Network Security

- The Internet has grown from small town of a few thousand users into a big city of millions of users.
  - The expanded network provides increased services.
  - The growth reduced the level of trust between network neighbors on the other hand.
  - Similar to the life in a small town like Houghton and a big city like Beijing.
    - Are you going to leave your door, you car unlocked?
    - You need double doors, window fences!

Network Security Advice

- Overreacting to the threat of break-ins may hinder the way you use the network.
  - One of my previous co-work never put his personal information on the web.
  - But Our company does this for him!

- Use common sense. RFC2196 stated this principle:
  Common Sense is the most appropriate tool that can be used to establish your security policy. Elaborate security schemes and mechanisms are impressive, and they do have their place, yet there is little point in investing money and time on an elaborate implementation scheme if simple controls are forgotten.
Security Planning

- Security policy can help you decide
  - what need to be protected
  - How much will you invest
  - Who will be responsible for carrying out the steps to protect it.

- Assessing the Threat
  - RFC2196, site security Handbook, identifies three distinct types of security threats:
    - Unauthorized access
    - Disclosure of information
    - Denial of service (Dos)

Security Planning

- Assessing the Threat
  - Access these threats in relation to
    - Number of users who would be affected
    - Sensitivity of the information that might be compromised
    - The threat of losing the services of a mission-critical system
    - Information security threats:
      - Threat to the secrecy
      - To the availability
      - To the integrity of data
Security Planning

- Distributed Control
  - Use subnets to distribute control
  - Use the network to distribute information
    - Develop a system for disseminating security information to each group.
      - Mailing lists
      - An internal web site
    - Subnet Administrators forward the relevant part to system administrators/individual users.
    - Network administrator should keep update with latest security information
      - Joining appropriate mailing lists and newsgroups
      - Browse web site

A few places to start looking for computer security information
- Your Unix vendor
- The Bugtraq archive
- Security newsgroup
  - Mixture of stuff
- FIRST web site
  - Forum of Incident Response and Security Teams
- NIST Computer Security Alerts
- CERT advisories
  - Computer Emergency Response Team http://www.cert.org
Security Planning

- SANS Institute
  - The System Administration, Networking and Security (SANS) Institute offers informative security newsletters that are delivered weekly via email.
  - http://www.sans.org
- Exploit sites
  - http://www.insecure.org

Writing a Security Policy

- Write a security policy that clearly states what is expected and from who. Security policy should define:
  - The network users' security responsibilities
  - The system administrator’s security responsibilities
  - The proper use of network resources
  - The actions taken when a security problem is detected
    - What should be done by the user, by the administrator?
    - Who should be notified?
    - Even the actions are simple actions, they should be written down so that they are readily available.
- RFC 2196 Site Security Handbook
Implementing Basic Security

- User Authentication
- Application Security
- Security Monitoring
- Access Control
- Encryption
- Firewalls

User Authentication

- Use good password
  - Not to
    - Use account name as a password
    - Use to password or well-published password for guest account
    - Use default passwords
    - Tell passwords to others

- Cracking ways
  - Easy guess
  - Dictionary guessing
  - Brute force guessing
The Shadow password file

- /etc/passwd is world readable
- The encrypted password is stored only in the shadow password file /etc/shadow, read only by root.
- On Solaris, shadow file entry:
  - username
  - Encrypted password, "NP", or "LK"
  - Date that password was last changed
  - Minimum # of days that must elapse before the password can be changed
  - Maximum # of days that the user can keep the password before it must be changed.
  - # of days before the password expired that the user is warned
  - The number of days the account can be inactive before it is locked
  - The date on which the account will be closed

The Shadow Password File

- Password aging
  - Lastchg, max and warn fields all play a role in password aging.
  - Expire field lets you create a user account that has a specified "life".
- The shadow password are changed by passwd utility with options such as -n, -w and -x.
- System wide default value
  - /etc/default/passwd
    [ruihong@cslserver ~]$ more /etc/default/passwd
    #ident "@(#)passwd.dfl 1.3 92/07/14 SMI"
    MAXWEEKS=
    MINWEEKS=
    PASSLENGTH=6
Choosing a Password

- A good password is an essential part of security.
  - Use the first letter of each word from a line in a book, song, or poem.
    - a big fat Pig would have 9 wings
      - abfPwh9w
    - In 1995 we had SNOW in Norfolk
      - I95whSiN
    - he got 12,000 dollars from lottery, NOT!
      - hg12KfiN!
  - Use the password generator
  - Use numbers and letters to create an imaginary password
    - 2hot4U?
  - Password needs to be easy to remember.

One-time Passwords

- If encryption is not available, use one-time password to login remote client to prevent password stealing.
- It is a hassle.
- OPIE
  - One-Time Passwords In Everything (OPIE) is a free software
  - Replace login, su and ftpd with its own version of these programs.
  - Generate list of passwords to use, indexed by sequence number.
Secure r command

- Unix r-command rsh, rcp, rlogin
  - Improperly configured r commands can open access to your computer facilities to virtually everyone in the world.

- R commands use a security system based on trusted hosts and users.
  - Trusted users on trusted hosts are allowed to access the local system without providing a password.
  - The system assumes that user accounts with the same name on both hosts are "owned" by the same user.
  - Trusted hosts and trusted users are defined in /etc/hosts.equiv and .rhosts

/etc/hosts.equiv

- Defines the hosts and users that are granted/denied "trusted" r command access to your system.
- [+|-][hostname] [+|-][username]
  - Example:
    rodent
    -rodent
    rodent -david
    rodent +becky
    + becky

- Do not use "+" as hostname
- Grant trusted access only the systems and users you actually trust, not every system attached to your local network.
Secure r command

- .rhosts
  - Grants or denies password-free r command access to a specific users' account.
  - It is placed in the users' home directory and contains entries that define the trusted hosts and users.
  - Same format as hosts.equiv.
  - Same function as hosts.equiv, only scope is different.
    - hosts.equiv is system wide
    - .rhosts is only user specific.
  - For example
    - Entry "horseshoe anthony" in ~resnick/.rhost and /etc/hosts.equiv
    - Convenient if you have different user name on different systems.

Secure r command

- Systems check /etc/hosts.equiv first and then .rhosts
- For root account, system only check .rhosts - so you don't grant remote root access accidentally from hosts.equiv.
- R command poses a security threat especially for remote access.
  - R command grant trust basted on a belief that the IP address uniquely identifies the correct computer.
  - What if somebody manipulate IP address?
  - What if the same user was created on remote host?
Secure Shell

- Secure shell replaces the standard r commands with secure commands that include
  - Authentication
    - Ensure that the trusted host really is the host it claims to be.
  - Encryption
    - Use public-key encryption schemes to ensure that every packet in the stream of packets is from the source it claims to be from.

- Two versions
  - SSH Secure Shell
  - OpenSSH

Secure Shell

- Basic components of secure shell
  - sshd:
    - deamon handles incoming SSH connections.
    - Started automatically at boot time
    - Sshd generates an encryption key
  - ssh
    - ssh is the secure shell user command, replacing rsh and rlogin.
    - Securely pass a command to a remote system or securely log into a remote system.
  - scp
    - Secure version of rcp
  - sftp
    - A version of FTP that operates over a secure shell connection.
Secure Shell

- `ssh-keygen`
  - Generate the public and private encryption keys used to secure the transmission for the secure shell.

- How does it work when ssh client connects to an sshd server?
  - They exchange public keys
  - Compare to `/etc/ssh_known_hosts` file and `.ssh/known_hosts` in the user's home directory
  - If the key is not found
    - The user is asked to verify that the new key should be accepted
  - If the key has changed
    - A very interesting warning shows up.
      - IT IS POSSIBLE SOMEBODY IS DOING SOMETHING NASTY!

Secure Shell

- If a match is found,
  - If the client is listed in `hosts.equiv`, `shost.equiv`, `.rhosts`, `.rshost`, then it is authenticated
  - Otherwise, the client is prompted for a password.
  - The client uses the key to encrypt a session key, which will be used to encrypt the remainder of the SSH session.

- Create a public-key challenge/response protocol for authentication
  - `ssh-keygen` on the client
    - Create key with a passphrase
  - Copy the public key to your account on the server.
    - From `.ssh/identity.pub` on client to `.ssh/authorized_keys` on server
  - You will be prompted for the passphrase when you log in using ssh to server
Secure Shell

- After SSH has been installed,
  - disable r support from /etc/inetd.conf for incoming connection
    - SSH is started separated
    - R command support is started from inet deamon.
  - Make sure outgoing connections are using ssh.
    - Consider replace rlog, rsh with ssh.
- Check /etc/ssh/ssh_config for configuration value.
  - What is identify file name
  - Weather or not strict host key checking
  - Forward X11
  - ... ...

Exercise

- Set up public-key authentication for ssh to icu0
- Set up putty to forward X11 traffic
**Application Security**

- Most break-ins occur when bugs in applications are exploited or when applications are mis-configured.
  - **Remove unnecessary Software**
    - Any software that allows an incoming connection from a remote site has the potential of being exploited by an intruder.
      - Remove every daemon from /etc/inetd.conf that you don’t absolutely need.
  - **Keep Software Updated**
    - Software update services updates the software via the network, lessening the burden of keeping software up to date.
    - The downside is loss of control from administrator point of view.

**Security Monitoring**

- Good security is an ongoing process.
- Monitor the systems to detect unauthorized user activity and to local and close security holes.
- Know your system
  - What is normal
    - What processes are normally running
    - Who is usually logged in
    - Who commonly logs in after hours
    - What is the normal level of network activity
Security Monitoring

- Use who and ps command
  - who give you
    - Who is logged in
    - What device they are using
    - When they logged in
    - What remote host they logged in from
  - who only see the unix login, not other connections within in applications
    - For example: Database connections

Looking for Trouble
- Looking for intruders' favor hidden files that begin with a dot.
  - `ls -a | grep '^\.'`

Examine certain key files
- `/etc/inetd.conf` and `/etc/xinetd.conf`
- `r` command security files
- `/etc/passwd`
- Files run by cron or at
- Executable files

Use find command to check files
- Example: find all the root setuid program
  - `find / -user root -perm -4000 -print`

Use some software to automate the process of scanning the filesystem.
- [http://www.tripwiresecurity.com](http://www.tripwiresecurity.com)
- [http://www.tripwire.org](http://www.tripwire.org)
Security Monitoring

☀ Checking login activity
  - who
  - last
  - History login info

Logins at odd times or from odd places are suspicious.

Security Monitoring

☐ Get some tools
  ☐ Manually monitoring is time consuming and prone to errors
  ☐ Get monitoring tools
    - Like anti-virus tool, the monitor tools need to be updated with the never ending change.

☐ Exercise: install Nagios
Access Control

- Limiting access.
  - Routers and hosts check the address of a host requesting a service against an access control list.
    - If remote host is permitted, the access is granted.
    - If remote host is not permitted, the access is denied.
  - Access control does not bypass any normal security checks.
  - Access control are common in terminal servers and routers.
  - Access control software is available for Unix hosts.
    - xinetd
    - TCP wrapper

wrapper

- Wrapper performs:
  - Logs requests for internet services
  - Provides an access control mechanism for Unix systems.
- Get tcpd installed if not already there
- Change /etc/inetd.conf
  - Replace the path to each network service daemon that you wish to place under access control with the path to tcpd.
    - For example:
      - finger stream tcp nowait nobody /usr/etc/in.fingerd
      - in.fingerd
      - finger stream tcp nowait nobody /usr/etc/tcpd in.fingerd
Wrapper

- Good candidates for access control are ftpd, tftp, telnetd and fingerd
- Wrapper can not control daemons that are not started by inetd. - but the access list may be used by other daemons.

Tcpd access control files
- Two files: /etc/hosts.allow and /etc/hosts.deny
- If files are not found, tcpd simply logs the request.
- If the files are found, tcpd checks the hosts.allow file first, then hosts.deny. It stops as soon as it finds a match for the host and the service in question.

Wrapper

Format
Service-list : host-list [: shell-command]
  Service-list: the service name
  Host-list: hostnames, domain names, Internet addresses or network numbers
  Shell-command: define additional processing

Examples
- Grants FTP and Telnet access to all hosts in wrotethebook.com domain
  ftpd, telnetd: .wrotethebook.com
- Keyword ALL to represent all services or all hostnames
  ALL: .wrotethebook.com
- Keyword LOCAL matches all local names - which contain no embedded dots.
  ALL: LOCAL, .wrotethebook.com
wrapper

- A little more complicated hosts.allow
  
  Imapd, ipopd3: 172.16.12.
  
  ALL EXCEPT imapd, ipopd3 : ALL

- Keyword EXCEPT can also be used in host-list
  
  ALL: .wrotethebook.com EXCEPT public.wrotethebook.com

- Use at_sign (@) to narrow the definition of services or hosts
  
  - Multi-homed host, provide service to local network only:
    
    in.telnet@172.16.12.2: 172.16.12.0/255.255.255.0

  - User name is required from the client as part of the access control test.
    
    in.rshd: KNOWN@robin.wrotethebook.com

Wrapper

- Defining an optional shell command
  
  - Shell command allows you to define additional processing.
  
  - Most likely the command is used in hosts.deny

  - Example: notify system administrator if there is a potential security attack.
    
    - In /etc/hosts.deny file
      
      ALL: ALL: (safe_finger -l @%h | /usr/sbin/mail -s %d-%h root ) &

    - After the variables were replaced by values:
      
      safe_finger -l @foo.bar.org | /usr/sbin/mail -s in.rshd-foo.bar.org root
Let's take a loot at files on cslserver
- /etc/inetd.conf
- /etc/hosts.allow
- /etc/hosts.deny

Controlling Access with xinetd

- Information in xinetd.conf file parallels values in inetd.conf
- What xinetd adds are capabilities similar to those of wrapper: access control and logging.
- Xinetd provides two logging parameters:
  - log_on_success and log_on_failure
    - USERID
    - HOST
    - PID
    - DURATION
    - EXIT
    - ATTEMPT
    - RECORD
Controlling Access with xinetd

- Xinet provides three parameters for access control.
  - Accept connection from certain hosts
    - `only_from`
      - A numeric address or address scope
      - A network name, defined in `/etc/networks`
      - A domain name
      - An IP address with associated address mask.
  - Reject connection from certain hosts
    - `no_access`
      - Same as `only_from`
  - Accept connections only at certain times of the day.
    - `access_times`
      - Hour:min – hour: min

Example

- `no_access = 172.16.12.250`
- `only_from = 172.16.12.0`
  - The more exact match takes precedence
- A sample POP3 entry from xinetd.conf

```
Service login
{
  socket_type = stream
  wait = no
  user = root
  log_on_success += USERID
  log_on_failure += USERID
  only_from = 172.16.12.0
  no_access = 172.16.12.231
  server = /usr/sbin/ipop3d
}
```
Encryption

- Few words about encryption
- When Is Symmetric Encryption Useful?
- Public_key Encryption tools
  - SSH
  - SSL
  - GPG
  - Stunnel
    - Stunnel is a program that uses SSL to encrypt traffic for daemons that do not encrypt their own traffic.

stunnel

- Stunnel needs a certificate to function properly.
  - Run make in SSL certificate directory
    - cd /usr/share/ssl/certs
    - make stunnel.pem
    - Answer the questions needed to uniquely identify the certificate.
- Once the certificate is created, stunnel is ready for use.
  - Let's look at how to use stunnel to secure POP and IMAP
    - The major benefit is password cannot be stolen from a POP or IMAP session.
    - The download of mail is encrypted too.
**stunnel**

- Use secure connection through SSL
  - Traditional POP’s port is 110
  - Traditional IMAP’s port is 143
  - SSL-secured POP’s port is 995
  - SSL-secured IMAP’s port is 993
  - During startup, run POP inside a SSL tunnel
    ```
    stunnel -d 995 -l /usr/sbin/opop3d - ipop3d
    ```
  - Use stunnel in inetd.conf
    ```
    Pops stream tcp nowait root /usr/sbin/stunnel -l
    /usr/sbin/ipop3d - ipop3d
    ```
  - Use stunnel in xinetd.conf

---

**Homework: How to encrypt NFS connection?**