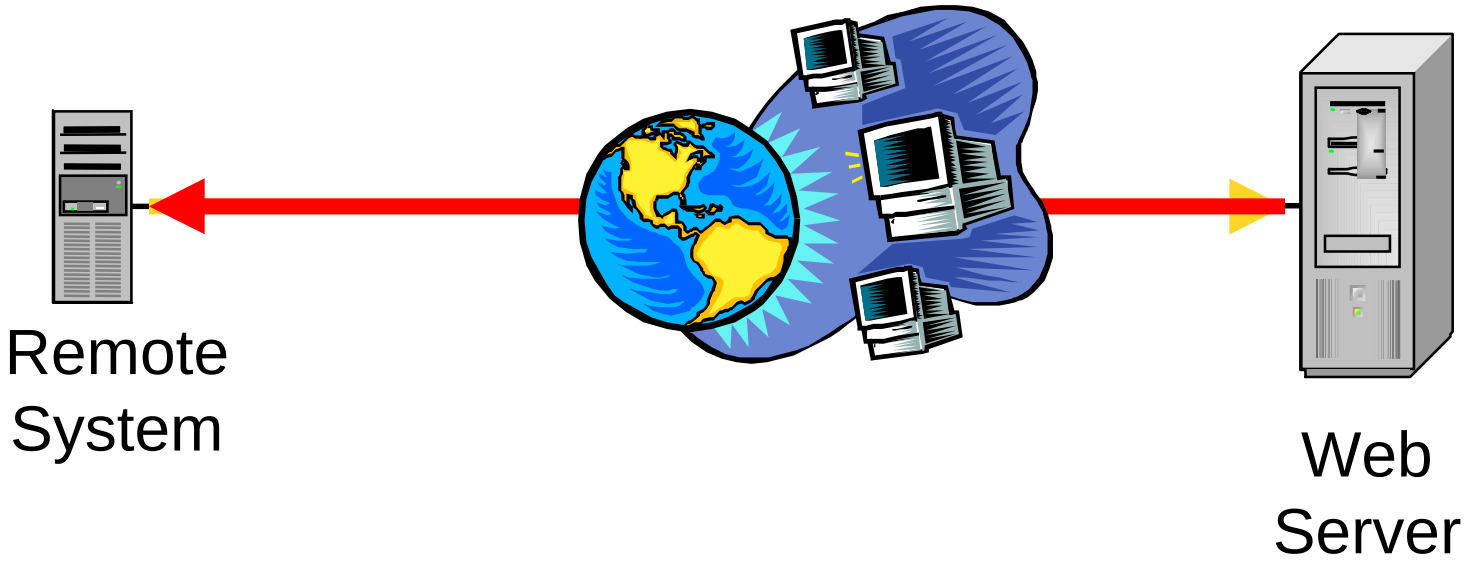


Each system returns a ping reply to the source (System D)

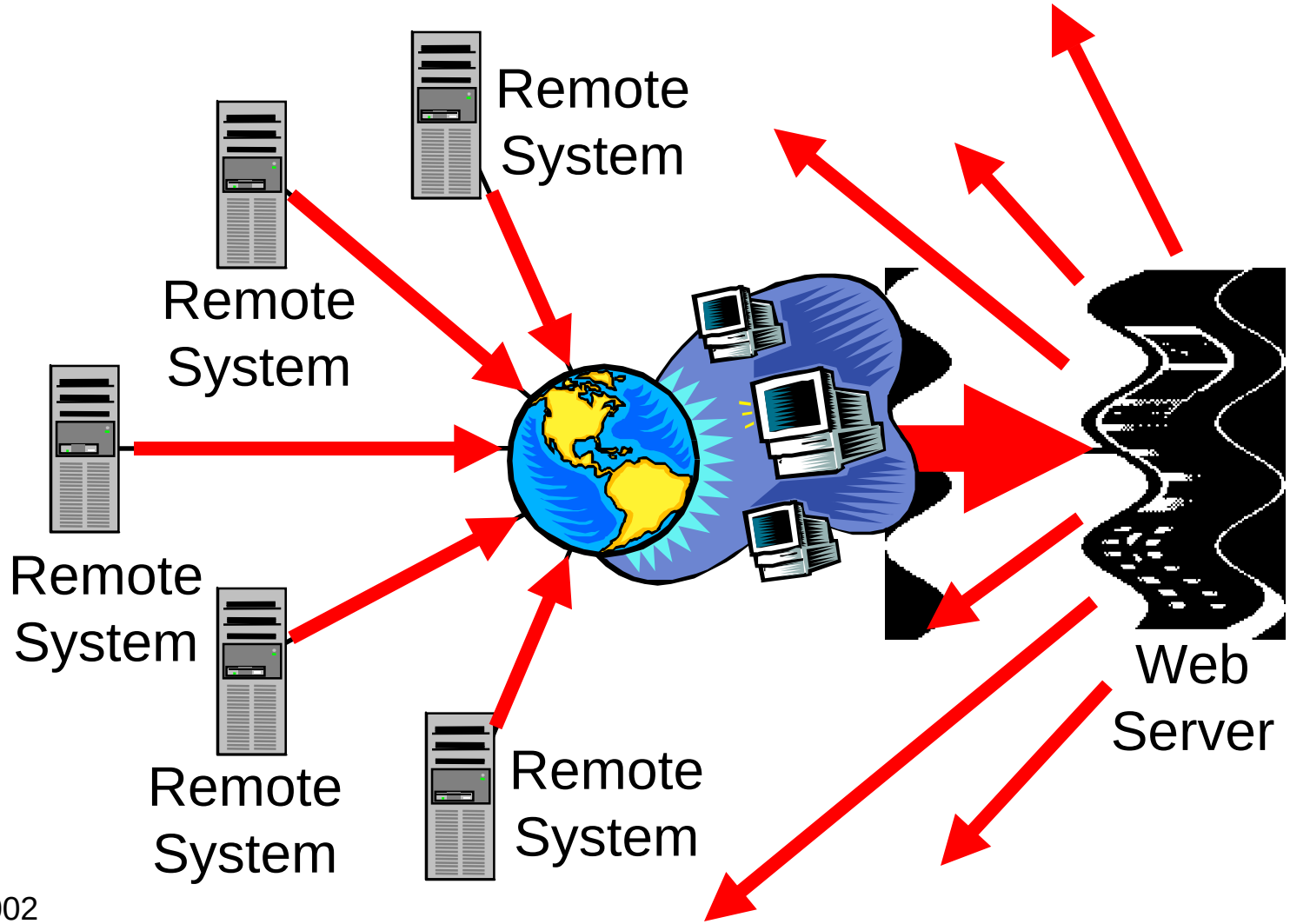
Distributed DoS



DDoS – ICMP (Ping)



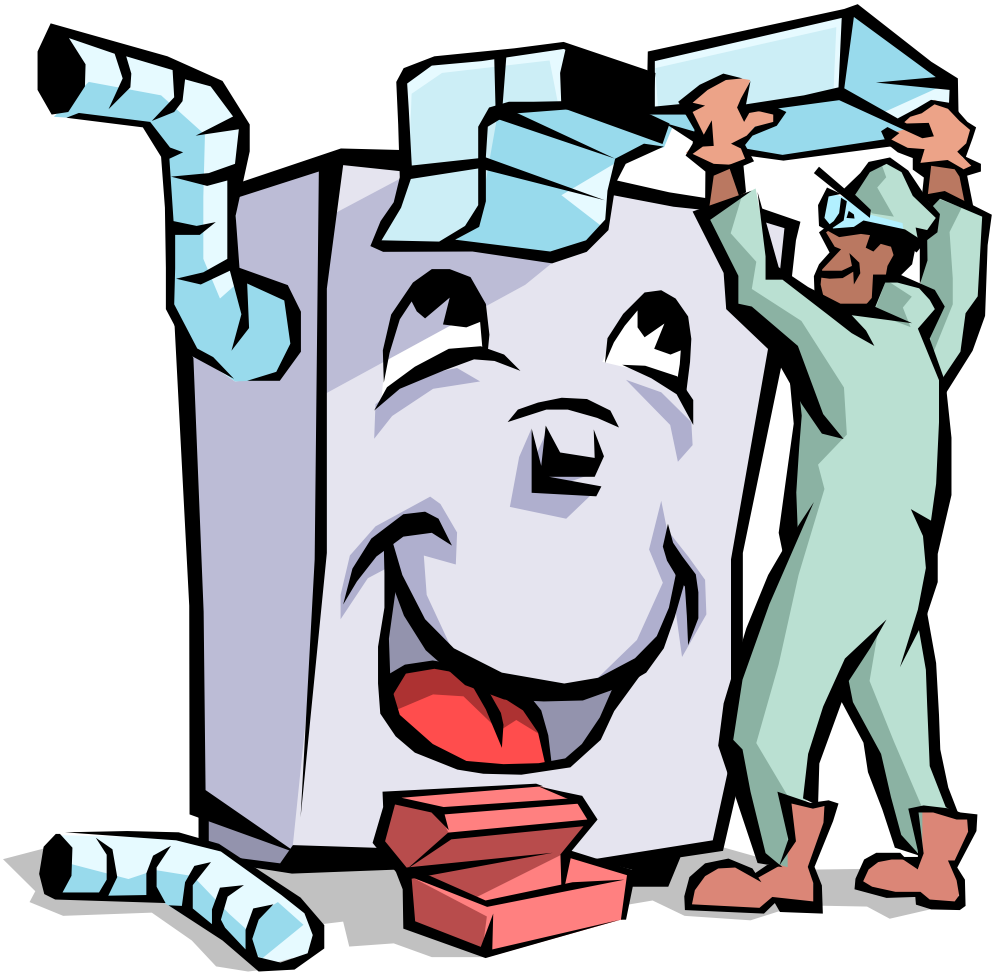
DDoS – ICMP (Ping) Flood



Viruses and Worms

- **Viruses**
  - **Historically more effective on desktop environments**
  - **Linux viruses have been very rare**
  - **Ineffective so far**
- **Worms**
  - **Historically more effective on server environments**
  - **A number of Linux worms have been written**
    - Lion
    - Adore
    - Cheese
    - Recent Apache worm
    - Some others
  - **Have had moderate effect**

III: The Solution

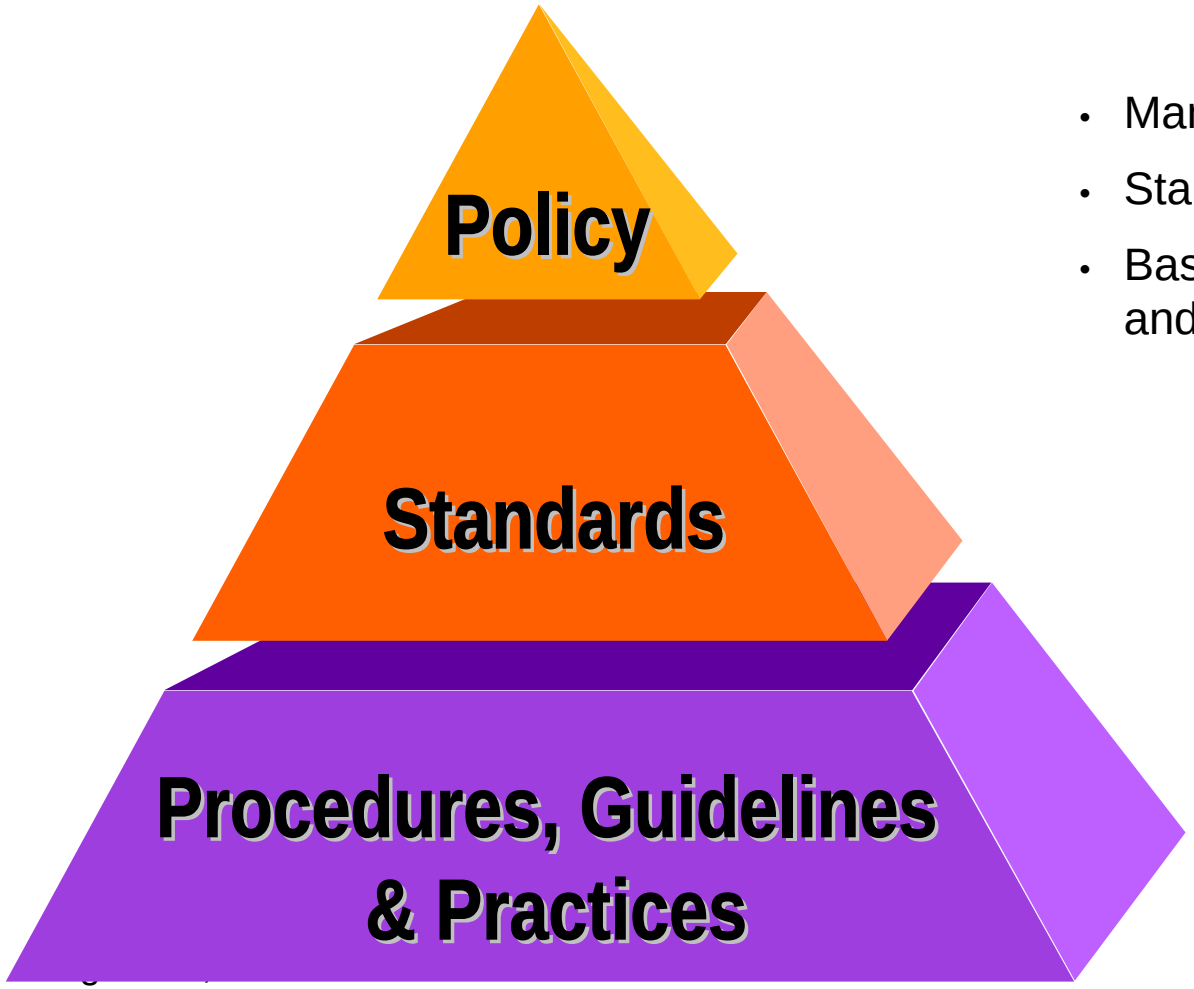


The Solution

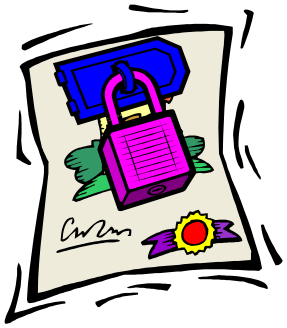
- **Start with a security policy**
- **Installation**
- **Network / system services**
- **System logging**
- **Firewalls**
- **Delegating Root**
- **Intrusion Detection**
- **Securing Email**
- **Virtual Private Networks**
- **Keep it Updated**
- **Assessment**

Start with a Security Policy

Policy Is Key to Security

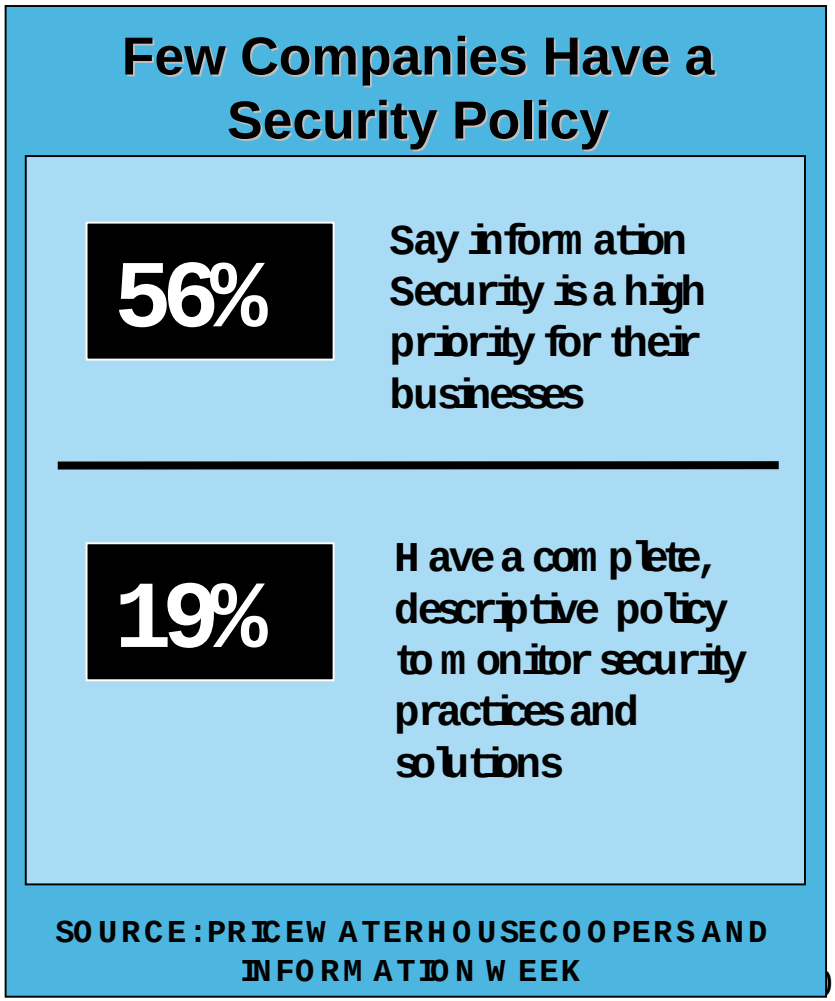


- Mandate to implement security
- Standard to measure security
- Basis for all security technology and procedures

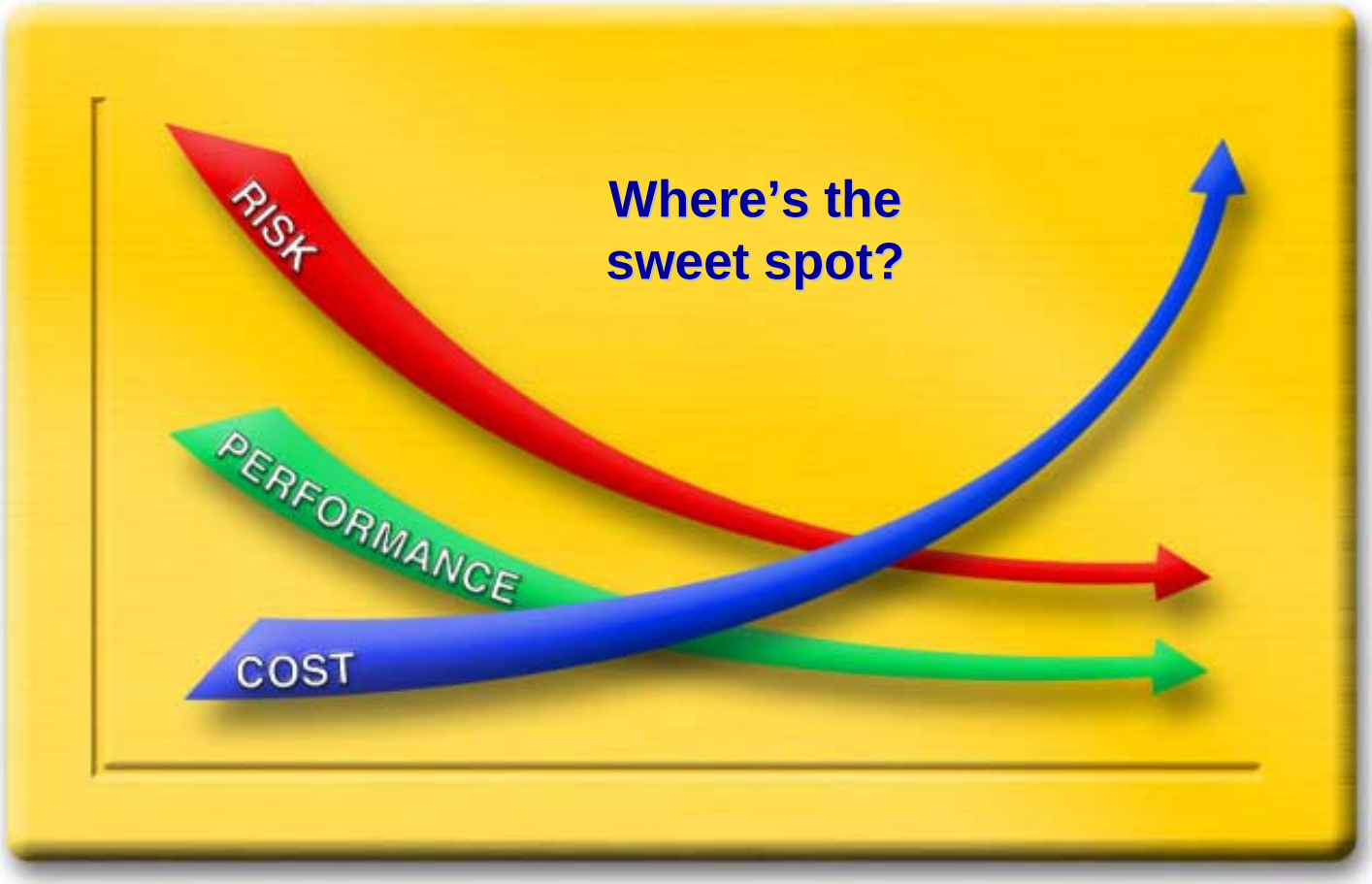


Use of Security Policy

- **Even though most businesses clearly agree that security is a high priority, few have a complete security policy**



Managing Security Risk



The Sans Security Policy Project

- **Goal is to offer everything needed for the rapid development and implementation of security policies**
- **Link to a short primer on security policies**
- **Contains example policies components**
- **Lists other resources on the web**

- **http://www.sans.org/newlook/resources/policies/policies.htm**

Installation

Install Only What You Need and Use

- **Do not use default install**
  - **Can include may utilities and services that you will never use**
  - **Only install the minimal packages required for the system to function as desired**
- **Each additional utility increases the chance that a vulnerability will be found that can lead to a system compromise**
- **Add at least one non-privileged user**
  - **This should be your default login**
  - **Use su or other delegation tools to elevate privilege (discussed later)**
- **Avoid installing servers with multiple functions (web, FTP, e-mail, …)**
  - **When ever possible move these onto separate and dedicated server systems**

Is Your Password “Hard to Guess?”

- **Don’t use easy to guess passwords**
  - **No password (carriage return)**
  - **Login name (login name = password)**
  - **Predictable names (root password to toor (root spelled backwards)**
- **Don’t use familiar names, dates and numbers**
  - **Family members (spouse, children, parents, your name)**
  - **Last name**
  - **Pet’s name**
  - **Birth date**
  - **Age**
- **Don’t use words that can be found in a dictionary (susceptible to dictionary attack)**

Pick a Strong Password

- **At least 8 characters long**
- **Use a combination of alpha/numeric characters**
- **Intermix upper case with lower case characters**
- **Combine with special characters in passwords such as punctuation marks**
- **Using the first character of each word in a phrase is a good way to create a strong password**

"A strong password, can make the difference" becomes "Asp,cmtd"

- **Avoid using common phrases**

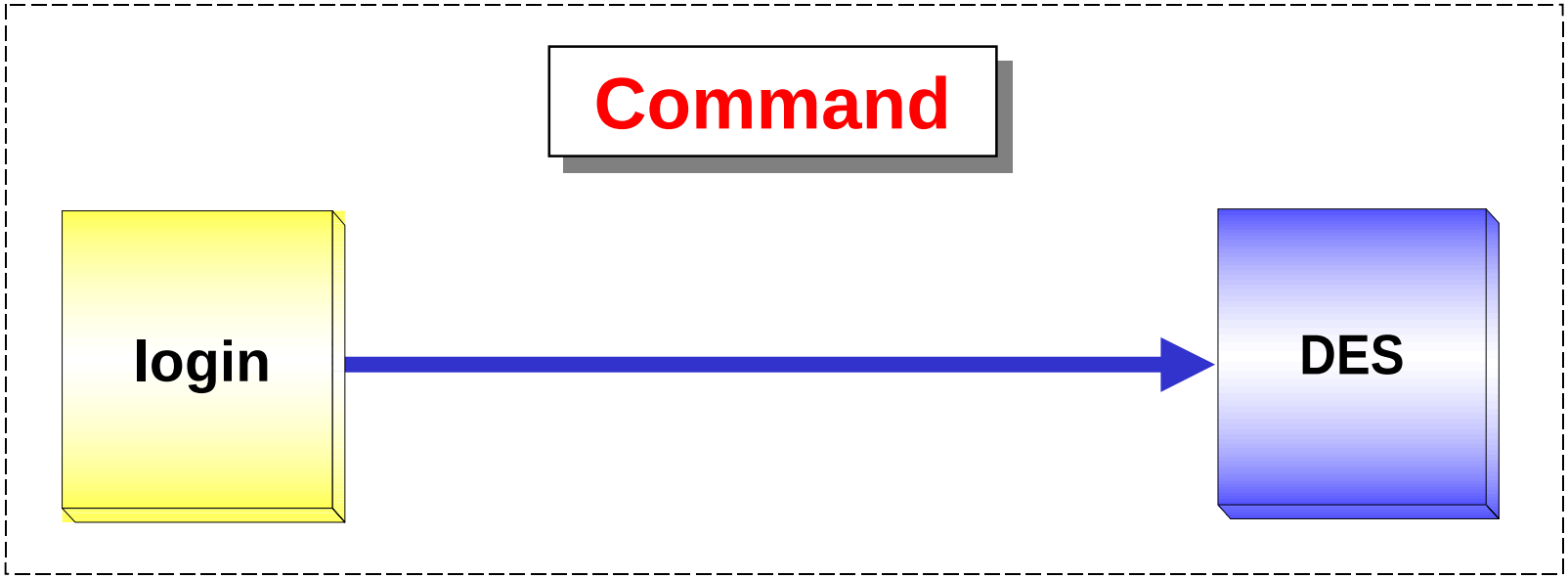
Use a Shadow Password

- **Original Unix implementation of password scheme placed login information and passwords into one file (/etc/passwd)**
- **This file needs to be readable by everyone who has access to the systems**
  - **If you do an "ls –l" you will receive a long directory listing including file and group owners obtained from the /etc/passwd file**
- **The /etc/passwd file could be used directly with a password cracker to crack passwords**
- **The shadow passwd file was implemented to deal with this problem**
  - **The shadow password file is intended to be readable by root only**
  - **User name information can still be read from the /etc/passwd file without compromise to user passwords**
- **The Linux shadow password file is located at /etc/shadow**

Traditional User Authentication

- **Linux password are represented by a unique one-way hash (numeric calculation) value**
- **The password can not be directly derived from the one-way hash value**
- **When a user attempts to login, the password string they enter is put through the one-way hash and compared with the original value**
- **The actual password is never stored on the system**
- **A DES one-way hash has been used to calculate this value in the past**
- **The password is limited to a maximum of 8 characters by DES**
- **Each command that needs to perform user authentication (login, su, …) is linked with one-way hash routines**
- **If a different authentication method is desired, each command must be re-linked with the new routines**

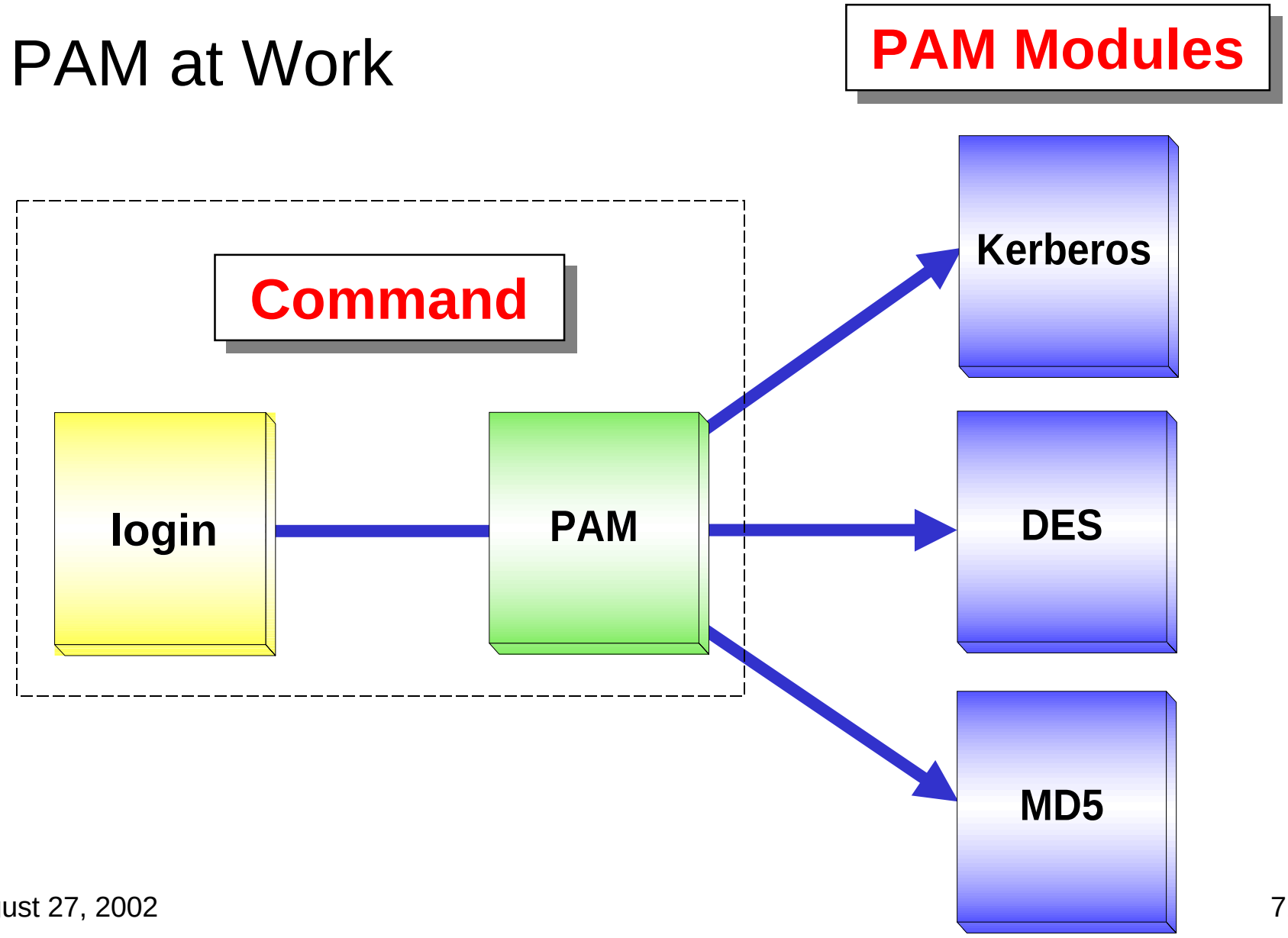
The Traditional Authentication Method



PAM – A New Face to Authentication

- **PAM functions as an abstraction layer**
- **It is linked with each command in the place of the actual one-way has routines**
- **Through the use of configuration files, PAM will load one of many authentication methods**
- **No longer is it required to link to a specific authentication method**
- **If the system administrator desires to use a different authentication method, they simply install the modules and edit the configuration files**
- **Re-linking is no longer required**
- **A large number of authentication modules are available on the internet and can simply be download and installed**

PAM at Work

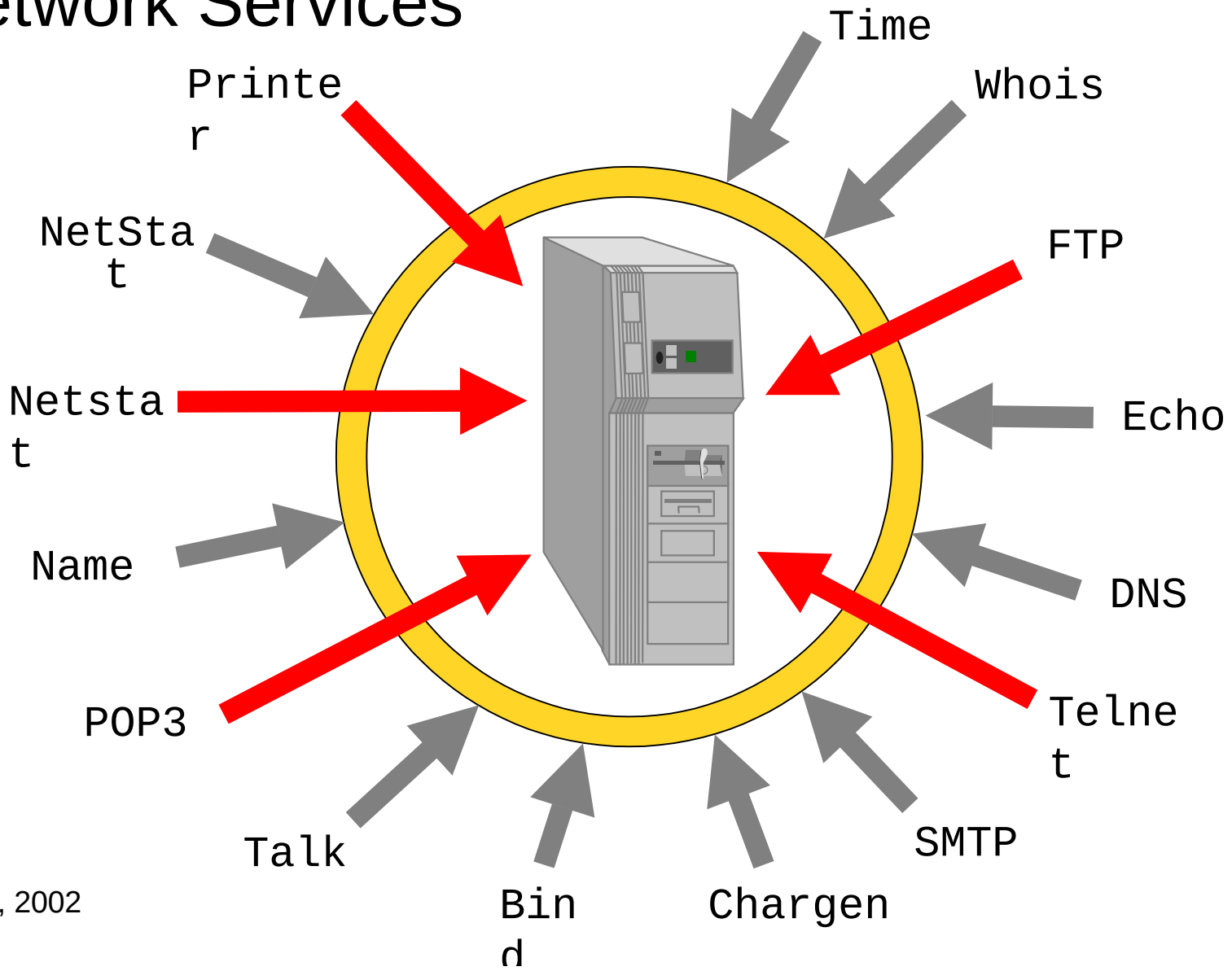


Network and System Services

Network Services

- **Network services allow one system to communicate with another**
  - **Apache web server is a network service that provides web site based capabilities**
  - **Typically run on port 80 (can run on additional or different ports)**
- **If a vulnerability is discovered in a network service, an attacker may be able to gain access to the system through an exploiting**

Network Services



Securing Network Services – Best Practices

- **Separate services onto separate systems (www, ftp, …)**
- **Identify all network services and remove all but required services**
- **Use “netstat –at” to identify all listening services**
- **Use “lsof –i +m” to find associated process for each listening port**

Using Netstat to find Network Services

```
# netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               State
tcp        0      0 *:printer                   LISTEN
tcp        0      0 *:http                      LISTEN
tcp        0      0 *:https                     LISTEN
tcp        0      0 *:32768                     LISTEN
tcp        0      0 *:sunrpc                    LISTEN
tcp        0      0 *:ssh                       LISTEN
#
```

Note: **Output has been modified for readability**

Using lsof to Find Associated Processes

```
# lsof -i +M
COMMAND     PID USER   TYPE DEVICE SIZE NODE NAME
portmap     726 root   IPv4  UDP *:sunrpc[portmapper]
portmap     726 root   IPv4  TCP *:sunrpc[portmapper]
rpc.statd   755 root   IPv4  UDP *:32768[status]
rpc.statd   755 root   IPv4  TCP *:32768[status]
sshd        904 root   IPv4  TCP *:ssh
lpd         998 root   IPv4  TCP *:printer
httpd      1028 root   IPv4  TCP *:https
httpd      1028 root   IPv4  TCP *:http
#
```

Note: **Output has been modified for readability**

Eliminating Unwanted Network Services

- **Most distributions start and stop network services from two locations**
  - **Init.d directory (/etc/init.d on most systems)**
  - **Inetd or xinetd**
- **Stop all unwanted network services and disable or remove them**
  - **If a network service is not needed, it is better to remove it to prevent accidental restart (also saves space)**

The init.d directory

- **Contains scripts to start and stop processes (including services)**
- **Links are made from each of these scripts to the run-level specific directories: rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d and rc6.d**
- **Part of the “Process Control Initialization” (see man-pages on init, Inittab, initscript and runlevel)**

Stopping a network service

- **To stop the portmap service:**
  - **cd /etc/init.d**
  - **./portmap stop** **# shutdown the service**
  - **chkconfig portmap off** **# disable service from starting**

    or
  - **rpm –qf /etc/init.d/portmap** **# which packages contains startup script**
  - **rpm –e portmap** **# remove the service completely**
  - **Apt-get remove portmap**

The Inetd Service

- **A supper service for starting other services**
  - **Saves memory and process table usage**
- **Configuration file (/etc/inetd.Conf) defines what network services inetd will monitor and the executable to call to handle each request**
- **Inetd monitors each network port specified in the "/etc/inetd.Conf" files**
- **When a connection is made to the system, inetd will identify the service type and call the appropriate executable to handle the request**
- **No ability to control access or throttle network connections**

A typical exert from the /etc/inet.d file

```
#echo          stream  tcp     nowait  root    internal
#echo          dgram   udp     wait    root    internal
#daytime       stream  tcp     nowait  root    internal
#daytime       dgram   udp     wait    root    internal
#chargen       stream  tcp     nowait  root    internal
#chargen       dgram   udp     wait    root    internal
#time          stream  tcp     nowait  root    internal
#time          dgram   udp     wait    root    internal
ftp     stream  tcp     nowait  root    in.ftpd -l -a
telnet  stream  tcp     nowait  root    in.telnetd
```

Some Problems With Inetd

- **All or nothing access control**
  - **All enabled services are available to every one**
  - **The TCP Wrappers package was written to compensate for this deficiency**
- **No connection limit**
  - **Attackers could continue to open connections until the process table is full and the system becomes unusable (DoS)**
- **Poor or nonexistent logging**
  - **By default connections are not logged**
  - **This is true for both successful for failed connection attempts**

The Xinetd Service (An Inetd Replacement)

- **Includes fine grained access control**
- **Adds enhanced logging features**
- **Provides process throttle to prevent Process-table flooding Denial-of-Service**
- **Forwarding of services requests to another system.**
- **The ability to specify unique banners for each network service.**
- **Xinetd monitors each network port specified in the “/etc/xinetd.Conf” file**
- **Generally configured to also monitor files in directory “/etc/xinitd.d”**
- **See man-page on xinetd, xinetd.conf and xinetd.log**
- **Allows for default settings (can be overridden on a per service basis)**

Eliminating services – Our example

- **Shutdown and removed the following services with scripts in /etc/init.d**
  - **Portmap** **(portmap service)**
  - **nfs-utils** **(statd service)**
  - **LPRng** **(printer service)**
  - **yp-tools** **(nfs-utils dependency)**
  - **Ypbind** **(nfs-utils dependency)**
  - **Ypserv** **(nfs-utils dependency)**
- **Xinetd was not being used for any network services and was also removed**

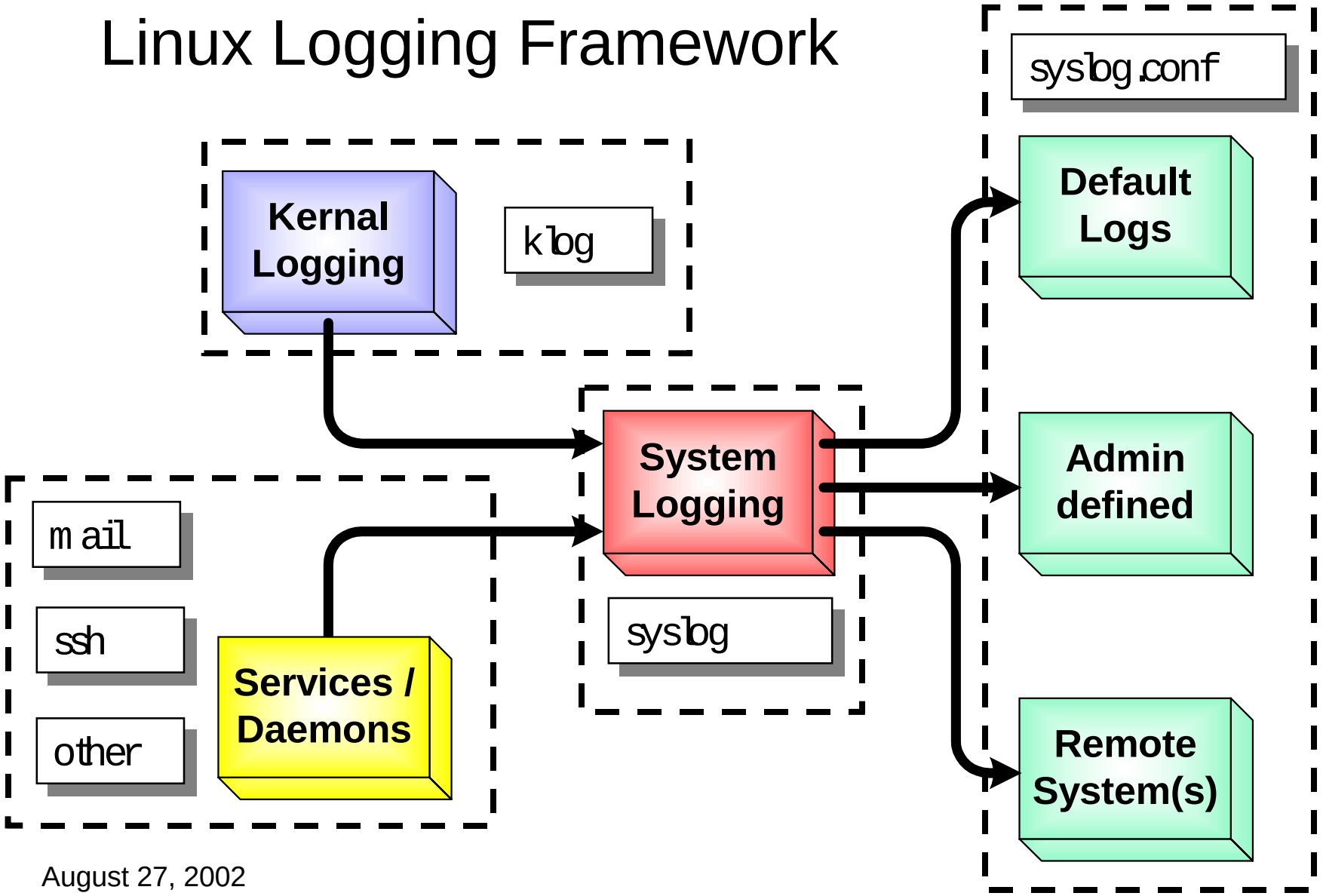
The results

```
# netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               State
tcp        0      0 *:http                      LISTEN
tcp        0      0 *:https                     LISTEN
tcp        0      0 *:ssh                       LISTEN
#
```

Note: **Output has been modified for readability**

System Logs

Linux Logging Framework



Syslog.conf

- **Some default locations (typically located in /var/log directory)**
  - **messages — Default location for most event messages**
  - **secure — Events where a password is required**
  - **maillog — email related events (pop, imap sendmail, …)**
  - **spooler —**
  - **auth —**
- **Adding additional logging**
  - **mail.* @mail-log-host**
- **Adding remote logging**
  - **mail.* @mail-log-host**

Examples from “Linux Administrator's Security Guide” by Kurt Seifried

Securing Syslog

- **Protect the logs by making them unreadable by anybody other than root**
  - **chmod 700 /var/admin**
- **Export log information to another system**
  - **if the system compromised, the attacker will also need to compromise this external system to remove the evidence**
  - **Adding the following to syslog.conf:**

***.* @external-system**

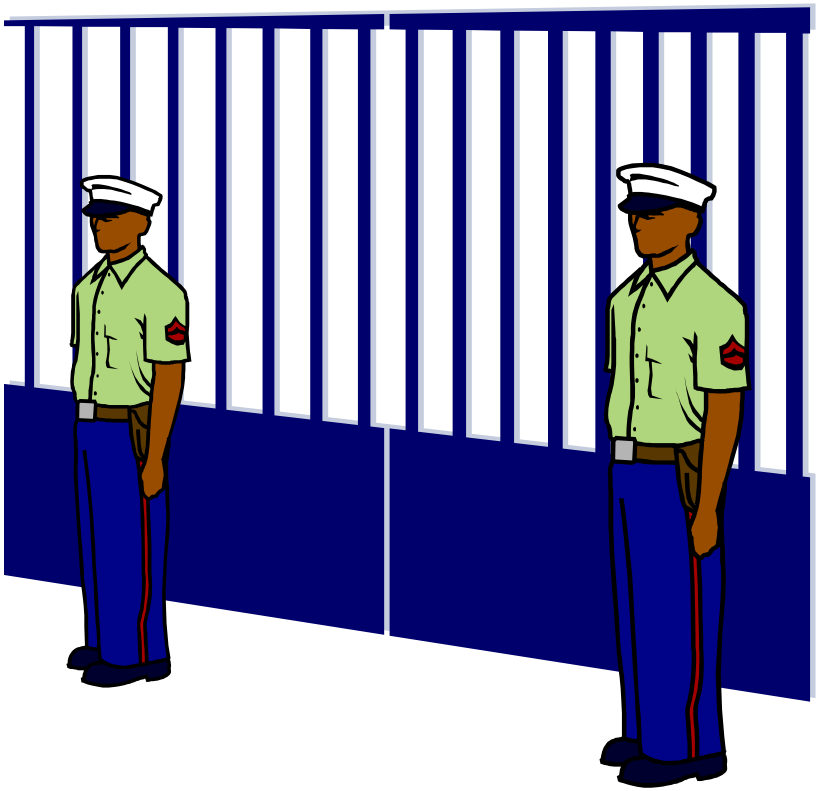
sends everything to external-system

Firewalls

Firewalls

- **Can be used to control access to a single system or an entire Network**
- **Used to control what gets in and out**
- **Limits the type of traffic**
- **A typical used would**
  - **Block all incoming**
  - **Allow all outgoing**
  - **Select protocols can be allows in or out**
- **Firewalls really enforce policy for traffic between networks (Intranet and Internet/Extranet)**

- **Goal: Keep the bad guys out!**



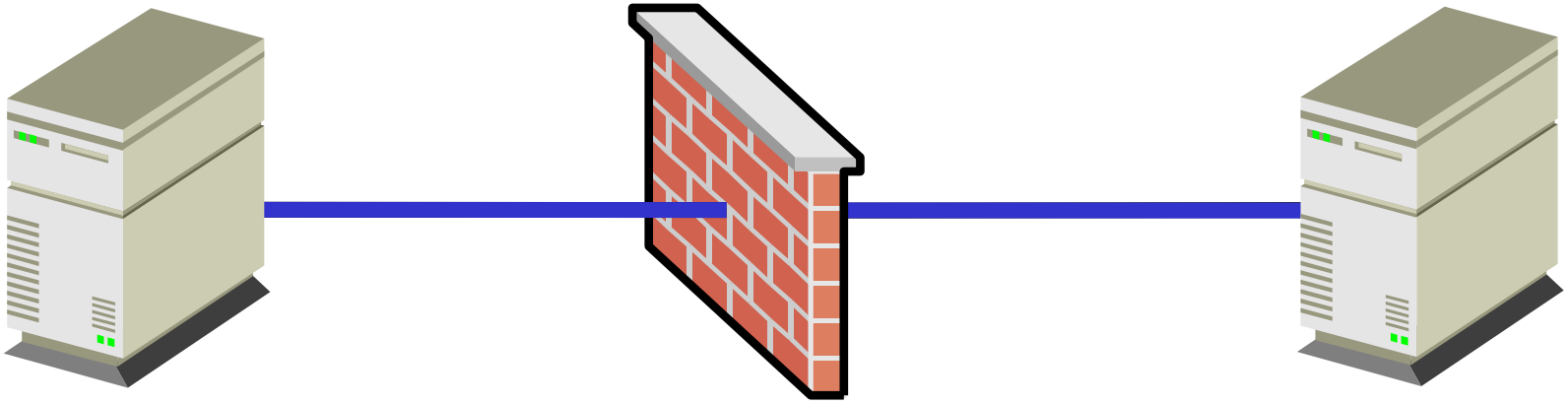
Types of Firewalls – Packet Filtering Firewall

- **This is the type of firewall built into the Linux kernel**
- **Control is at the network level**
  - **Data is only allow to enter or leave the system if the firewall rule allows it**
  - **As packets arrive they are filtered by their type – source address, destination address and port information**
- **Filtering firewalls do not provide for password controls. User can not identify themselves**
  - **The only identity a user has is the IP number assigned to their workstation**
  - **This can be a problem if you are going to use DHCP (Dynamic IP assignments)**
- **Filtering firewalls are more transparent to the user**
  - **The user does not have to setup rules in their applications to use the Internet**

Types of Firewalls – Proxy Server

- **Proxies are mostly used to control, or monitor, outbound traffic**
  - **Some application proxies cache the requested data**
  - **This lowers bandwidth requirements and decreases the access the same data for the next user**
  - **It also gives unquestionable evidence of what was transferred**
- **There are two types of proxy servers**
  - **Application proxies - that do the work for you**
  - **SOCKS proxies - that cross wire ports**

Types of Firewalls – Application Proxy



An Application proxy accts as a go-between (proxy) – Content can be verified and logged – authentication can also be established

Types of Firewalls – Application Proxy

- **Because proxy servers are handling all the communications, they can log everything**
  - **Every web URL**
  - **Every ftp download**
  - **Verify that content if valid (http requests are valid http)**
- **Authentication can also be performed at the application proxy**
  - **Before a connection to the outside is made, the server can ask the user to login first**
  - **To a web user this would make every site look like it required a login**

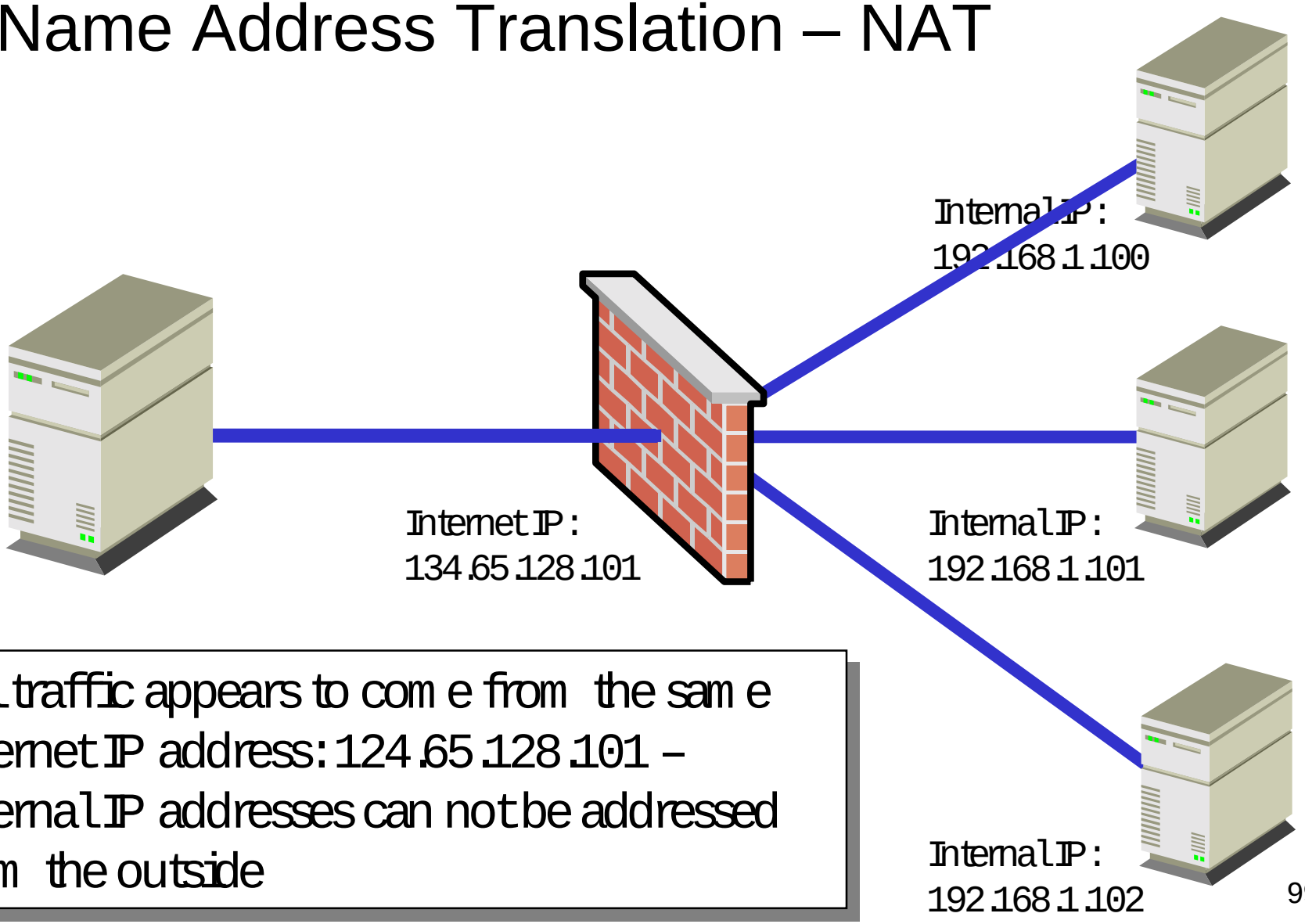
Types of Firewalls – SOCKS Proxy

- **A SOCKS server is a lot like an old switch board**
  - **It simply cross wires your connection through the system to another outside connection**
- **Most SOCKS server only work with TCP type connections**
  - **And like filtering firewalls they don't provide for user authentication. They can however record where each user connected to**

Name Address Translation – NAT

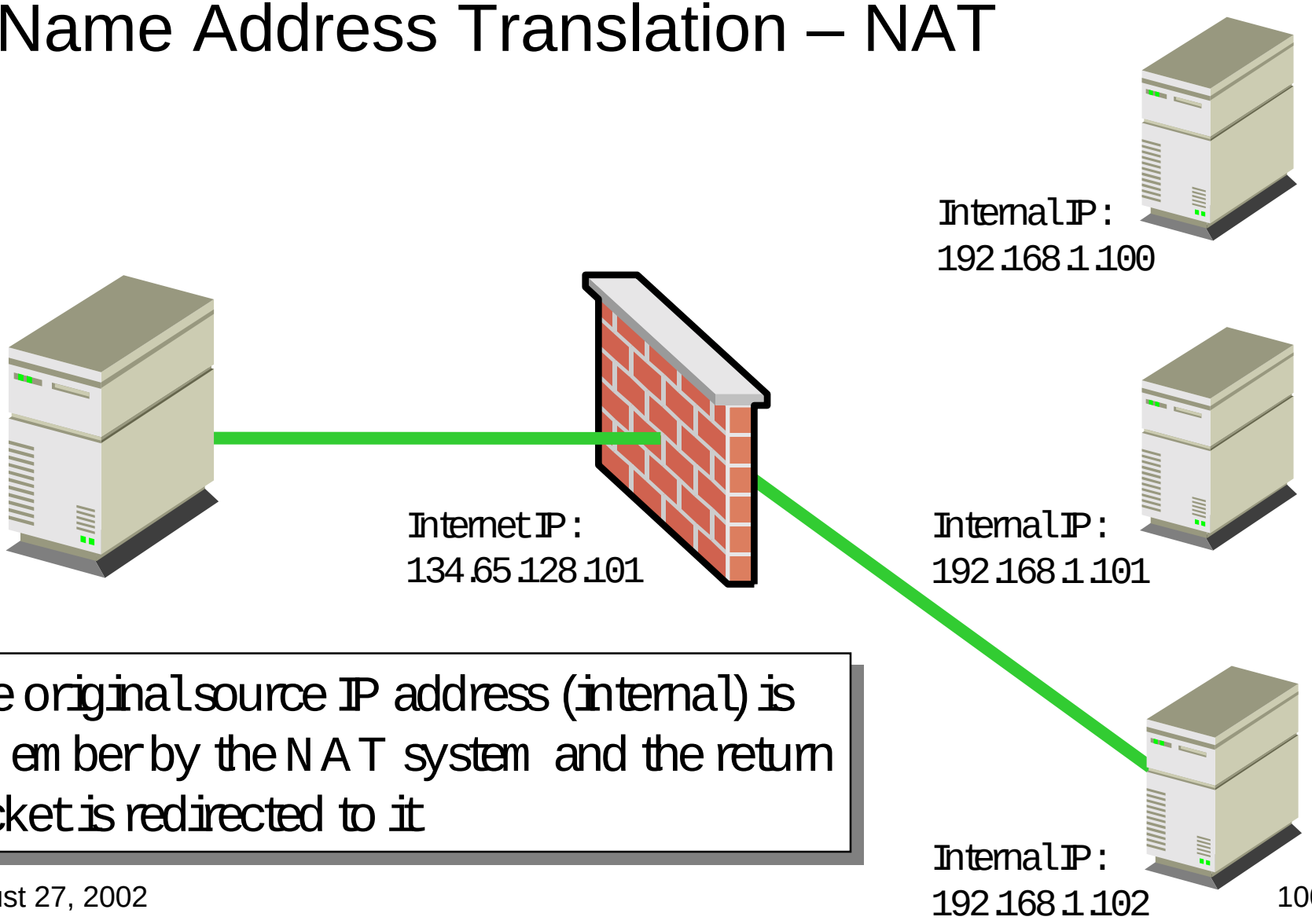
- **Normally, network traffic will travel from a source (such as your home computer) to the destination (such as a web site)**
  - **Through multiple links**
  - **The package is typically forwarded to the next link unaltered**
- **On a system doing Name Address Translation (NAT) the source IP address will be changed to its own – dropping the original**
- **The original source IP address (usually an non-routable internal address) is remembered by the NAT system**
- **Return packets (sent to the NAT system) will be redirected to the correct originating system**
- **This level of indirections make the internal systems non-addressable and protected from direct outside attacks**

Name Address Translation – NAT



All traffic appears to come from the same Internet IP address: 124.65.128.101 – Internal IP addresses can not be addressed from the outside

Name Address Translation – NAT

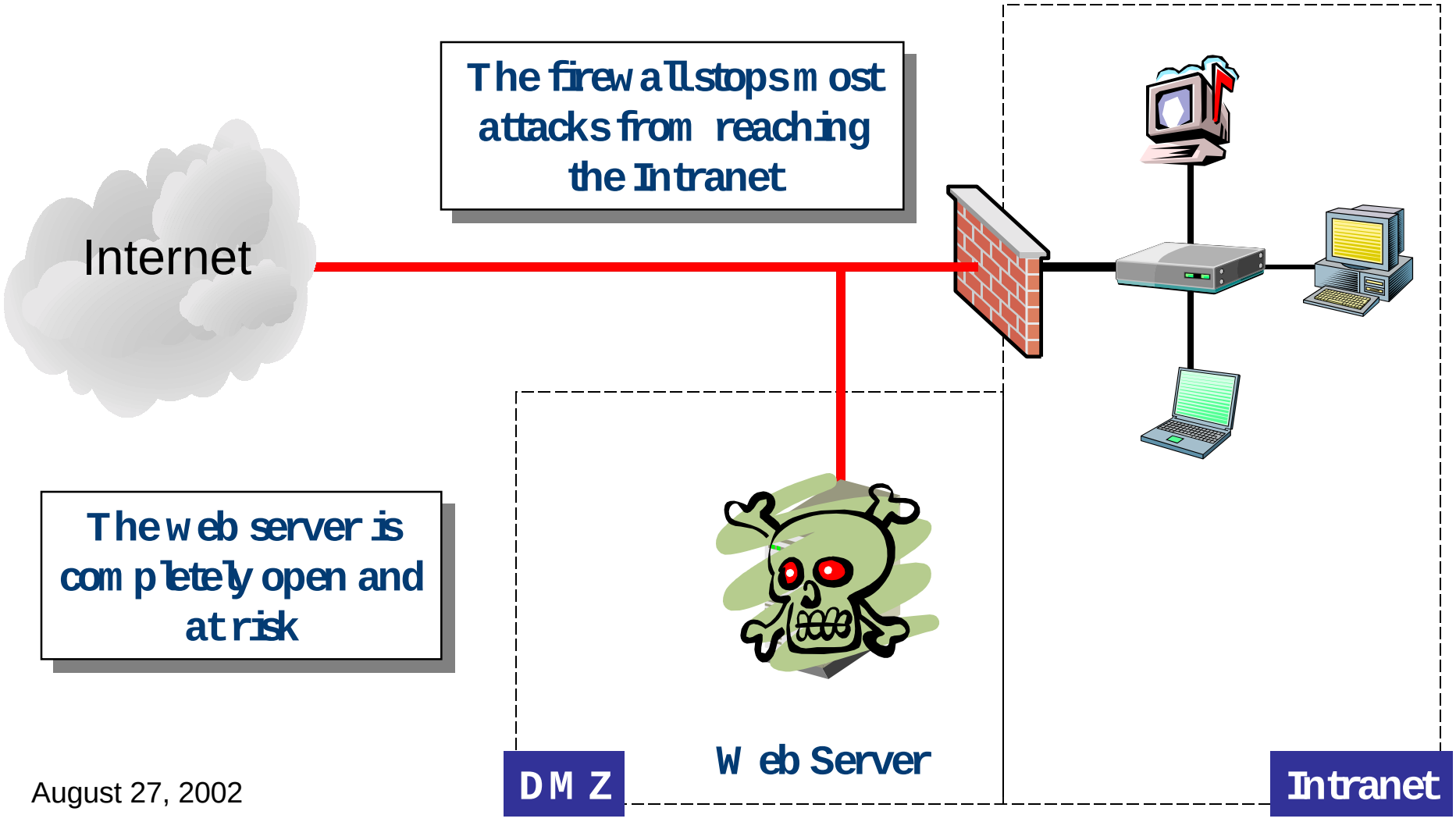


The original source IP address (internal) is remember by the NAT system and the return packet is redirected to it

The First Rules of Firewalls

- **Disable everything**
  - **All incoming and outgoing traffic should be stopped**
- **Slowly allow required network traffic to pass through**
  - **Take this step with great care**
- **The inverse of this is problematic and very dangerous**
  - **Opening up all traffic**
  - **Close that which you don't want**
  - **You will inevitably make a mistake**

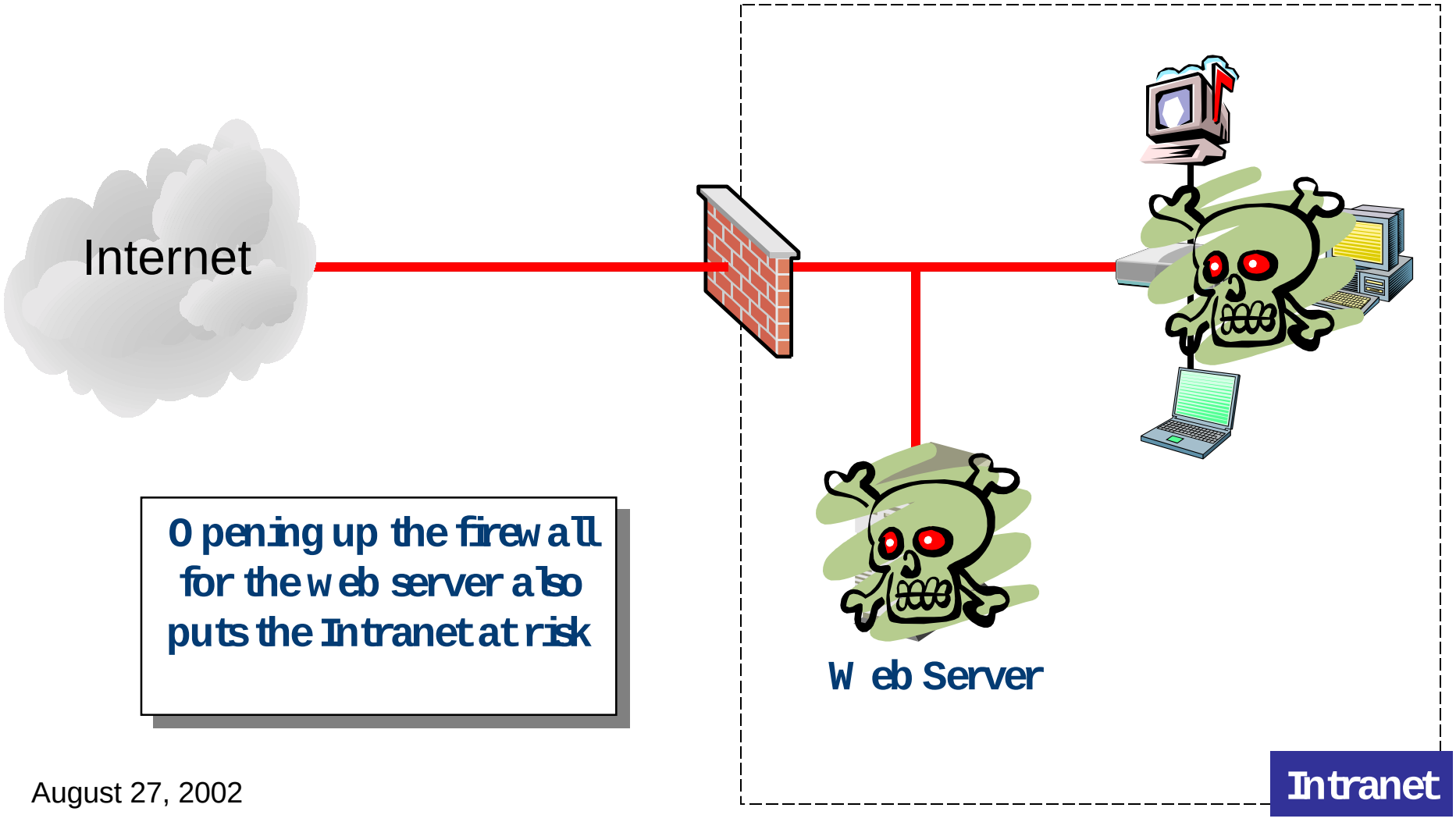
Common Enterprise Firewall Configuration



The Web Server Behind the Firewall

- **A common solution is to place the web server behind the firewall**
  - **The firewall is configured to only allow the specific web related traffic to pass through the firewall**
  - **This traffic is restricted to the firewall only**
- **The problem:**
  - **There are currently tools that can be downloaded from the Internet that allow tunneling attacks through html traffic.**
  - **These could pass directly though the firewall**
  - **If the web server is compromised, the entire Intranet is at risk**

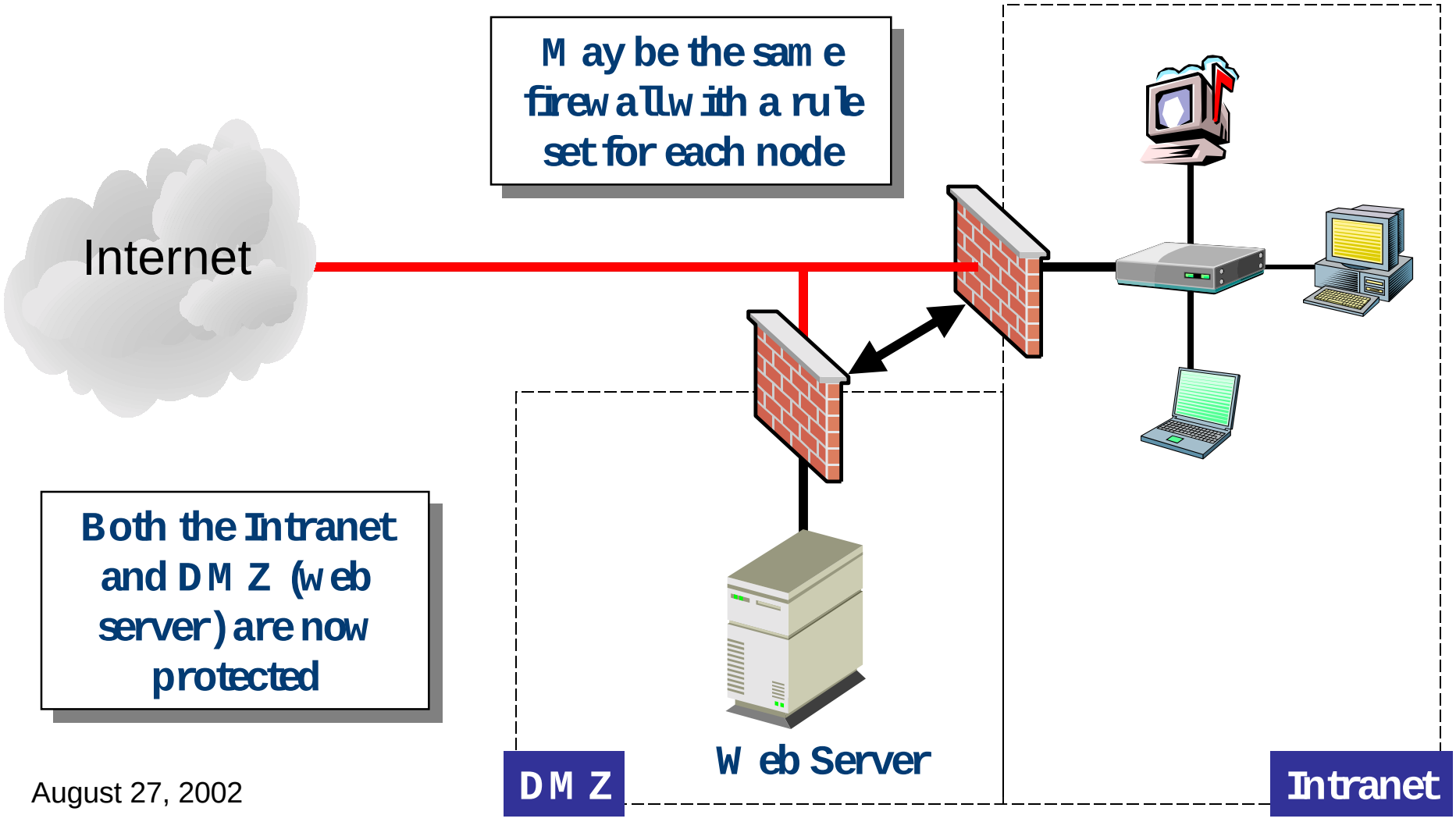
Placing the Web Server in the Intranet



A Better Solution

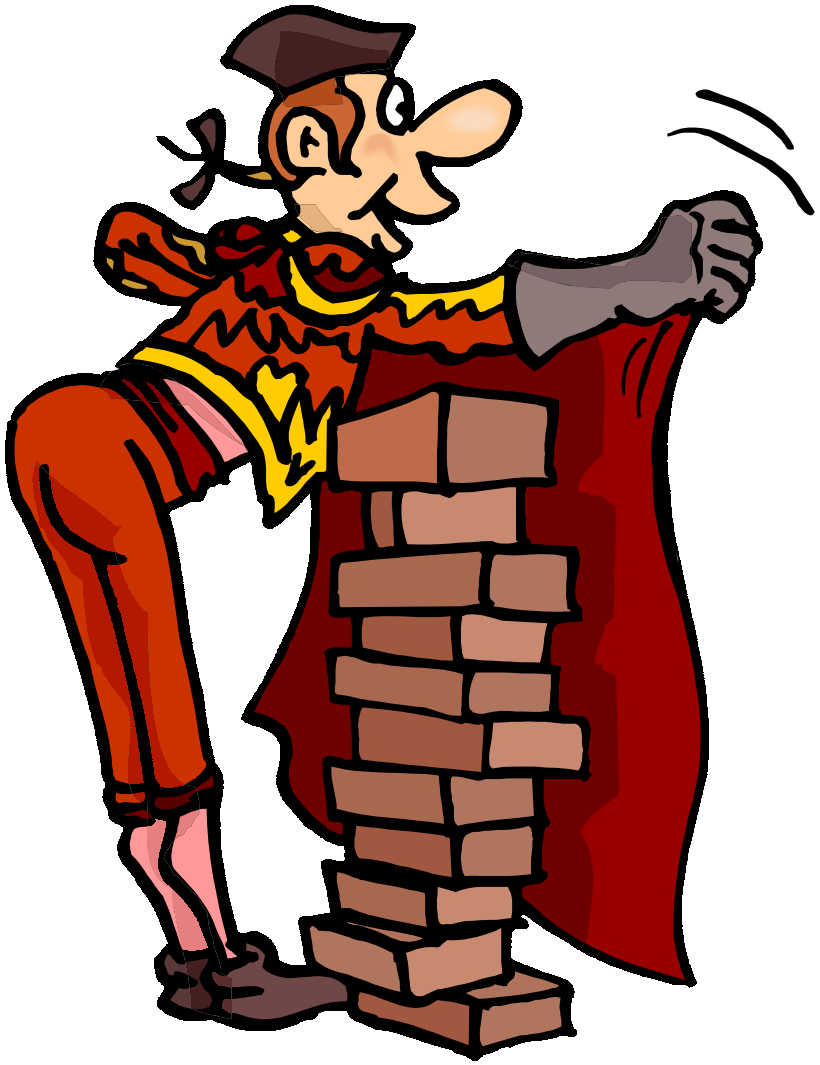
- **A better solution is to use separate firewalls**
  - **One for the Intranet**
  - **Another for the DMZ**
- **Each firewall will have a unique rule set specific to what it is protecting**

Adding a DMZ firewall



Firewalls and Configuration Tools

- **Firewalls**
  - **Ipchains (Linux 2.2)**
  - **Iptables (Linux 2.4)**
  - **Mason**
- **Proxies**
  - **Squid**
  - **SOCKS**
  - **Hogwash**
- **Configuration Tools**
  - **Firestarter**
  - **Kfirewall**
  - **Guarddog**
- **Others**



Delegating Root

Delegating Root Access of the Pitfalls

- **Often it is necessary (especially in larger organizations) to allow others to have root access to your system to perform some task**
  - **For example, users that need to perform regular system backs**
- **The SU (supper user) was written to allow a normal user to elevate their privileges to root by giving the root password**
- **There are some problems with SU**
  - **SU is all or nothing – if you use su to elevate your privilege to root, you have complete access to the system**
  - **They may also modify the system, install new software, backdoor or completely destroy it – not good**
  - **You must give out the root password – this could inadvertently be spread by others, further compromising the security of the system**

Delegating Root With Sudo

- **Sudo (Supper User Do) was developed to help allow an administrator to delegate restricted root access**
  - **Root access is restricted to specific task (commands)**
  - **For example, user who need to perform regular backup procedures are granted root level access to the backup system only**
- **Sudo is called as:**

  sudo [sudo args] command [command args]
- **Control is maintain in a configuration file: “/etc/sudoers”**

Sudo configuration file – sudoers

```
# User alias specification
User_Alias    ADMIN = jim
User_Alias    BACKUP_ADMINS = steve, sue
User_Alias    DEVELOPERS = mark, louis, james

# Host alias specifications
Host_Alias    BACKUP_SYSTEMS = news, mail
Host_Alias    DEV_SYSTEMS = dev1, dev2, redsys

# Command alias specifications
Cmnd_alias    BACKUP = /usr/local/bin/backup

# Users
root          ALL (ALL) = ALL
ADMIN         ALL (ALL) = ALL
BACKUP_ADMINS ALL = BACKUP
DEVELOPERS    DEV_SYSTEMS = /usr/local/test/
```

What Does It Say

- **Those designated ADMIN users are allowed to execute any command on the system – they have full root access**
- **Those users designated BACKUP_ADMINS are allowed to execute the /usr/local/bin/backup command only**
- **Those user designated DEVELOPERS may access the /usr/local/test/ areas on those systems designated DEV_SYSTEMS**

- **WARNING: be very careful when delegating root access with sudo – if you allow a user to run vi as root they may also**
  - **Edit any configuration file on the system - /etc/passwd**
  - **Spawn a shell command with root level privileges**

Intrusion Detection

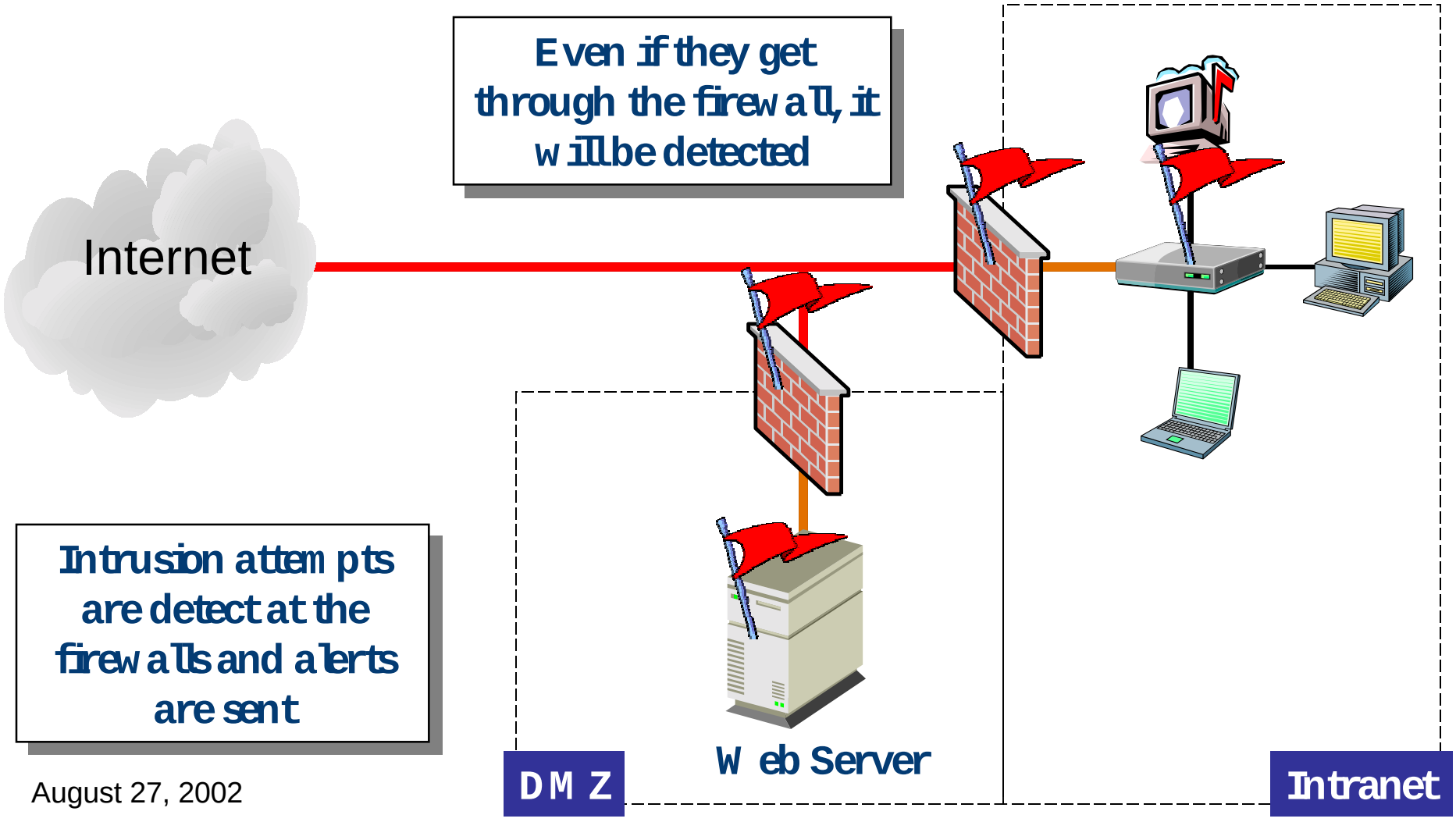
Intrusion Detection Systems (IDS)

- **Looks for evidence that an attack is or has occurred (event)**
- **Gathers all available details about the event**
- **Logs event information**
- **Notify interested parties of the event**

The Intrusion Detection Model (2 types)

- **Network based intrusion detection**
  - **Installed on dedicated server (one per network node)**
  - **Monitors network data on the its visible network (configures system as a network traffic sniffer)**
  - **Identifies data signatures that may identify a known attack**
  - **Early warning system (hints at the possibility of attack)**
- **Host based intrusion detection**
  - **Installed on each system to be monitored**
  - **Monitors systems (logs, files, MS registry, ,,.)**
  - **Advanced systems included client/server management system (event data from one system can be compared with event data from another)**
  - **Provides solid evidence of attacks and abuse**

Intrusion Detection Monitoring



Linux Intrusion Detection System (LIDS)

- **On traditional Unix / Linux systems the root user is exempt from file-system restrictions – root may read any file regardless of access permissions**
- **In the event of a system compromise this can easily lead to additional abuse**
- **The Linux Intrusion Detection System (LIDS) is a Linux kernel patch that will allow users to take away the all-powerful nature of root**
- **They will be able to give programs exactly the access they need, and no more**
- **The root user can be stripped of all his majesty until he is no more powerful than any other user**
- **In the end, it is possible to have a completely functioning system, without worry that some wayward process or malicious cracker can destroy a machine beyond reparability**

Snort

- **Snort is a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks**
  - **It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more**
- **Snort uses a flexible rules language to describe traffic that it should collect or pass**
- **Snort has a real-time alerting capability**
  - **Alerting mechanisms for syslog**
  - **A user specified file**
  - **A UNIX socket**
  - **WinPopup messages to Windows clients using Samba's smbclient**

Snort – Continued

- **Snort has three primary uses**
  - **It can be used as a straight packet sniffer like tcpdump(1)**
  - **Packet logger (useful for network traffic debugging, etc)**
  - **full blown network intrusion detection system.**

Tripwire and other derivatives

- **Tripwire**
  - **Tripwire is a tool that checks to see what has changed on your system**
  - **The program monitors key attributes of files that should not change, including binary signature, size, expected change of size, etc**
- **AIDE (Advanced Intrusion Detection Environment)**
  - **Is a free replacement for Tripwire. It does the same things as the semi-free Tripwire and more**
- **Both create a signed database of file specific information such as owners, groups, file size, file md5 sum, …**
- **If changes are made to a file being monitored, tripwire or AIDE will log or notify the system administrator**

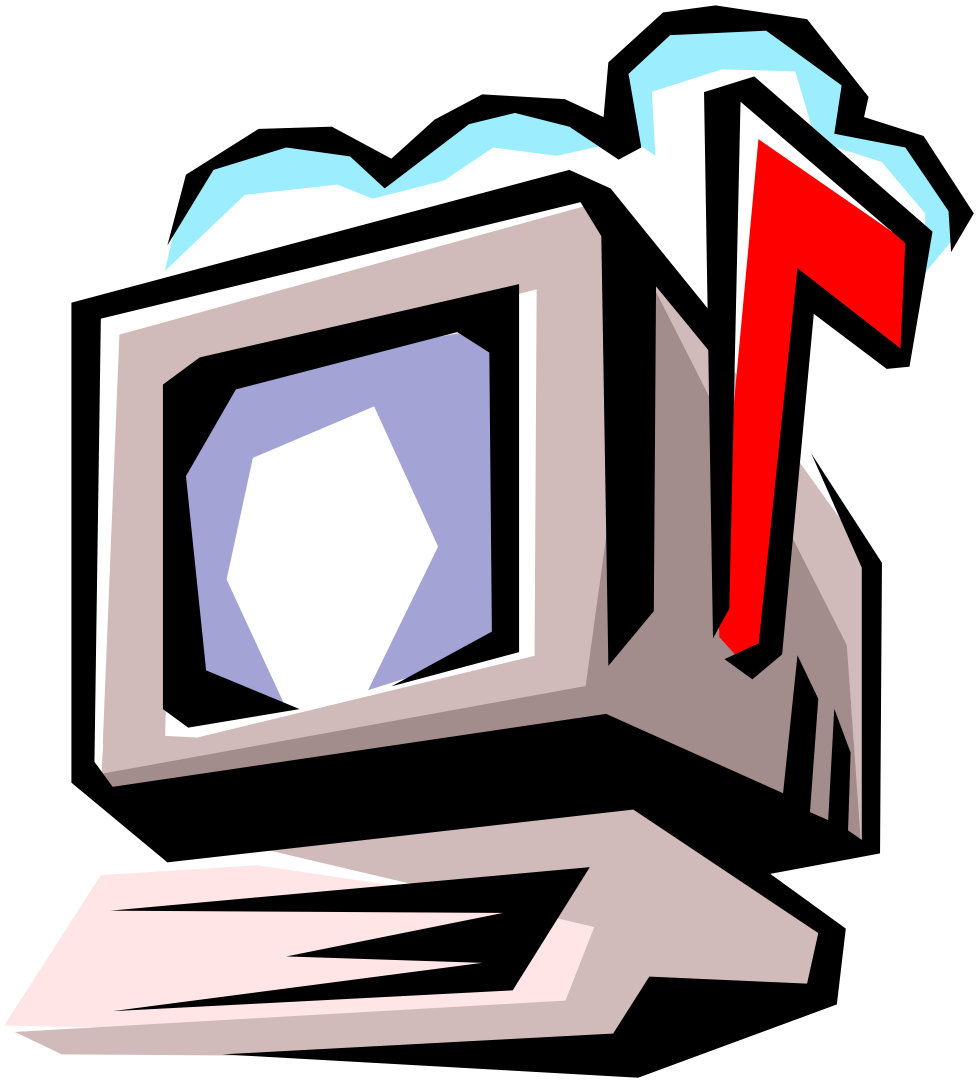
Port Scan Detection – Portscan

- **Portscan monitors network connection attempts**
- **Identifies connection patterns that are indicative of some form of portscan activity**
- **Logs these events**

Log Monitoring

- **There are a number of programs that can be used to monitor system logs**
- **The perform event correlation and notify the system administrator of identified attack signatures**
- **Some of these are”**
  - **Psionic Logcheck**
  - **Color Log**
  - **WOTS**
  - **Swatch**

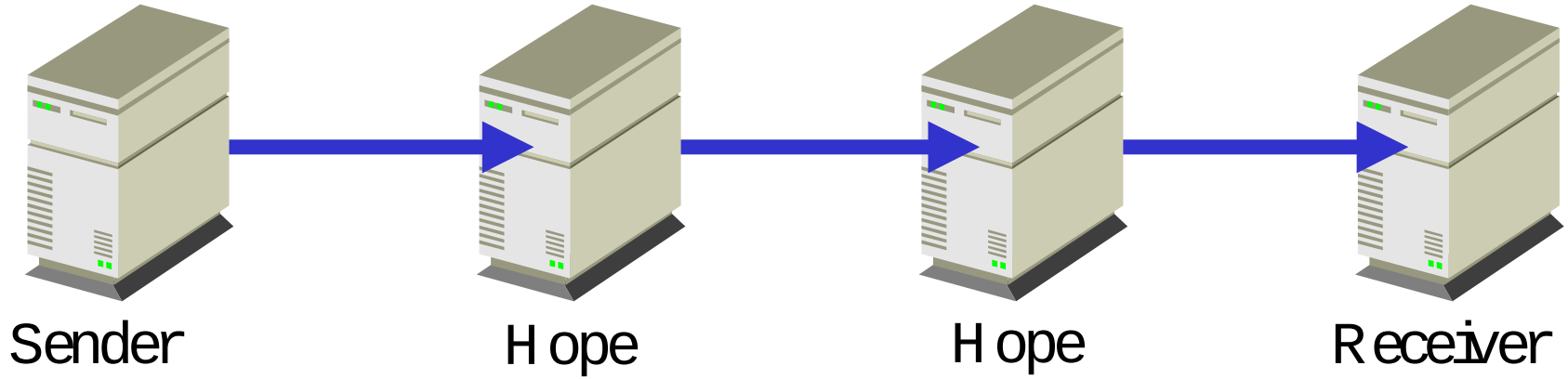
Securing Email With Encryption



The Insecurities of Email

- **Sending email to another party across the Internet must pass across one or more mail hopes**
- **Theses hopes are not under your control and therefore are not to be trusted**
- **Anyone on any of these hopes could intercept and read you email**
- **Do you send confidential email this way?**

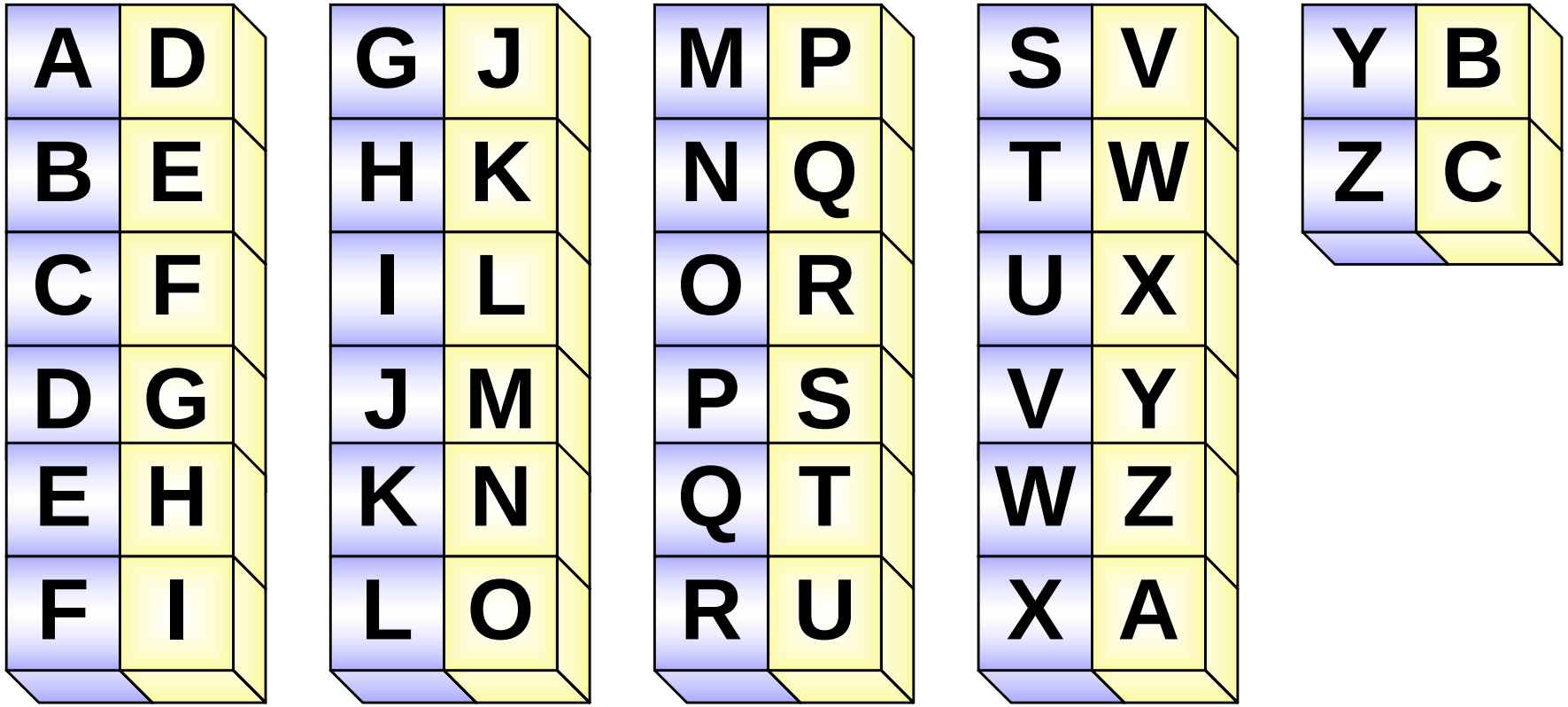
Email Passes Through Multiple Hopes



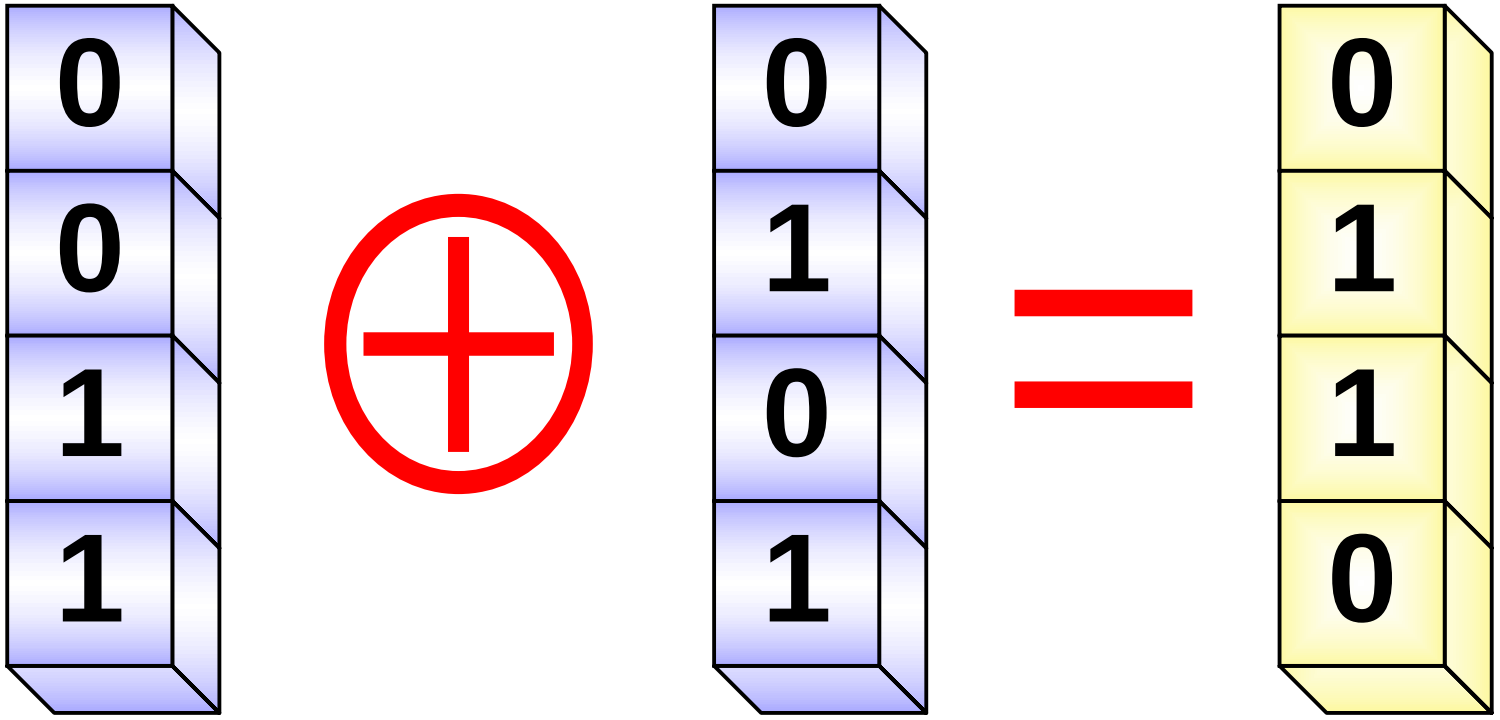
Encrypting Email

- **By encrypting email, you prevent anyone that can intercept it during transit from reading its contents – it will be unreadable**
- **Encryption has has a long history – the substitution cipher was the first form of encryption known to be used**
  - **Julius Caesar derived a for of substitution cipher (known as the Caesar Cipher) to convey secret orders to his generals**
  - **Using the Caesar Cipher, the text "LINUX SECURITY" becomes "OLQXA VHFXULWB"**
- **Another form of encryption is known as XOR Encryption (Exclusive Or)**
  - **The message is XORed with an known seed to produce an obfuscated result**
  - **It is considered a very weak form of encryption**

The Caesar Cipher



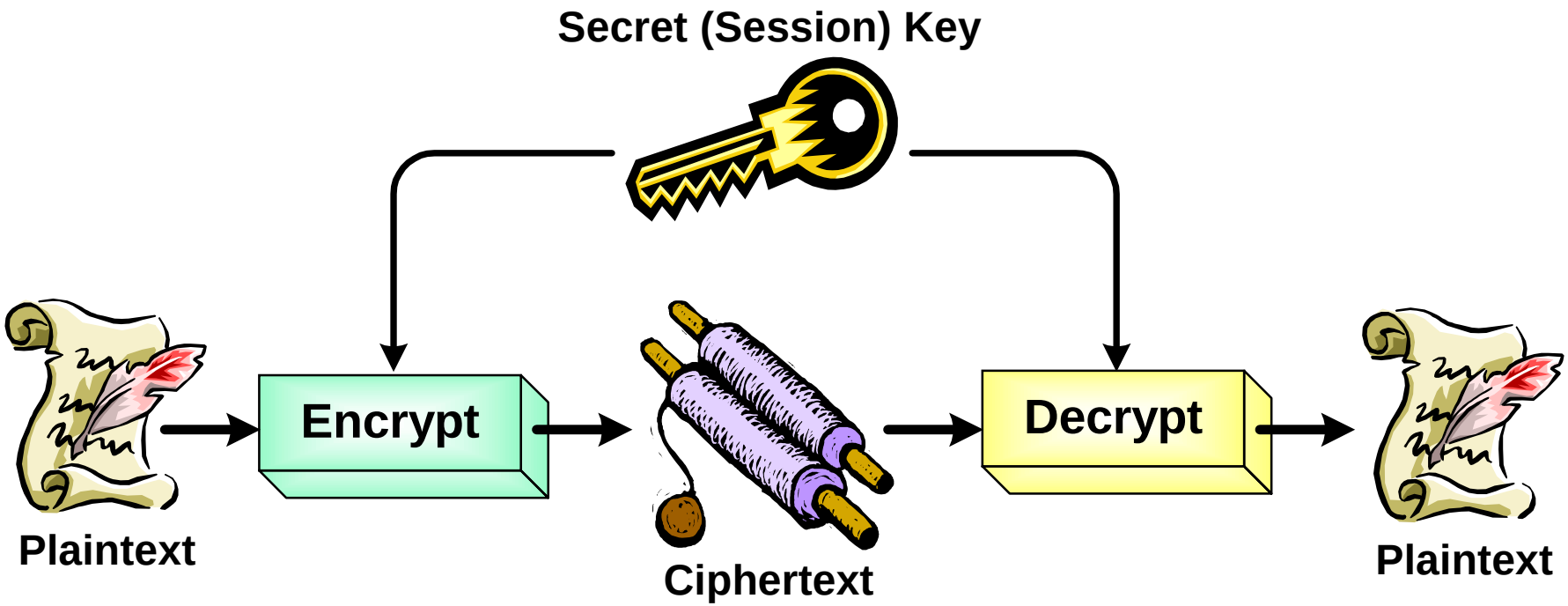
Exclusive OR (XOR)



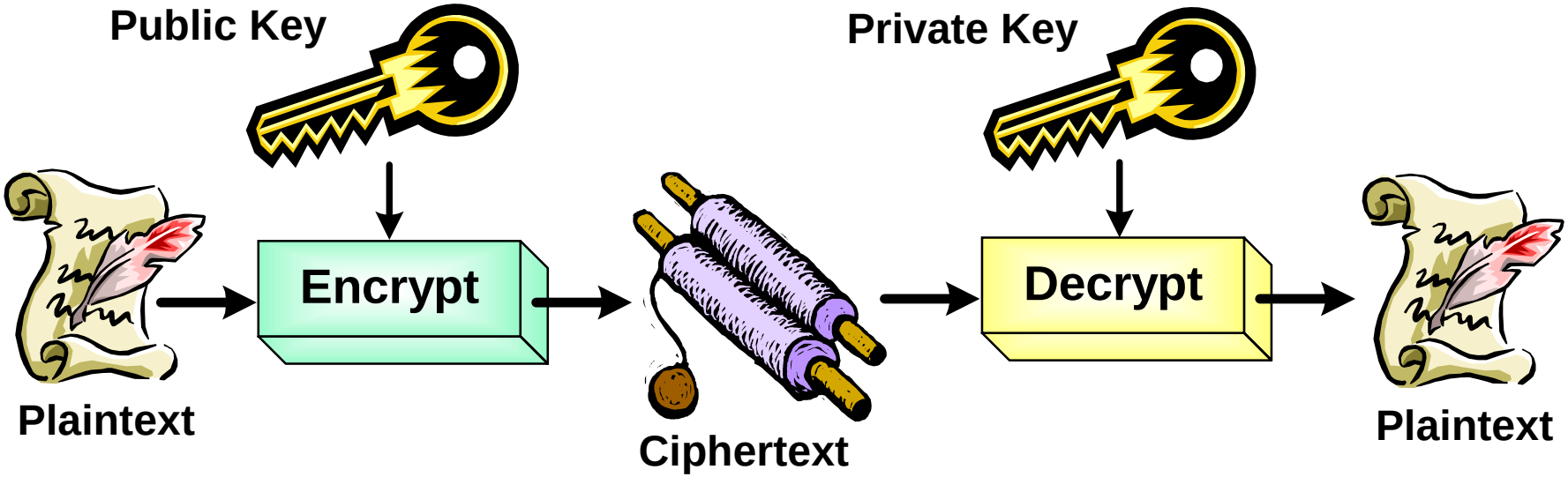
Modern Encryption

- **Modern cryptographic algorithms use keys to identify the mapping being used**
- **These keys are typically measured in the number of bits in the key (key size)**
- **Larger key sizes increase the number of possible possible mappings – decreasing the chance that the cipher will be broken**
- **Modern key-based cryptographic algorithms can be categorized into two types – each has its strengths and weaknesses**
  - **Secret key (symmetric) cryptography uses a single key to encrypt and decrypt messages**
  - **Public key cryptography (asymmetric encryption) uses a key pair – the private key is used to encrypt and the public to decrypt**

Secret (Symmetric) Key Cryptography



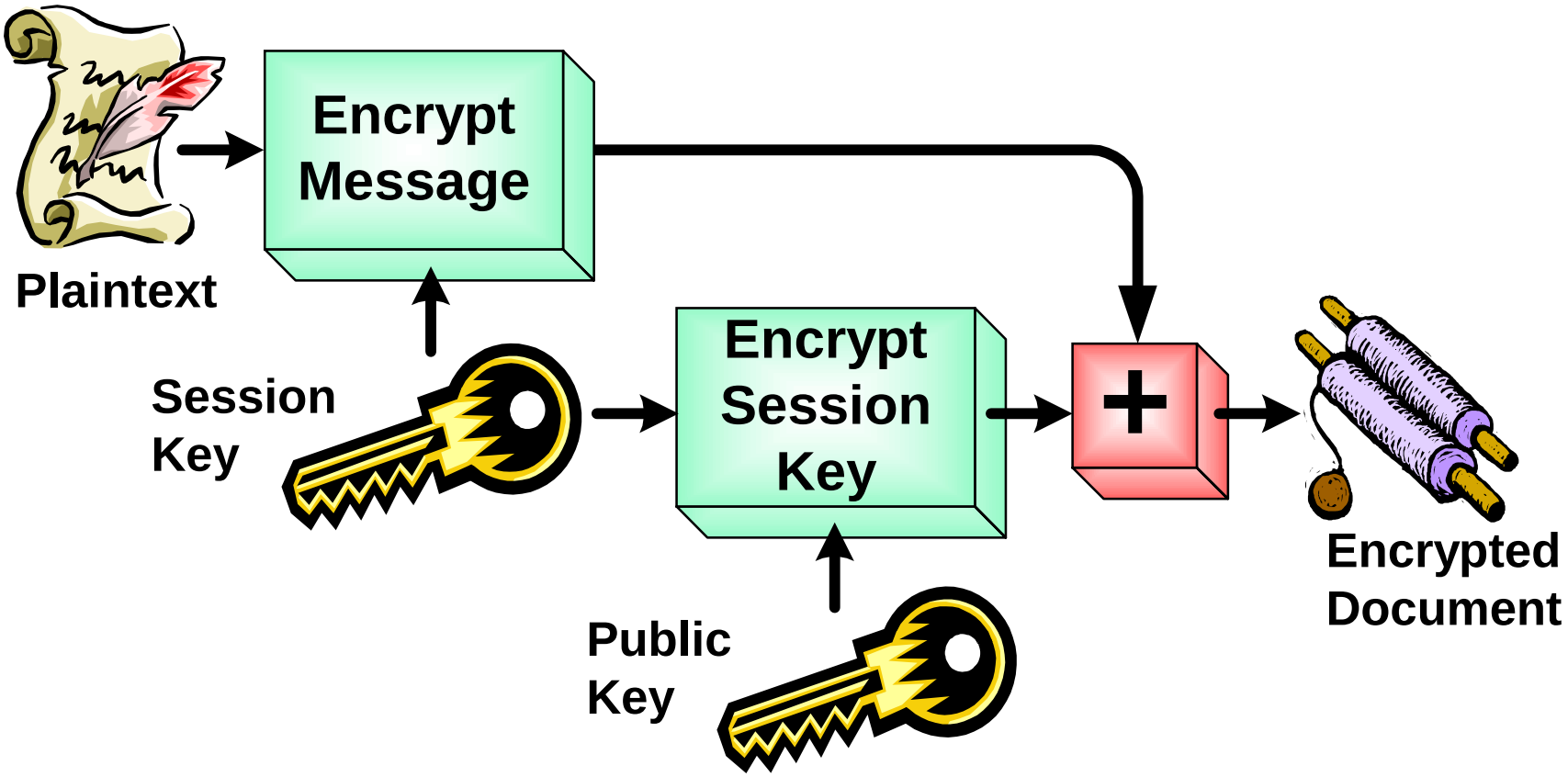
Public Key Cryptography



Encrypting Email

- **PGP (Pretty Good Privacy) was designed to ease the sending of encrypted email**
- **GnuPG (GNU Privacy Guard) was designed as a free replacement for PGP of follows the OpenPGP standard**
- **Both follow the same steps to encrypt email**
  - **Generate a Symmetric Key (Symmetric encryption is substantially faster that public key encryption)**
  - **Encrypt the email message**
  - **Encrypt the symmetric key with the recipients public key and append to the encrypted email message**
  - **Send the encrypted email message**

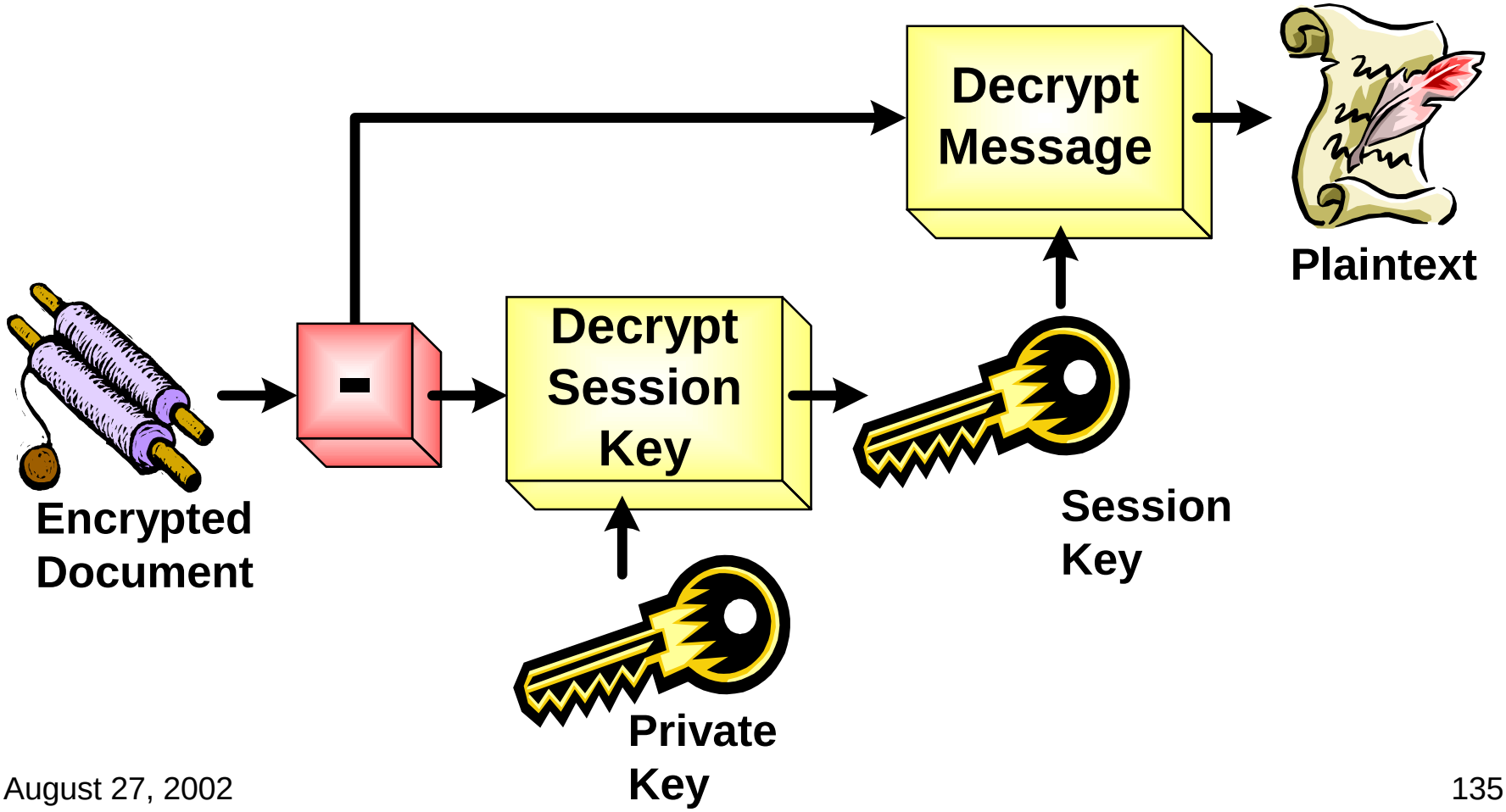
Encrypting Email



Decrypting Email

- **To decrypt an received encrypted email, PGP or GnuPG will perform the following steps**
  - **Detach the encrypted symmetric key from the message body**
  - **Decrypt the symmetric key with the recipients private key**
  - **Use the now decrypted symmetric key to decrypt the email message**
  - **Display the decrypted email message**

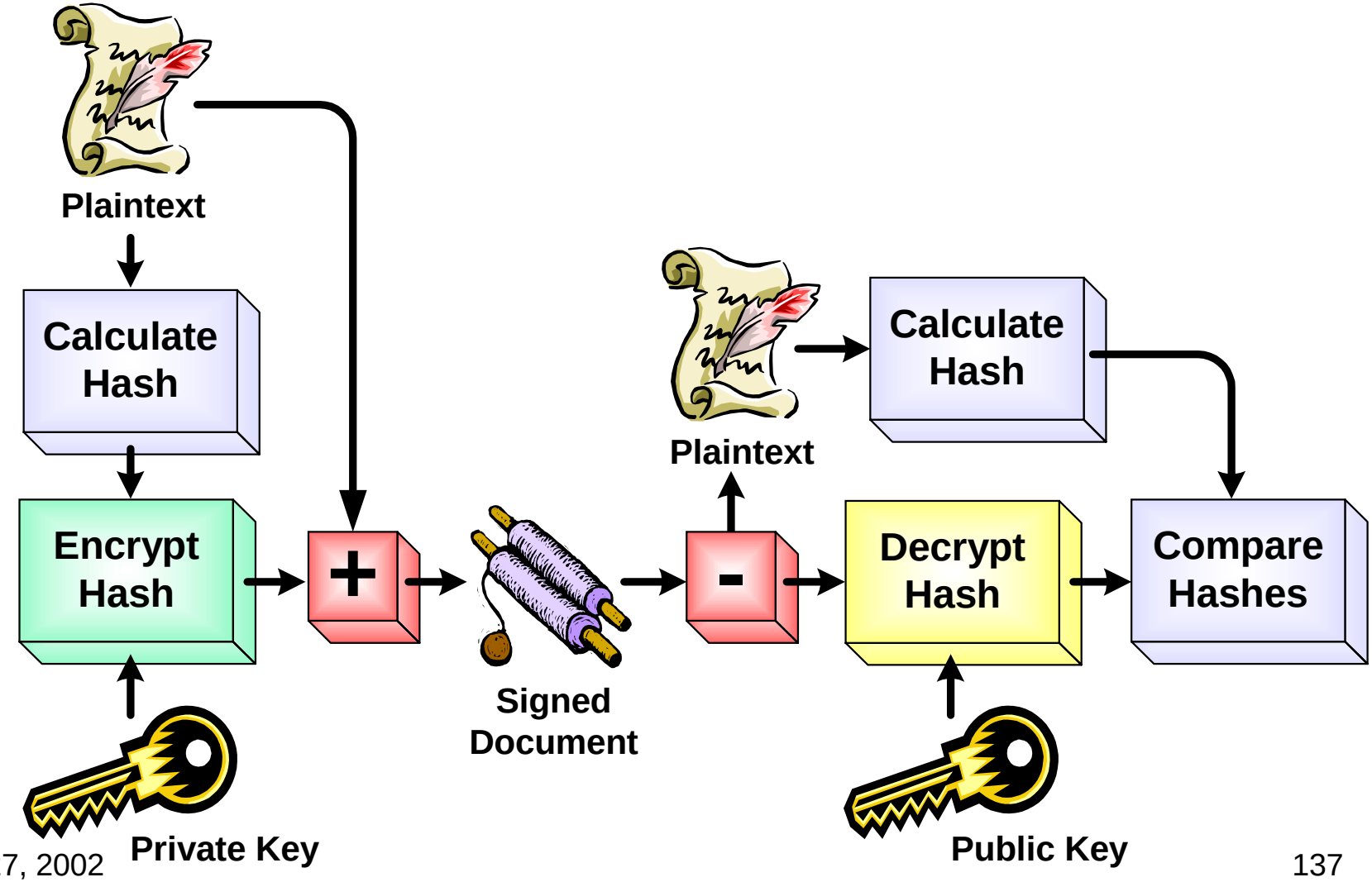
Decrypting Email



Digitally Signing the Email Contents

- **It is possible to digitally sign email contents**
- **This give a level of confidence that:**
  - **The contents have not been modified during transit an**
  - **The message is indeed from the sender and not an imposter**
- **To sign a message the following steps are made to the email message**
  - **Calculate an MD5 checksum of the email message**
  - **Encrypt the MD5 checksum with the senders public key**
  - **Attach the encrypt checksum to the mail message**
- **The following steps are made To verify the signature of for a signed email**
  - **Decrypt the encrypted MD5 checksum using the sender public key**
  - **Verify the decrypted MD5 checksum with the real MD5 checksum of the received message – the signature is valid if they match**

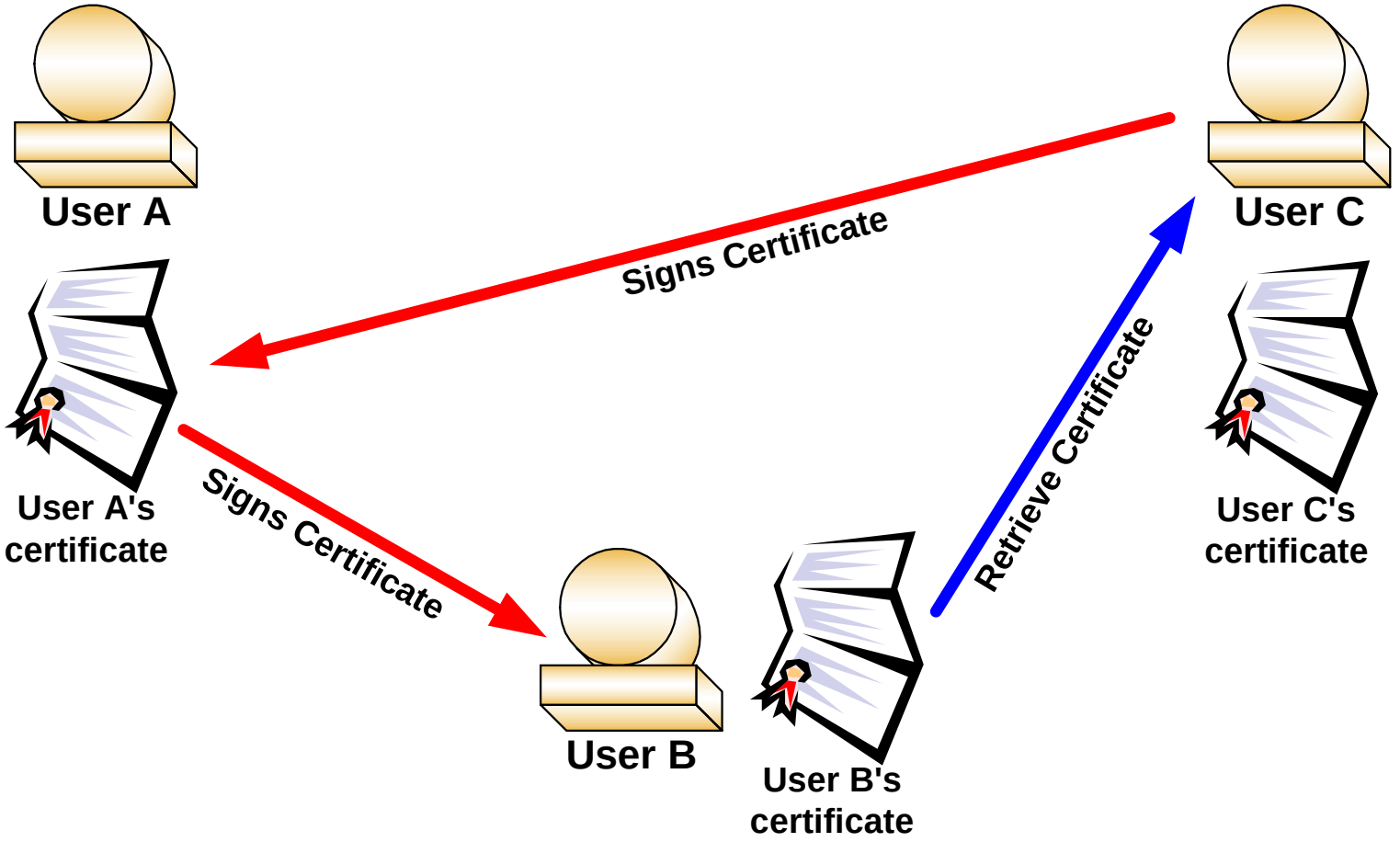
Signing an email message



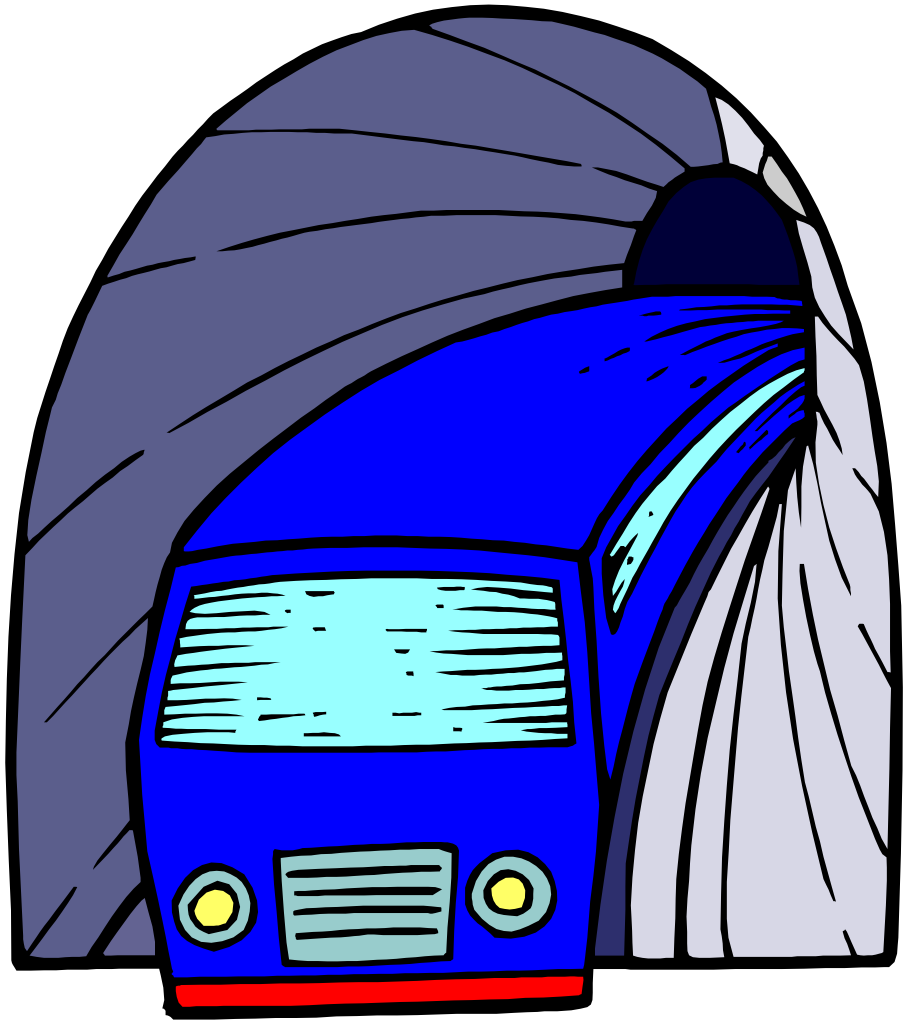
Developing a web of trust

- **For email encryption to be successful, you must distribute your public key to others**
- **There are public keyrings available that allow you to place you public key for others to find**
- **But how do you trust that the public keys that you receive or retrieve from public key servers are valid?**
- **The OpenPGP standard allows you to sign other persons public key**
  - **If you have verified and trust the other persons key**
  - **Others will then see you signature**
  - **If they trust your signature the then can trust this key**

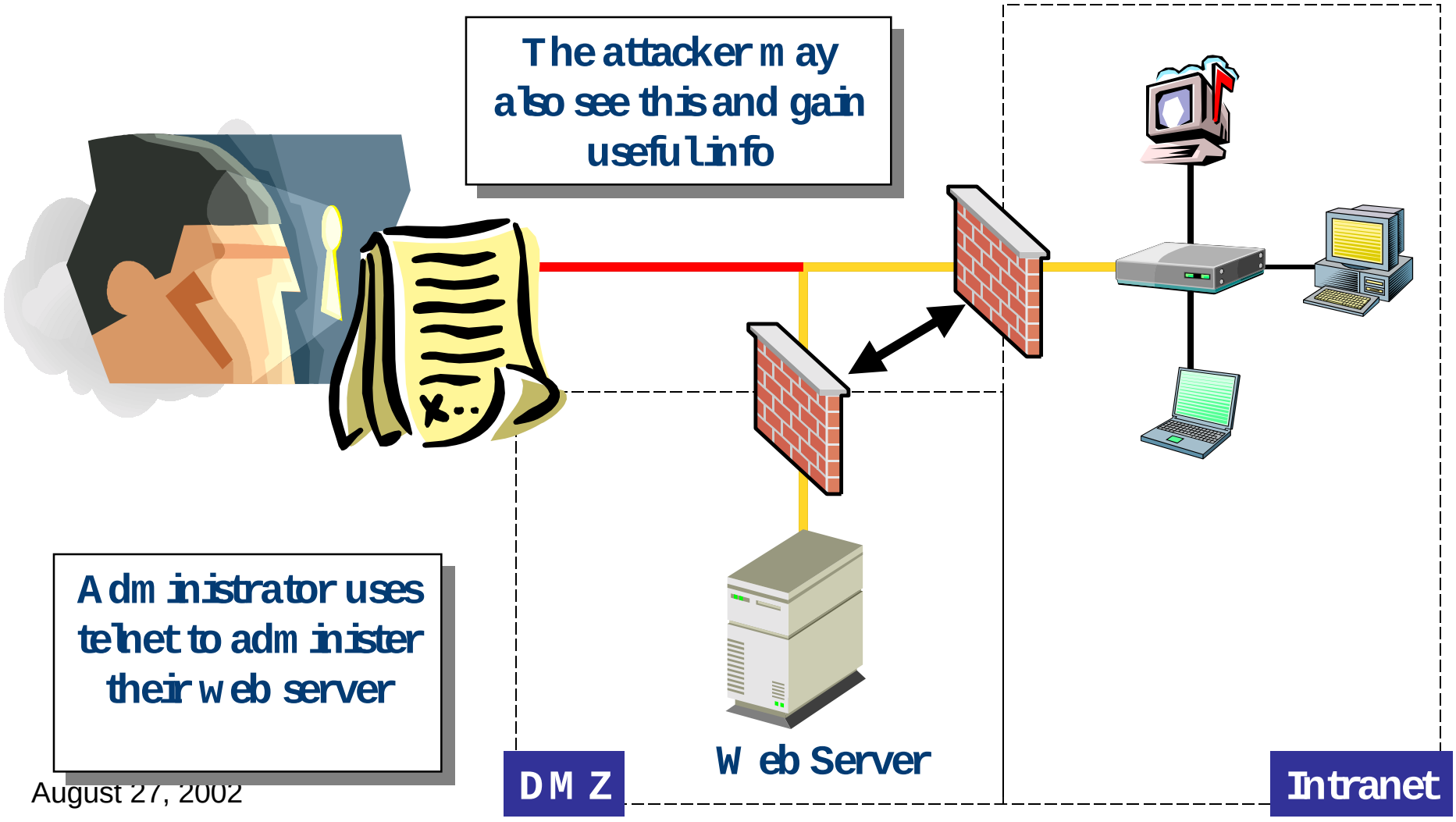
Developing a Web of Trust



Virtual Private Networks



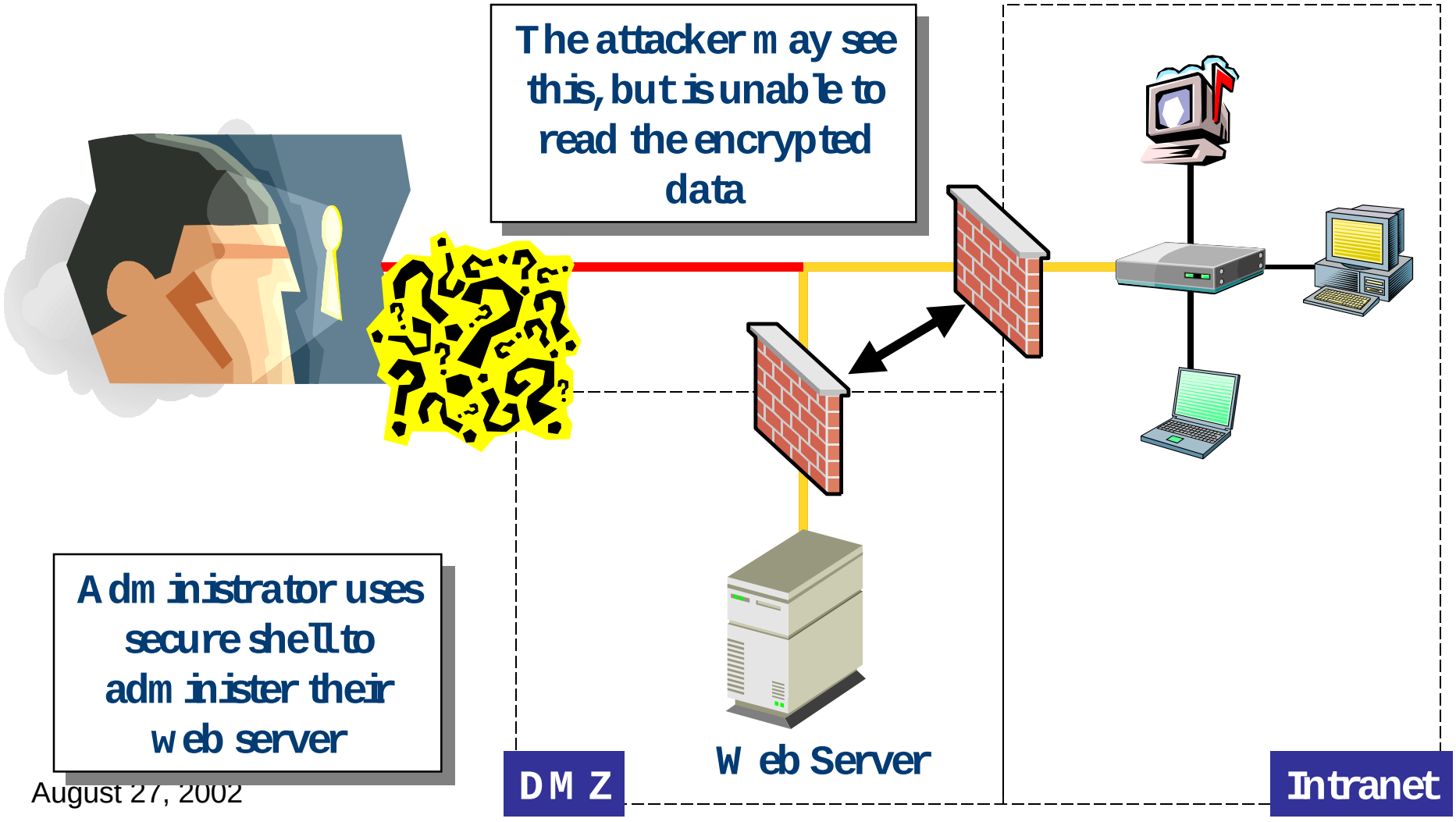
Network Traffic Is Sent in Clear Text



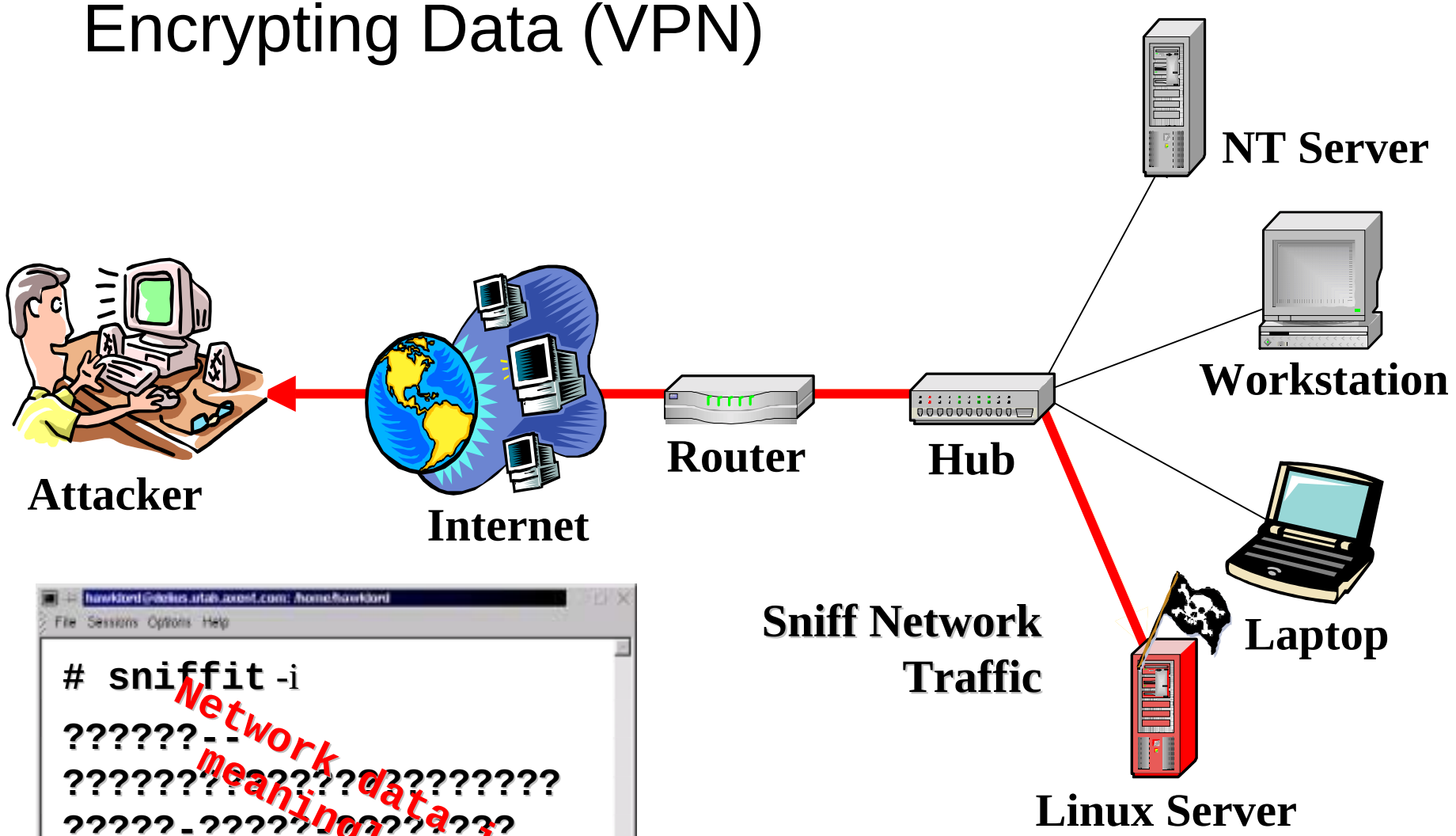
Encryption Is the Key

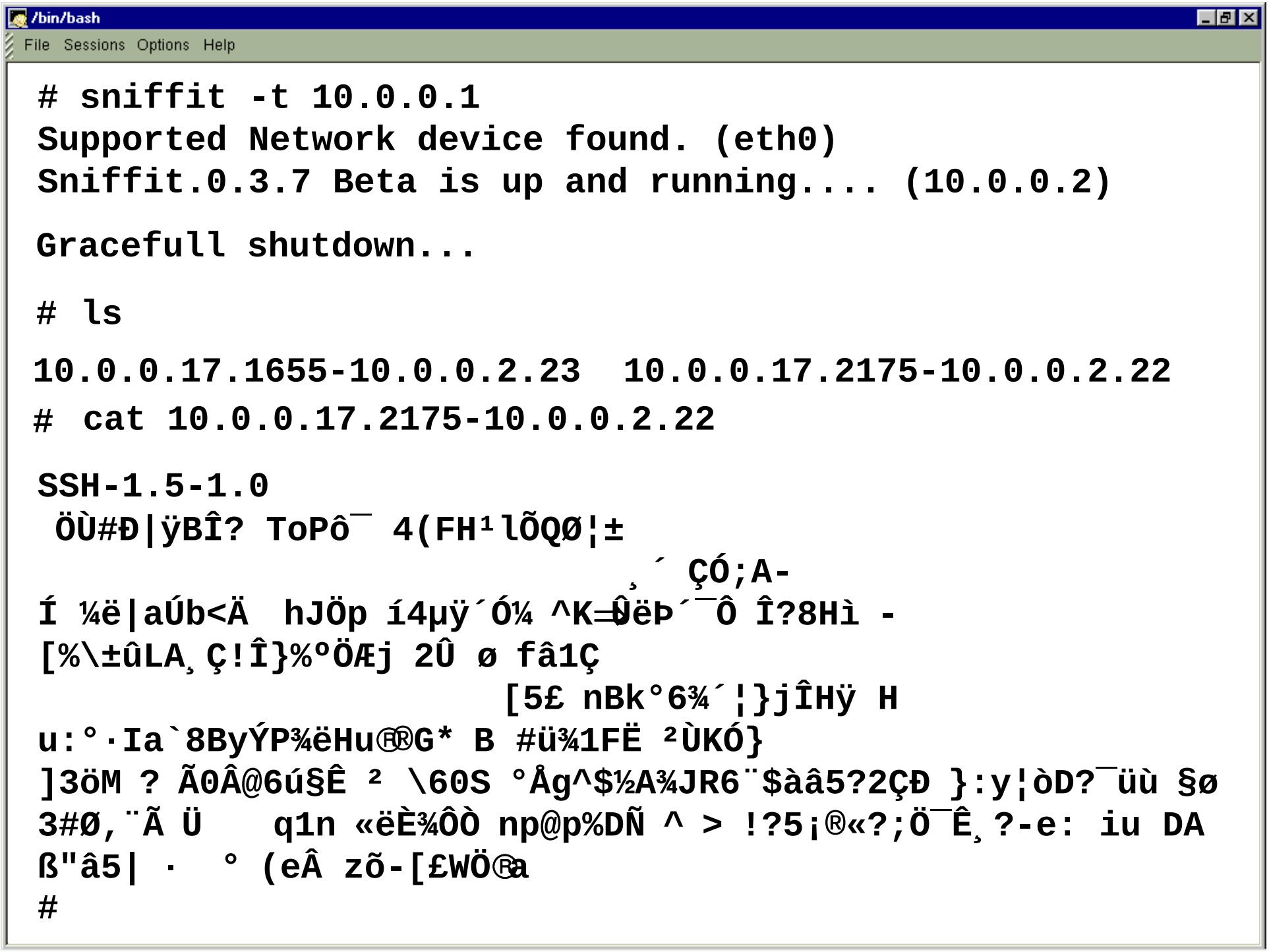
- **Encrypting the data being transmitted will prevent others from understanding the administrative information**
  - **They will still be able to sniff the encrypted data**
  - **It simply will not be readable**
- **For example, one very common tools is the SSH (or OpenSSH) program**

Protecting data with SSH



Encrypting Data (VPN)

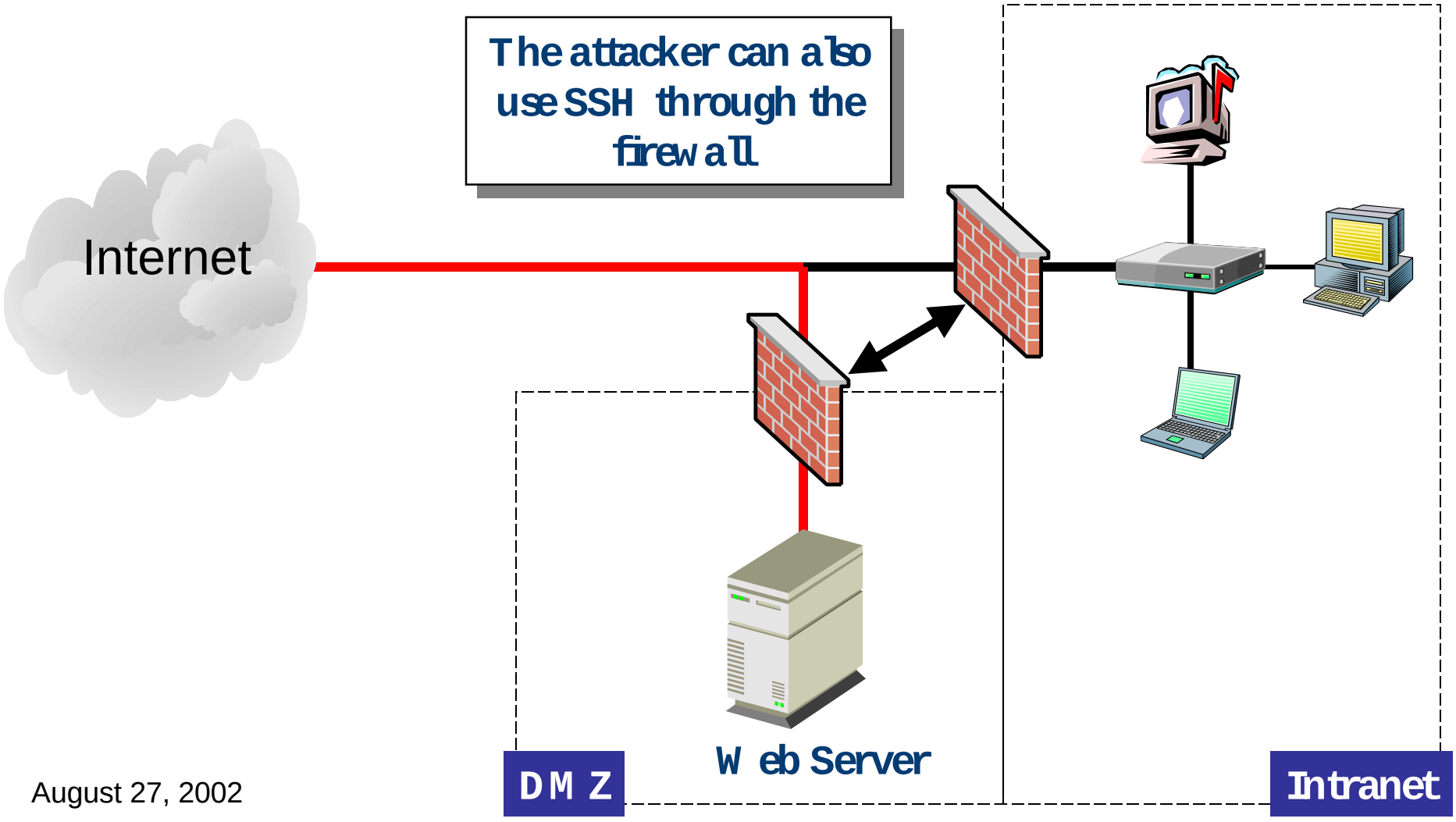



/bin/bash
File Sessions Options Help
sniffit -t 10.0.0.1
Supported Network device found. (eth0)
Sniffit.0.3.7 Beta is up and running.... (10.0.0.2)
Gracefull shutdown...
ls
10.0.0.17.1655-10.0.0.2.23 10.0.0.17.2175-10.0.0.2.22
cat 10.0.0.17.2175-10.0.0.2.22
SSH-1.5-1.0
ÖÙ#Đ|ÿBÎ? ToPô¯ 4(FH¹lÕQØ¦±
¸´ ÇÓ;A-
Í ¼ë|aÚb<Ä hJÖp í4µÿ´Ó¼ ^K=ÛëÞ´¯Ô Î?8Hì -
[%\±ûLA¸Ç!Î}%ºÖÆj 2Û ø fâ1Ç
[5£ nBk°6¾´¦}jÎHÿ H
u:°·Ia`8ByÝP¾ëHu®®G* B #ü¾1FË ²ÙKÓ}
]3öM ? Ã0Â@6ú§Ê ² \60S °Åg^$½A¾JR6¨$àâ5?2ÇÐ }:y¦òD?¯üù §ø
3#Ø,¨Ã Ü q1n «ëÈ¾ÔÒ np@p%DÑ ^ > !?5¡®«?;Ö¯Ê¸?-e: iu DA
ß"â5| · ° (eÂ zõ-[£WÖ®a
#

Issues With SSH

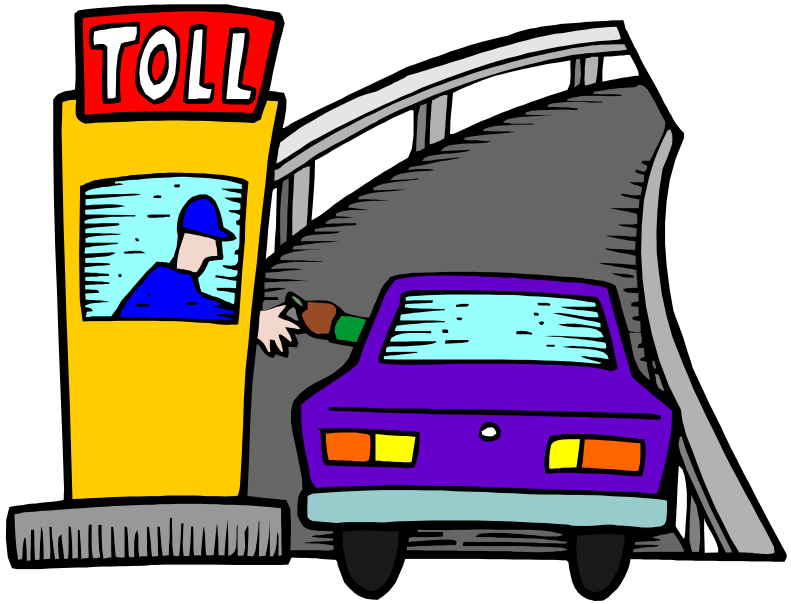
- **SSH (and OpenSSH) is an excellent program**
- **It provides good encryption and authentication**
- **Unfortunately its use in this situation does require that you open your firewall to allow SSH traffic through**
  - **There have been a number of SSH vulnerabilities discovered that that can lead to compromise**

Using SSH


The attacker can also use SSH through the firewall
Internet
Web Server
DMZ
Intranet

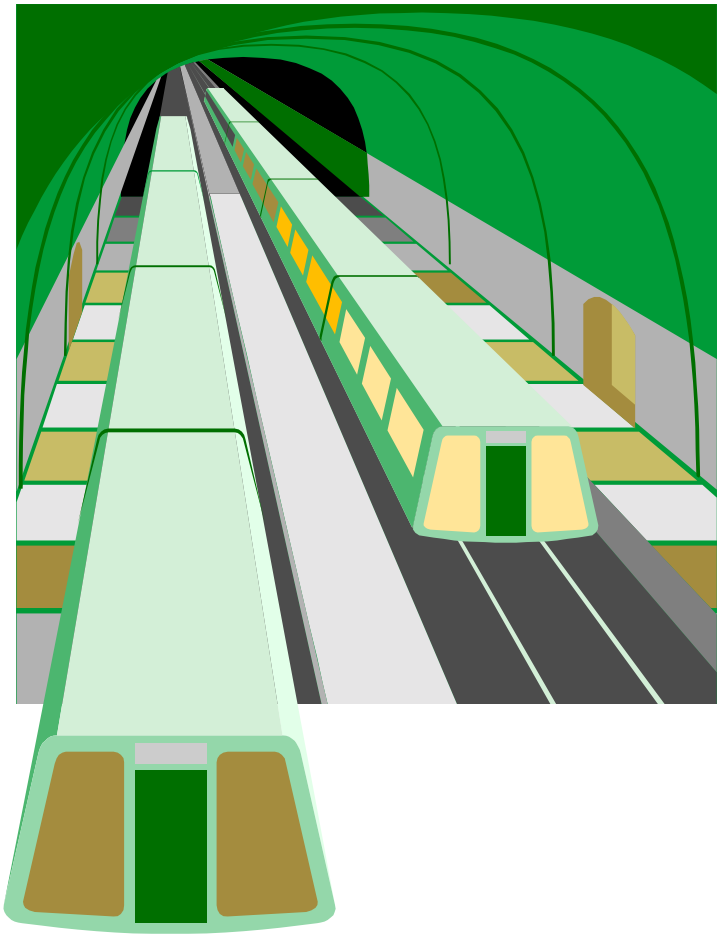
Virtual Private Network (VPN) to the Rescue

- **The use of a Virtual Private Network (VPN) provides a more secure alternative**
- **It can provide strong authentication at the firewall**
  - **You will still need to open up the fire wall to allow VPN traffic**
- **Only authorized traffic will be allowed through the firewall to the web server**

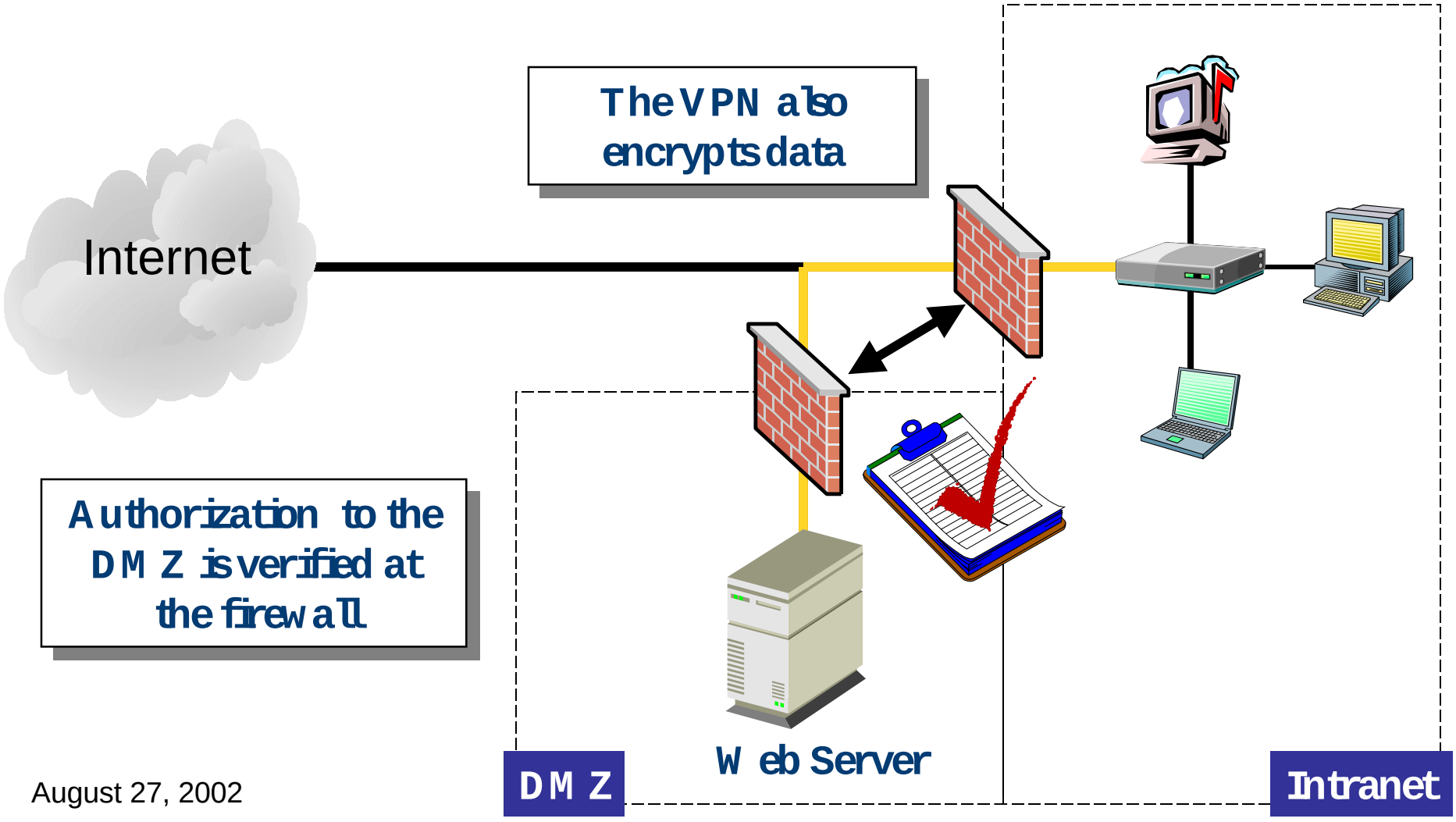


What VPN's are available

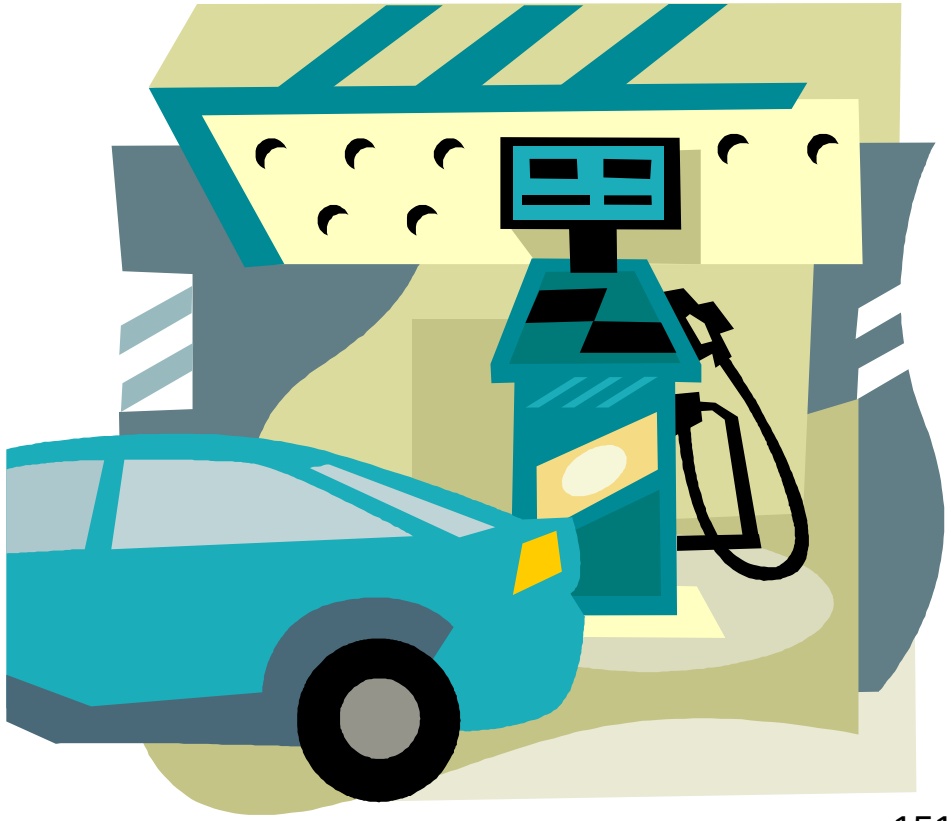
- **ssh**
- **Vpnd**
- **Free / SWAN**



Using a Virtual Private Network (VPN)



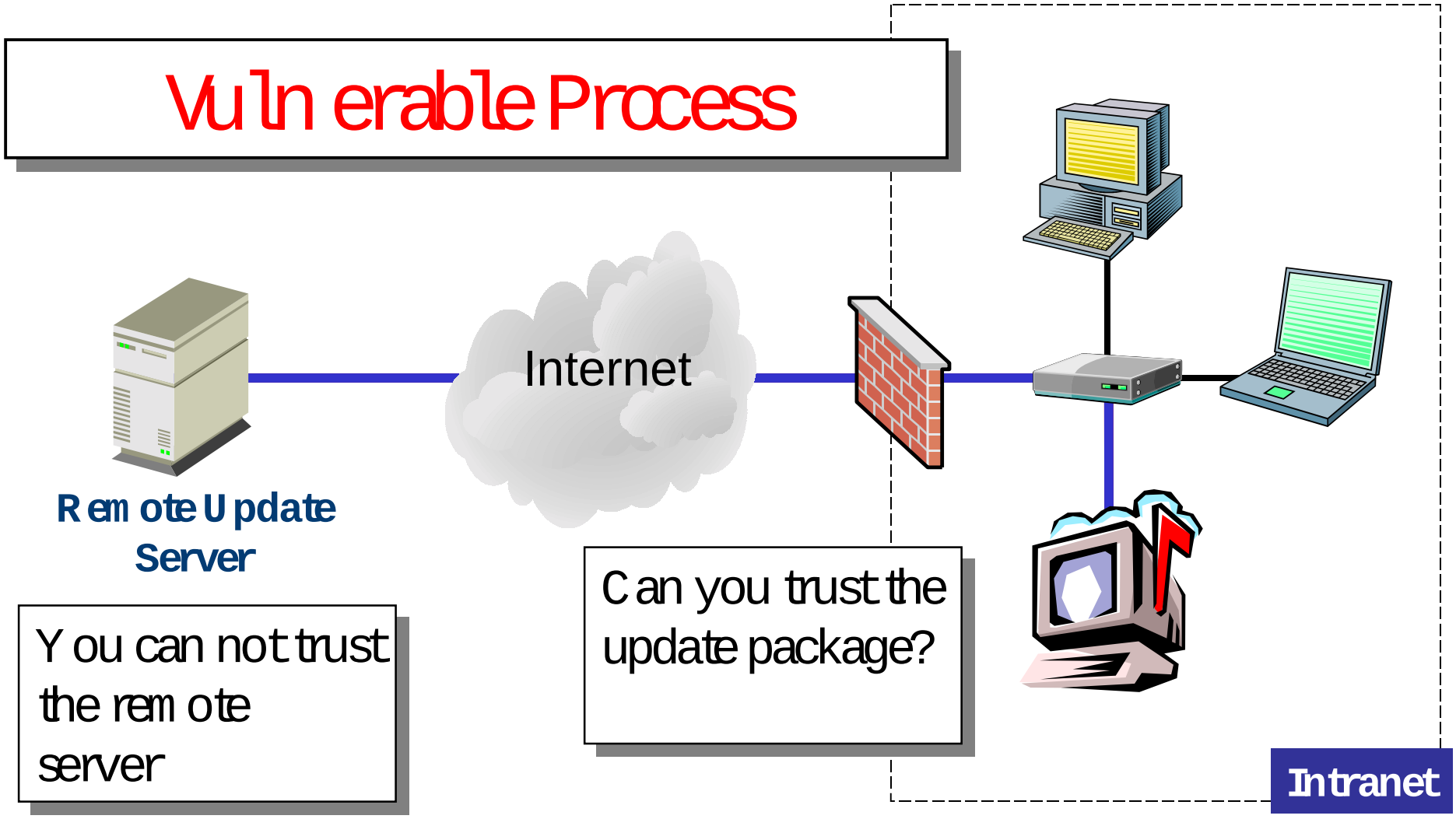
Keep it Updated



The need to keep your system updated

- **When a new version of Linux is released by a vendor, it will usually contain the latest versions of each software package**
- **Over time vulnerabilities will generally be found for multiple software packages**
- **The vendor will respond to this vulnerability by providing an updated version of the software package for download**
- **It is then up to you to download these updates and apply them to your system**
- **Failure to do this will leave your system vulnerable to attack**
- **Currently there are three types of packages in use — Red Hat Packages (RPM), Debian Packages (DEB) and tar archives generally compressed with gzip or bzip2**

Downloading and Installing updates

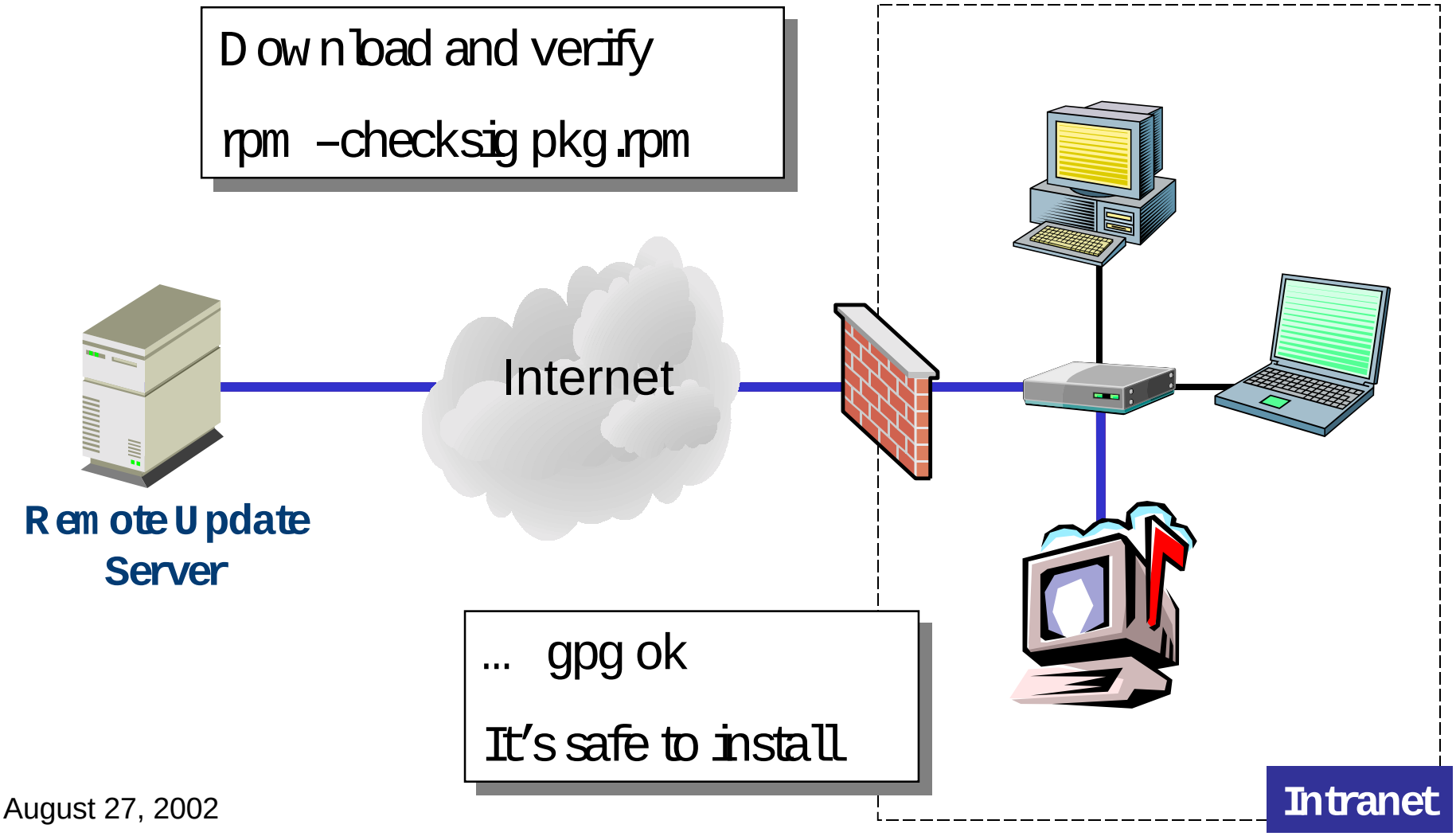


Verifying Update Packages

- **The problem**
  - **The remote server is also at risk from attack and therefore it's content is also at risk**
  - **Update packages can be modified by attackers**
  - **Users may download modified packages that include a backdoor or other hostile code**
- **RPM packages can be signed by the vendor or other third party**
  - **Based on a md5 hash of the package contents**
  - **Allows the user to verify the package source and content integrity**
  - **If the package is modified the signature will not verify**

    rpm –checksig package.rpm
- **Debian and tar packages currently lack this capability and therefore will never be able to obtain the same level of trust**

Downloading and Installing updates

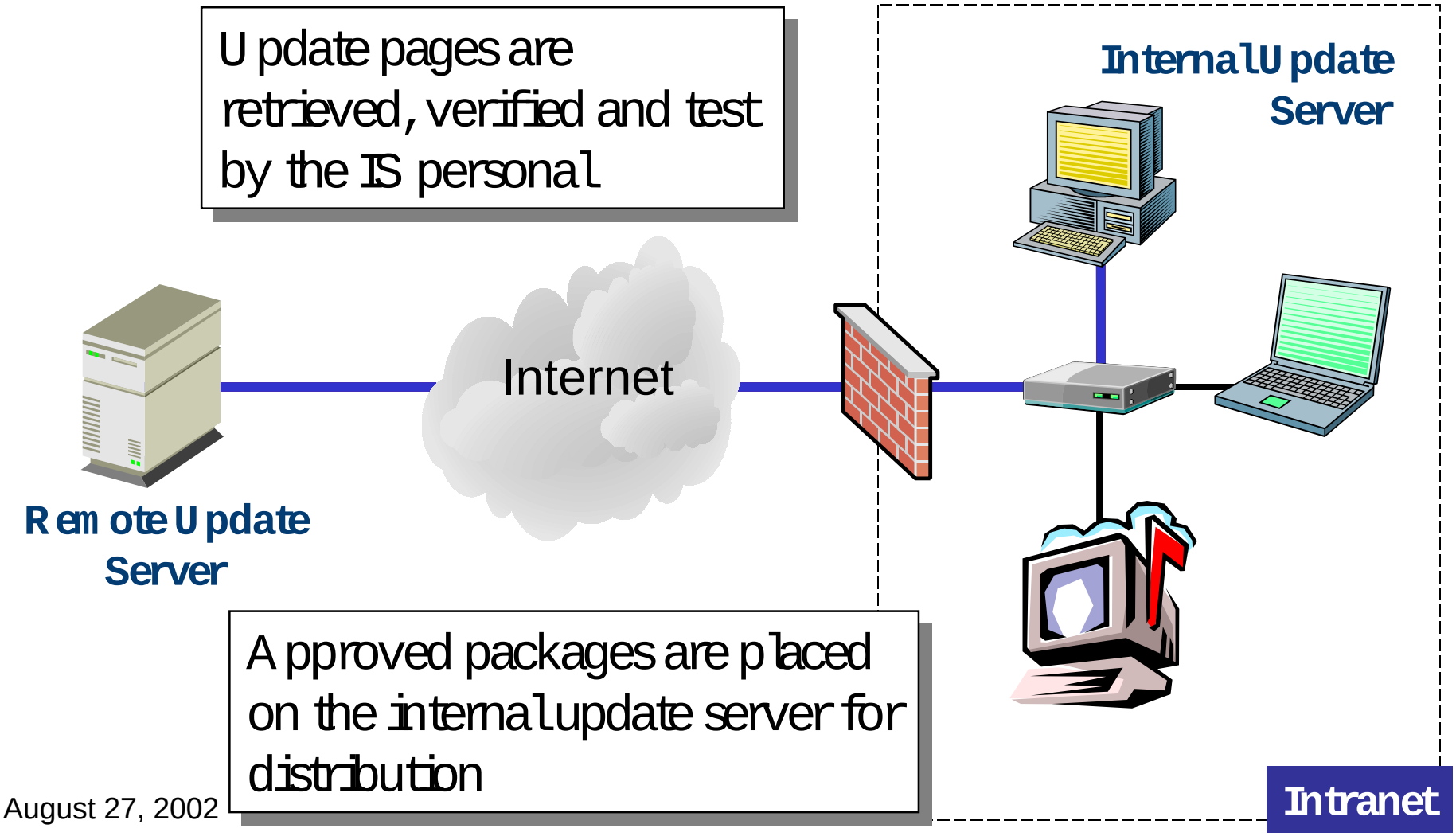


Large scale update management

- **Vender QA time on update packages is usually much less than that performed prior to a distribution release**
- **There is a significant larger chance that an update could potentially break other functionality in an unpredictable way**
- **For this reason most IS departments will wish to test update packages before distributing them to others**
- **An internal ftp server can be used to distribute approved packages**
- **The package can be signed by the IS department**

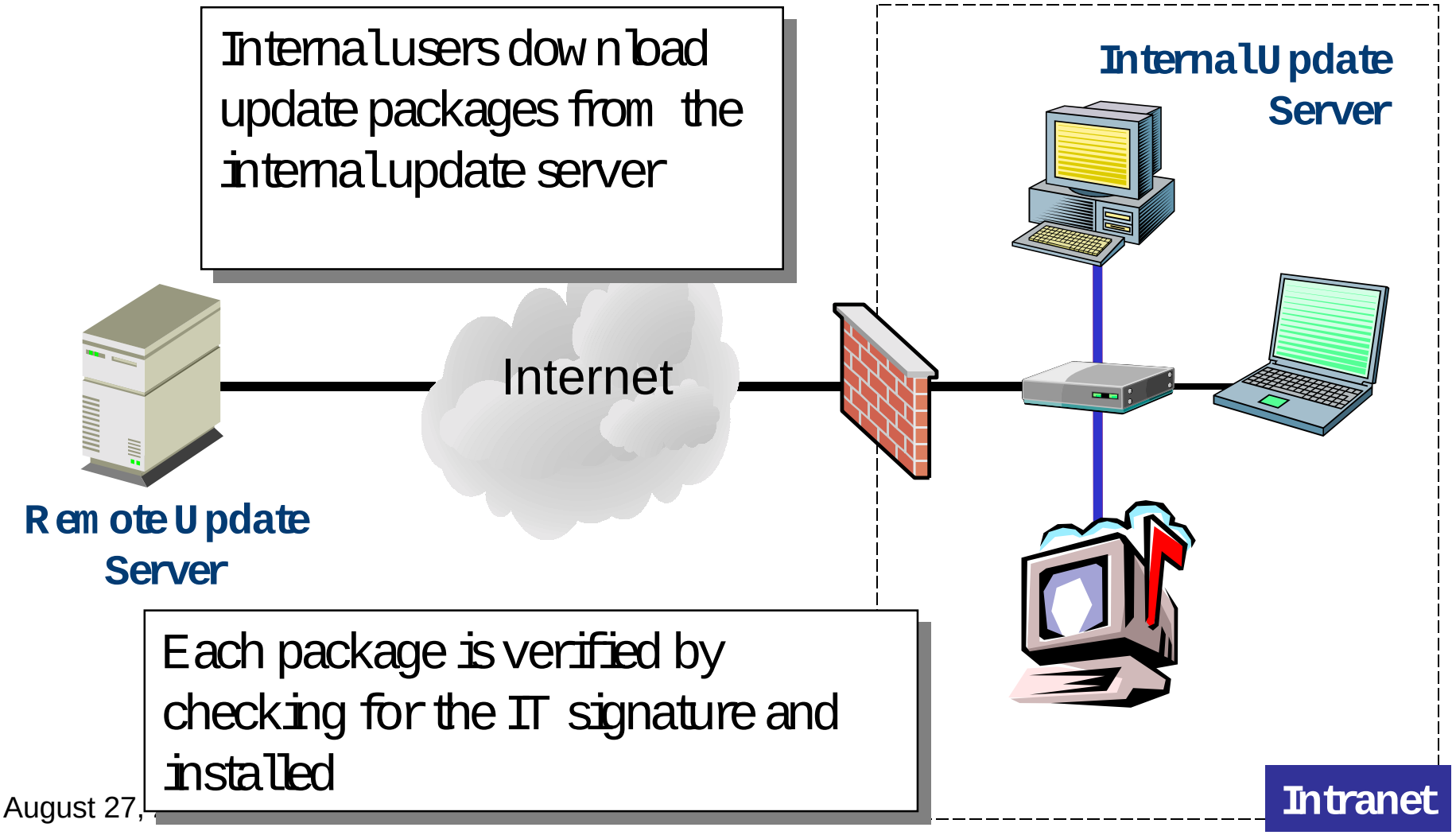
Downloading and Installing Updates

Update pages are retrieved, verified and test by the IS personal



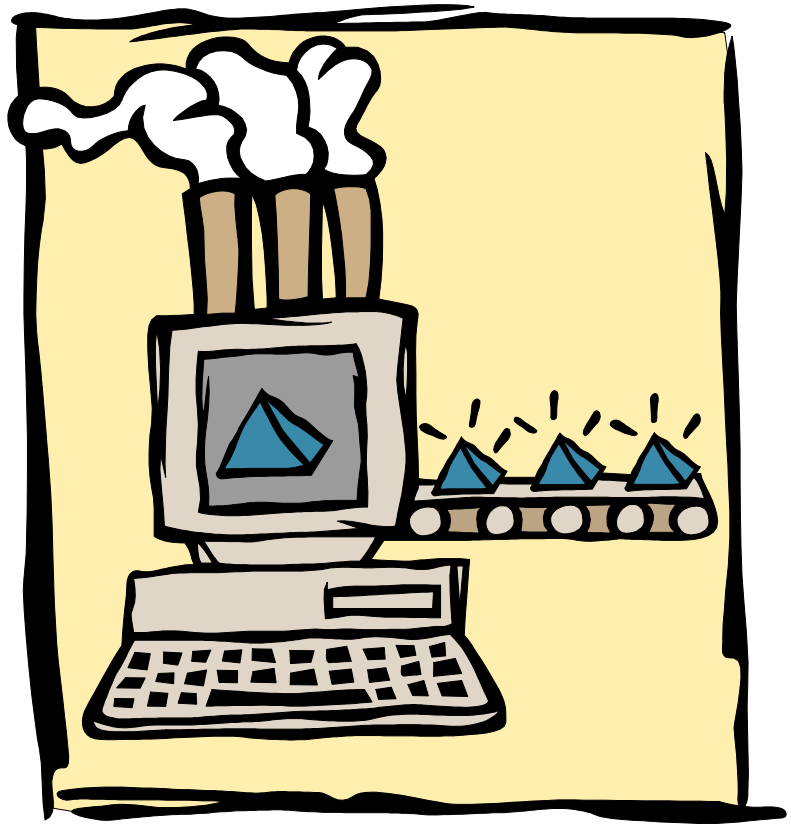
Approved packages are placed on the internal update server for distribution

Downloading and Installing Updates



Automating the Process With autorpm

- **Autorpm is designed to help automate much of this process**
  - **Mirror RPMs from an FTP site**
  - **Keep installed RPMs consistent with an FTP site or local directory**
  - **Keep installed RPMs in a cluster or network of systems consistent**
- **Autorpm can be configured to check and all cryptographic signatures and only install those packages that can be verified**



An Example Autorpm Scenario

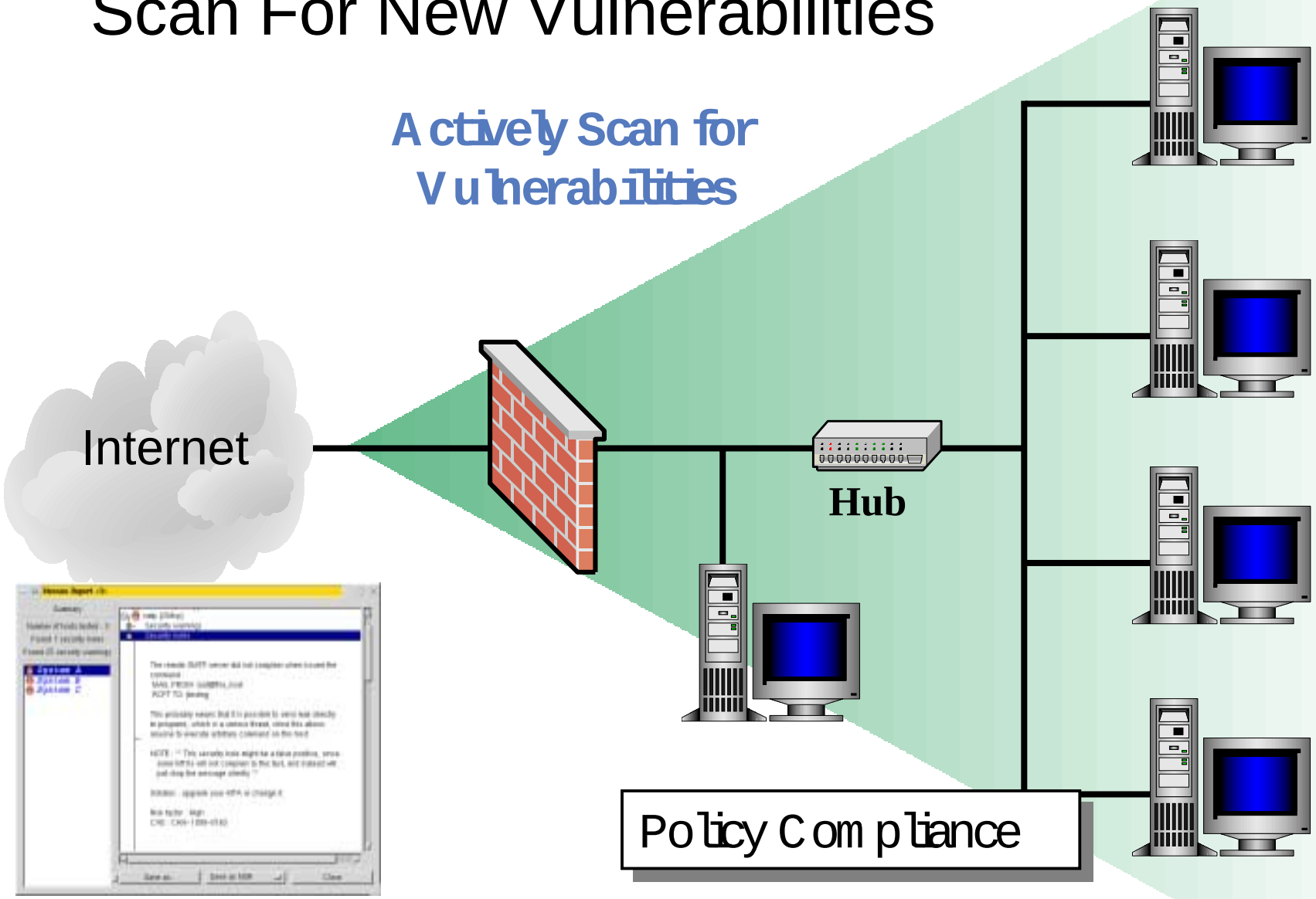
- **The IS department installs autorpm reconfigured to update from the internal update server on all Linux desktop systems**
- **The IS public key plus the Linux distribution key (RedHat, Mandrake, …) are also installed onto the root account**
- **A cron entry is added to run autorpm once a day**
- **Update package that the IS department verifies and places on the internal update server will now be automatically distributed**

Assessment: Finding Vulnerabilities

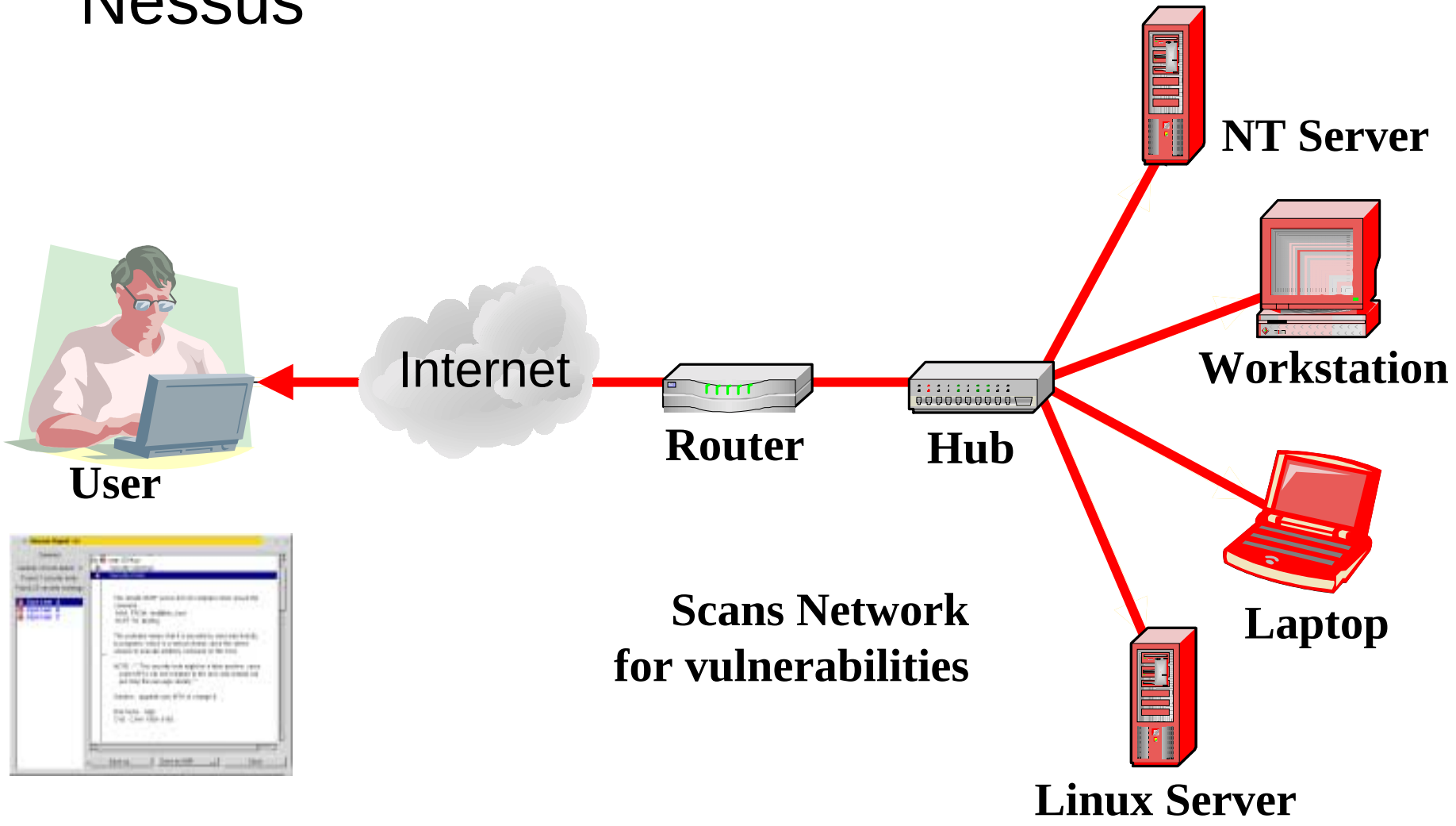
Find Vulnerabilities Before Others

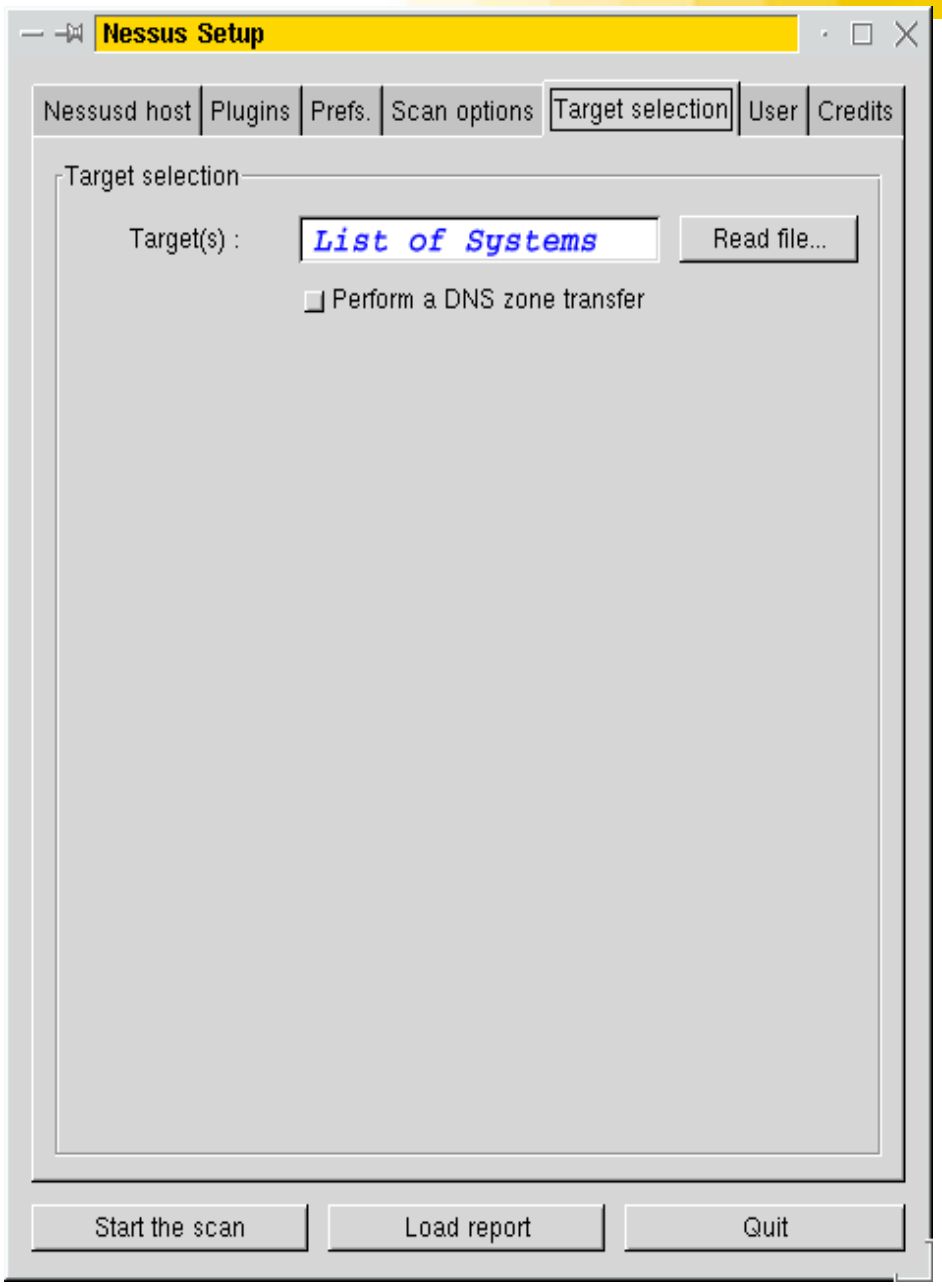
- **Find vulnerabilities before they can be exploited**
- **Correct the problems that you find**
- **Use the tools that the attackers use**
- **Vulnerability scanners combine many of the exploits found in hundreds of attack tools into a easy to use interface**
  - **Detailed reports are created for review**
  - **Most include suggested procedures to remove the vulnerability**
- **Open source tools exist for small business and home users**
- **Commercial products generally provide a better assessment**
  - **Symantec ESM and NetRecon**
  - **…**

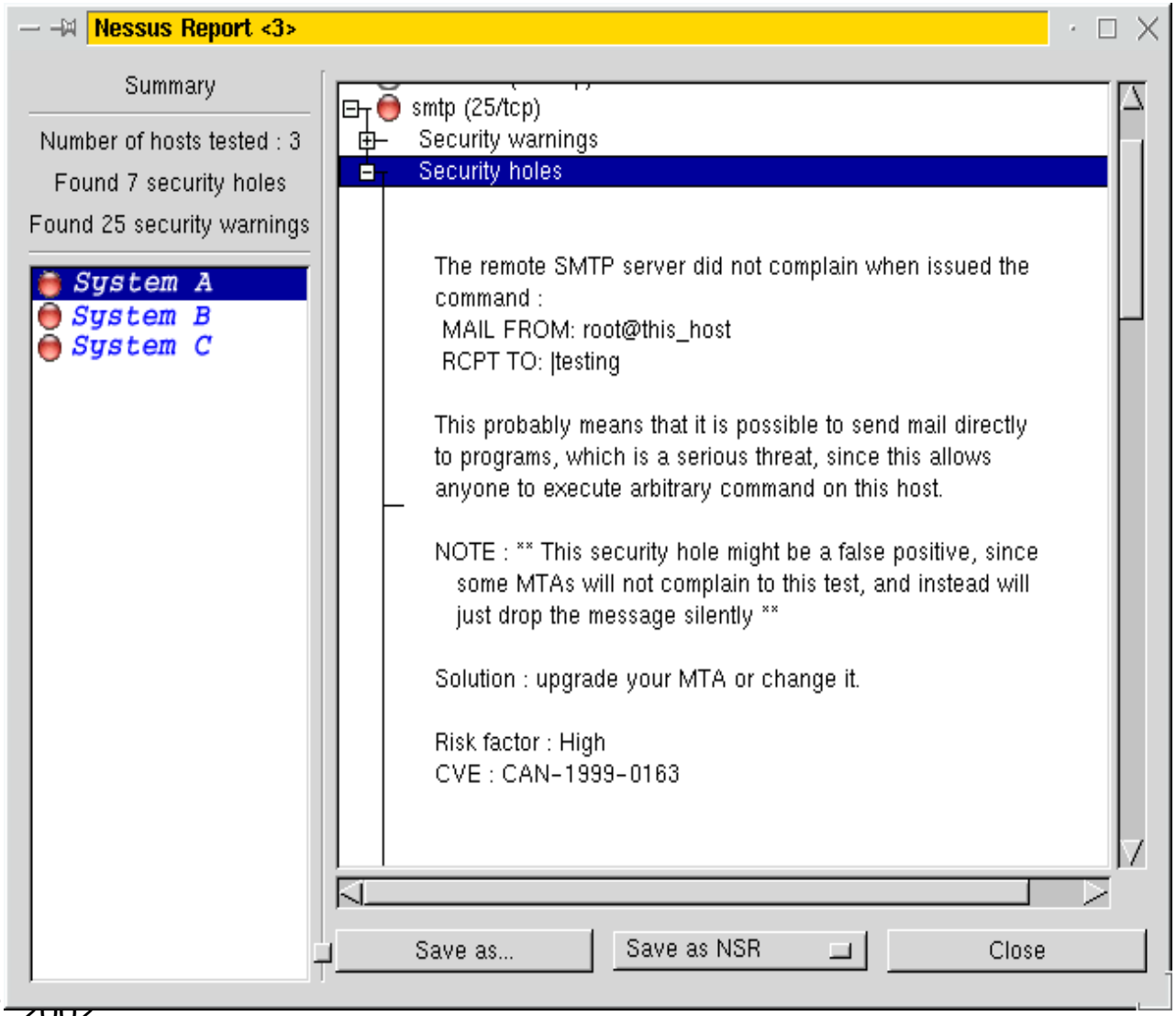
Scan For New Vulnerabilities



Nessus


NT Server
Workstation
Internet
Router
Hub
User
Laptop
Scans Network
for vulnerabilities
Linux Server


Nessus Setup
Nessusd host
Plugins
Prefs.
Scan options
Target selection
User
Credits
Target selection
Target(s) :
List of Systems
Read file...
Perform a DNS zone transfer
Start the scan
Load report
Quit


Nessus Report <3>
Summary
Number of hosts tested : 3
Found 7 security holes
Found 25 security warnings
System A
System B
System C
smtp (25/tcp)
Security warnings
Security holes
The remote SMTP server did not complain when issued the command :
MAIL FROM: root@this_host
RCPT TO: |testing
This probably means that it is possible to send mail directly to programs, which is a serious threat, since this allows anyone to execute arbitrary command on this host.
NOTE : ** This security hole might be a false positive, since some MTAs will not complain to this test, and instead will just drop the message silently **
Solution : upgrade your MTA or change it.
Risk factor : High
CVE : CAN-1999-0163
Save as...
Save as NSR
Close

IV: Where Can I Find More Information?

Where You Can Find More Information

- **Symantec Corporation**
  - http://www.symantec.com
- **Security Focus (Home of BUGTRAQ) and now part of Symantec**
  - http://www.securityfocus.com
- **Packet Storm**
  - http://www.packetstormsecurity.com
- **CVE (Common Vulnerability and Exposures)**
  - http://cve.mitre.org

Where You Can Find More Information

- **SANS Institute**
  - http://www.sans.org
- **The Center for Internet Security**
  - http://www.cisecurity.org
- **Linux Security**
  - http://www.linuxsecurity.com
- **Network Security Library**
  - http://secinf.net
- **Virtual Private network daemon (vpnd)**
  - http://sunsite.dk/vpnd/
- **The Linux Documentation Project**
  - http://linuxdoc.org

V: Conclusion

Conclusion

- **The Linux Operating System (like others) is susceptible to security attacks**
- **Successful attacks can be a serious issue**
  - **Downtime**
  - **Embarrassment**
  - **Lost revenue**
- **You should consider security from the very beginning**
- **You have to understand the technical aspects to combat the threat**
- **Remember that the first step to securing your site should be the development of a security policy that fits your needs**

VI: Questions?