Chapter 24: System Security

- Introduction
- Policy
- Networks
- Users
- Authentication
- Processes
- Files
- Retrospective

Introduction

- How does administering security affect a system?
- Focus on two systems
  - DMZ web server
  - User system in development subnet
- Assumptions
  - DMZ system: assume any user of trusted administrative host has authenticated to that system correctly and is a “trusted” user
  - Development system: standard UNIX or UNIX-like system which a set of developers can use

Policy

- Web server policy discussed in Chapter 23
  - Focus on consequences
- Development system policy components, effects
- Comparison

DMZ Web Server: Consequences of Policy

1. Incoming web connections come from outer firewall
2. Users log in from trusted administrative host; web pages also downloaded through it
3. Log messages go to DMZ log host only
4. Web server may query DMZ DNS system for IP addresses
5. Other than these, no network services provided
6. Runs CGI scripts
   - One writes enciphered data to spool area
7. Implements services correctly, restricts access as much as possible
8. Public keys reside on web server

Constraints on DMZ Web Server

WC1 No unrequested network connections except HTTP, HTTPS from outer firewall and SSH from trusted administrative host

- Replies to DNS queries from DMZ DNS okay

WC2 User access only to those with user access to trusted administrative host

- Number of these users as small as possible
- All actions attributed to individual account, not group or group account

Constraints on DMZ Web Server

WC3 Configured to provide minimal access to system

- Transfer of enciphered file to spool area should *not* be under web server control

WC4 Software is high assurance

- Needs extensive logging

WC5 Contains as few programs, as little software, configuration information, and other data as possible

- Minimizes effects of successful attack

Development System

- Development network (*devnet*) background
  - Firewall separating it from other subnets
  - DNS server
  - Logging server for all logs
  - File servers
  - User database information servers
  - Isolated system used to build "base system configuration" for deployment to user systems
  - User systems
- What follows applies *only* to user systems

Devnet User System: Policy Components

1. Only authorized users can use devnet systems; can work on any workstation
2. Sysadmins must be able to access workstations at any time
3. Authorized users trusted not to attack systems
4. All network communications except email confidential, integrity checked
5. Base standard configuration cannot be changed
6. Backups allow any system to be restored
7. Periodic, ongoing audits of devnet systems

Consequences for Infrastructure

- Firewall at boundary enforces network security policy
  - Changes to network policy made only at firewall
  - Devnet systems need not be as tightly secured
- No direct access between Internet, devnet systems
  - Developers who need to do so have separate workstations connected to commercial ISP
  - These are physically disconnected from devnet and cannot be easily reconnected

Consequences for User Systems

DC1 Communications authenticated, enciphered, integrity checked

- Consistent naming scheme across systems

DC2 Each workstation has privileged accounts for administrators

- Multiple administrative accounts to limit access to particular privileged functions

DC3 Notion of “audit” or “login” identity associated with each action

- So actions can be tied to individuals

Consequences for User Systems

DC4 Need approval to install program, and must install it in special area

- – Separates it from base system software

DC5 Each workstation protects base system software from being altered

- – Best way: keep it on read-only media

DC6 Employee's files be available continuously

- – Even if workstation goes down
- – Same permissions wherever employee accesses them

Consequences for User Systems

DC7 Workstations store only transient files, so need not be backed up

- Permanent files stores on file server, mounted remotely
- Software, kernel on read-only media

DC8 Logging system to hold logs needed

- Security officers need access to systems, network

Procedural Mechanisms

- Some restrictions cannot be enforced by technology
  - Moving files between ISP workstation, devnet workstation using a floppy
  - No technological way to prevent this except by removing floppy drive
    - Infeasible due to nature of ISP workstations
  - Drib has made procedures, consequences for violating procedures, very clear

Comparison

- Spring from different roles
  - DMZ web server not a general-use computer
  - Devnet workstation is
- DMZ web server policy: focus on web server
  - System provides that service (and supporting services) only; only administrative users have access as users
- Devnet workstation policy: focus on more complex environment
  - Software creation, testing, maintenance
  - Many different users

Networks

- Both systems need appropriate network protections
  - Firewalls provide much of this, but separation of privilege says the systems should too
- How do administrators configure these?

DMZ Web Server

- Accepts web requests only from inner firewall
  - May allow internal users to access web site for testing purposes in near future
- Configuration file for web server software:

```
order allow, deny              evaluate allow, then deny lines
allow from outer_firewall      anything outer firewall sends is okay
allow from inner_firewall      anything inner firewall sends is okay
deny from all                  don’t accept anything else
```

- Note inner firewall prevents internal hosts from accessing DMZ web server (for now)
  - If changed, web server configuration will stay same

DMZ Web Server: Web Server

- Accepts SSH connections only from trusted administrative host
- Configuration file for web software:

```
order allow, deny            evaluate allow, then deny lines
allow from outer_firewall    anything outer firewall sends is okay
allow from inner_firewall    anything inner firewall sends is okay
deny from all                don't accept anything else
```

- Note inner firewall prevents internal hosts from accessing DMZ web server (for now)
  - If changed, web server configuration will stay same

DMZ Web Server: SSH Server

- Accepts SSH connections only from authorized users coming in from trusted administrative server
  - SSH provides per host *and* per user authentication
  - Public keys pre-loaded on web server
- Configuration file for *ssh* server:

```
allow trusted_admin_server     connections from admin server okay
deny all                       refuse all others
```

- Note inner firewall prevents other internal hosts from accessing SSH server on this system
  - Not expected to change

Availability

- Need to restart servers if they crash
  - Automated, to make restart quick
- Script

```
#! /bin/sh
echo $$ > /var/servers/webdwrapper.pid
while true
do
  /usr/local/bin/webd
  sleep 30
done
```

- If server terminates, 30 sec later it restarts

DMZ Web Server: Clients

- DNS client to get IP addresses, host names from DMZ DNS
  - Client ignores extraneous data
  - If different responses to query, discard both
- Logging client to send log messages to DMZ log server
  - Log any attempted connections to any port

Devnet Workstation

- Servers:
  - – Mail (SMTP) server
    - Very simple. just forwards mail to central devnet mail server
  - – SSH server
  - – Line printer spooler
  - – Logging server
- All use access control wrappers
  - – Used to restrict connections from within devnet as well as duplicate firewall restrictions

Access Control Wrappers

- TCP wrappers configured to intercept requests to active ports on workstations
  - Determines origin (IP address) of request
  - If okay, allows connection transparently
  - Log request
- Access controlled by configuration file
  - Second program examines network requests from variety of ports
  - If illicit activity indicated, adds commands to configuration file to block access requests from that origin

FTP, Web Services in Devnet

- Special server systems
  - Neither is on *any* devnet workstation
  - To make files, pages available place them in special areas on file server
    - FTP, Web servers remotely mount these areas and make them available to the server daemons
- Benefits
  - Minimizes number of services that devnet workstations have to run
  - Minimizes number of systems that provide these services

Checking Security

- Security officers scan network ports on systems
  - Compare to expected list of authorized systems and open ports
    - Discrepencies lead to questions
- Security officers attack devnet systems
  - Goal: see how well they withstand attacks
  - Results used to change software, procedures to improve security

Comparison

- Location
  - DMZ web server: all systems assumed hostile, so server replicates firewall restrictions
  - Devnet workstation: internal systems trusted, so workstation relies on firewall to block attacks from non-devnet systems
- Use
  - DMZ web server: serve web pages, accept commercial transactions
  - Devnet workstation: many tasks to provide pleasant development environment for developers

Users

- What accounts are needed to run systems?
  - User accounts (“users”)
  - Administrative accounts (“sysadmins”)
- How should these be configured and maintained?

DMZ Web Server

- At most 2 users and 1 sysadmin
  - First user reads (serves) web pages, writes to web transaction areas
  - Second user moves files from web transaction area to commerce transaction spooling area
  - Sysadmin manages system

User Accounts

- Web server account: *webbie*
- Commerce server account: *ecommie*
- CGI script (as webbie) creates file with ACL, in directory with same ACL:
  - (*ecommie*, { *read*, *write* })
- Commerce server copies file into spooling area (enciphering it appropriately), then deletes original file
  - Note: *webbie* can no longer read, write, delete file

Sysadmin Accounts

- One user account per system administrator
  - Ties actions to individual
- Never log into sysadmin account remotely
  - Must log into user account, then access sysadmin account
    - Supports tying events to individual users
    - If audit UID not supported, may be more difficult …
- This is allowed from console
  - Useful if major problems
  - Three people in room with console at all times

Devnet Workstation

- One user account per developer
- Administrative accounts as needed
- Groups correspond to projects
- All identities consistent across all devnet workstations
  - Example: trusted host protocols, in which a user authenticated to host A can log into host B without re-authenticating

Naming Problems

- Host *stokes* trusts host *navier*
  - User Abraham has account *abby* on *navier*
  - Different user Abigail has account *abby* on *stokes*
  - Now Abraham can log into Abigail’s account without authentication!
- File server: hosts *navier, stokes* both use it
  - User *abby* has UID 8924 on *navier*
  - User *siobhan* has UID 8924 on *stokes*
  - File server determines access based on UID
  - Now *abby* can read *siobhan*’s files, and vice versa

UINFO System

- Central repository defining users, accounts
  - Uses NIS protocol
  - All systems on devnet, except firewall, use it
    - No user accounts on workstations
  - Sysadmin accounts present on UINFO system
    - Also on each devnet workstation to allow sysadmins to fix problems with workstation accessing UINFO system (and for local restores)
- Enables developers can log in to any devnet workstation

About NIS

- NIS uses cleartext messages to send info
  - Violates requirement as no integrity checking
- Not a problem in this context
  - Nonadministrative info: sent enciphered, integrity-checked
  - Administrative (NIS) info: vulnerable to fake answers
    - Idea is that a rogue system sends bogus reply before UINFO can
  - Not possible from inside system as are secured
  - Not possible from outside as firewall will block message

Comparison

- Differences lie in use of systems
  - DMZ web server: in area accessible to untrusted users
    - Limiting number of users limits damage successful attacker can do
    - User info on system, so don’t need to worry about network attacks on that info
    - Few points of access
  - Devnet workstation: in area accessible to only trusted users
    - General user access system
    - Shares user base with other systems
    - Many points of access

Authentication

- Focus here is on techniques used
- All systems require some form

DMZ Web Server

- SSH: cryptographic authentication for hosts
  - Does not use IP addresses
  - Reject connection if authentication fails
- SSH: crypto for user; password on failure
  - Experimenting with smart card systems, so uses PAM
- Passwords: use MD-5 hash to protect passwords
  - Can be as long as desired
  - Proactive password checking to ensure they are hard to guess
  - No password aging

Devnet Workstation

- Requires authentication as unauthorized people have access to physically secure area
  - Janitors, managers, etc.
- Passwords: proactively checked
  - Use DES-based hash for NIS compatibility
    - Max password length: 8 chars
  - Aging in effect; time bounds (min 3d, max 90d)
- SSH: like DMZ web server, *except*:
  - *root* access blocked
  - Must log in as ordinary user, then change to *root*

Comparison

- Both use strong authentication
  - All certificates installed by trusted sysadmins
- Both allow reusable passwords
  - One uses MD-5, other DES-based hash
  - One does not age passwords, other does

Processes

- What each system must run
  - Goal is to minimize the number of these

DMZ Web Server

- Necessary processes:
  - Web server
    - Enough privileges to read pages, execute CGI scripts
  - Commerce server
    - Enough privileges to copy files from web server’s area to spool area; not enough to alter web pages
  - SSH server (privileged)
  - Login server (privileged)
    - If a physical terminal or console
  - Any essential OS services (privileged)
    - Page daemon, etc.

Potential Problem

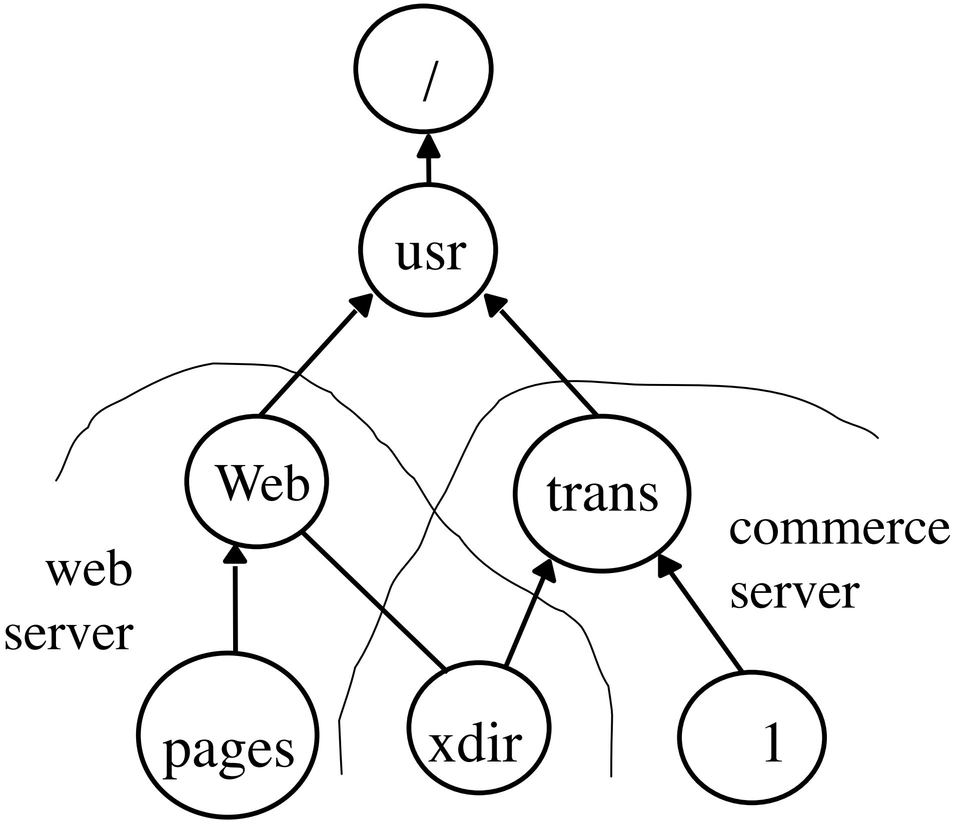
- UNIX systems: need privileges to bind to ports under 1024
  - Including port 80 (for web servers)
  - But web server is unprivileged!
- Solution 1: Server starts privileged, opens port, drops privileges
- Solution 2: Write wrapper to open port, drop privilege, invoke web server
  - The wrapper passes open port to web server

File Access

- Augment ACLs with something like capabilities
- Change process notion of “root directory” to limit access to files in file system
- Example: web server needs to access page
  - Without change: “/usr/Web/pages/index.html”
  - After change: “/pages/index.html”
    - Cannot refer to “/usr/trans” as cannot name it

Example

- Web server changes root directory to /usr/Web
- Commerce server changes root directory to /usr/trans
- Note “xdir” accessible to both processes



Interprocess Communications

- Web server needs to tell commerce server a file is ready
- Use shared directory
  - Web server places file with name "trns*nnnn*" in directory (*n* is digit)
  - Commerce server periodically checks directory for files of that name, operates on them
  - Alternative: web server signals commerce server to get file using signal mechanism

Devnet Workstation

- Servers provide administrative info
  - Run with as few privileges as possible
    - Best: user *nobody* and group *nogroup*
  - Use master daemon to listen at ports, spawn less privileged servers to service request
  - Servers change notion of root directory
- Clients
  - NIS client to talk to UINFO system
  - File server client to allow file server access

Devnet Workstation

- Logging mechanism
  - Records OS calls, parameters, results
  - Saves it locally, sent to central logging server
    - Intrusion detection done; can augment logging as needed
    - Initially, process start, end, audit and effective UIDs recorded
- Disk space
  - If disk utilization over 95%, program scans local systems and deletes all temp files and editor backup files not in use
    - Meaning have not been accessed in last 3 days

Comparison

- DMZ web server: only necessary processes
  - New software developed, compiled elsewhere
  - Processes run in very restrictive environment
  - Processes write to local log, directly to log server
- Devnet workstation: provides environment for developers
  - More processes for more tasks
  - Process environment less restrictive to allow sharing, etc.
  - Processes write to log server, which does all logging

Files

- Protections differ due to differences in policies
  - Use physical limits whenever possible, as these cannot be corrupted
  - Use access controls otherwise

DMZ Web Server

- System programs, configuration files, etc. are on CD-ROM
  - If attacker succeeds in breaking in, modifying in-core processes, then sysadmins simply reboot to recover
  - Public key for internal commerce server here, too
- Only web pages change
  - Too often to make putting them on CD-ROM
  - Small hard drive holds pages, spool areas, temp directories, sysadmin home directory

Example

- Web server: user *webbie*
  - When running, root directory is root of web page directory, “/mnt/www”
  - CGI programs owned by *root*, located in directory (“/mnt/www/cgi-bin”) mounted from CD-ROM
    - Keys in “/mnt/www/keys”
  - Transaction files in “/mnt/www/pages/trans”
    - Readable, writable by *webbie*, *ecommie*
- Commerce server: user *ecommie*
  - Periodically checks “/mnt/www/pages/trans”
  - Moves files out to “/home/com/transact”

DMZ Web Server

- Everything statically linked
  - No compilers, dynamic loaders, etc.
- Command interpreter for sysadmin
  - Programs to start, stop servers
  - Programs to edit, create, delete, view files
  - Programs to monitor systems
- No other programs
  - None to read mail or news, no batching, no web browsers, etc.

DMZ Web Server

- Checking integrity of DMZ web server
  - Not done
- If question:
  - Stop web server
  - Transfer all remaining transaction files
  - Reboot system from CD-ROM
  - Reformat hard drive
  - Reload contents of user directories, web pages from WWW-clone
  - Restart servers

Devnet Workstation

- Standard configuration for these
  - Provides folks with needed tools, configurations
  - Configuration is on bootable CD-ROM
- CD-ROM created on isolated workstation
  - Changes made to that workstation, then new CD-ROM created and distributed
- Workstations also have hard drive for local writable storage
  - Mounted under CD-ROM
  - Can be wiped if any question of integrity

Devnet Workstation

- Logs on log server examined using intrusion detection systems
  - Security officers validate by analyzing 30 min worth of log entries and comparing result to reports from IDS
- Scans of writable media look for files matching known patterns of intrusions
  - If found, reboot and wipe hard drive
  - Then do full check of file server

Comparison

- Both use physical means to prevent system software from being compromised
  - Attackers can’t alter CD-ROMs
- Reloading systems
  - DMZ web server: save transaction files, regenerate system from WWW-clone
    - Actually, push files over to internal network system
  - Devnet workstation: just reboot, reformat hard drive
    - Files on hard drive are transient or replicated (logs)

Comparison

- Devnet workstation: users trusted not to attack it
  - Any developer can use any devnet workstation
  - Developers may *unintentionally* introduce Trojan horses, etc
    - Hence everything critical on read-only media
- DMZ web server: fewer trusted users
  - Self-contained; no mounting files remotely, none of its files mounted remotely
  - CD-ROM has minimal web server system augmented only by additional programs tailored for Drib’s purpose

Summary: DMZ Web Server

- Runs as few services as possible
- Keeps everything on unalterable media
- Checks source of all connections
  - Web: from outer firewall only
  - SSH: from trusted administrative host only
- Web, commerce servers transfer files via shared directory
  - They do not directly communicate

Summary: Devnet Workstation

- Runs as few programs, servers as possible
  - Many more than DMZ web server, though
- Security prominent but not dominant
  - Must not interfere with ability of developer to do job
  - Security mechanisms hinder attackers, help find attackers, and enable rapid recovery from successful attack
- Access from network allowed
  - Firewall(s) assumed to keep out unwanted users, so security mechanisms are second line of defense

Key Points

- Use security policy to derive security mechanisms
- Apply basic principles, concepts of security
  - Least privilege, separation of privilege (defense in depth), economy of mechanism (as few services as possible)
  - Identify who, what you are trusting