WEB SECURITY

By Dr. V. DEEPIKA

Unit - II

Privacy and Security for Users

- The Web's War on Your Privacy
- Privacy-Protecting Techniques
- Privacy-Protecting Technologies
- Backups and Antitheft

Web Server Security

- Physical Security for Servers
- Host Security for Servers
- Securing Web Applications

Web Server Security

1. Physical Security for Servers

- Physical security includes all protective measures before logical access (typing commands).

Examples:

- Alarm systems alerting police during break-ins.
- Key locks on power supplies to prevent unauthorized shutdown.
- Locked computer rooms with closed-circuit cameras.
- Uninterruptable Power Supplies (UPS) and power conditioners to protect against power grid issues.

Importance:

- Even strong encryption and firewalls fail if physical access is compromised.

Example: janitor stealing unattended laptop/server → total security breach

Planning for the Forgotten Threats

- Physical security is often **undervalued** by organizations.

Case studies:

  - Investment firm secured daytime access but ignored night cleaning staff.
  - Magazine lost $100,000+ in computers due to insider misuse of key cards.

- Catastrophic events (e.g., **September 11, 2001**) show limits of physical security.

Key lesson:

  - Catastrophic risks should not prevent **disaster planning**.
  - Organizations with **off-site mirror facilities** recovered fastest.

Challenges:

  - Physical security varies by site.
  - Cannot be preinstalled, downloaded, or sold as software.

Planning for the Forgotten Threats (cont..)

Goal of discussion:

- Provide **starting points**, not fixed solutions.

The Physical Security Plan

- **First step:** create a **written physical security plan**.
- Should be:
  - Part of written security policy.
  - Reviewed by experts.
  - Approved by senior management.
- **Purpose**:
  - Planning + political/organizational buy-in.

The Physical Security Plan (cont..)

Security Plan Should Include:

- Physical assets being protected.
- Locations of assets.
- Security perimeter and its weaknesses.
- Threats (attacks, accidents, natural disasters).
- Existing defenses and improvements.
- Cost estimates.
- Value of protected information.
- Sensitive document → contains weakest defense points.
- Smaller setups still benefit from basic planning.

The Physical Security Plan (cont..)

Five Key Questions:

- Who has physical access?
- What if access is abused violently?
- What if competitors enter unnoticed?
- What if fire destroys systems?
- How will users react after a disaster?

The Disaster Recovery Plan

Definition: plan to restore systems after theft or damage.

Recommendations:

  - Rapid acquisition of replacement equipment.
  - Regular testing of backup restoration.
- Vendor systems can be borrowed for testing.
- Ensure **secure disk wiping** before returning borrowed systems.

Other Contingencies

- Loss of phone/network service.
- Vendor continuity and support.
- Staff absenteeism.
- Death/incapacitation of key personnel.
- Emphasis on **organizational resilience**.

Protecting Computer Hardware

Computers are:

  - Valuable like jewelry.
  - Frequently accessed like office equipment.
- Greatest loss = **data**, not hardware.

Risks:

  - No backup or stolen backups.
  - Time required to rebuild systems.
  - Legal, financial, and reputational damage.

Power sensitivity:

- Vulnerable to surges from lightning or appliances (vacuum cleaner example).

The Environment

- Fire
- Smoke
- Dust
- Earthquake
- Explosion
- Temperature Extremes
- Bugs (biological)
- Electrical Noise
- Lightning
- Vibration
- Humidity
- Water
- Environmental Monitoring

The Environment (cont..)

Fire

- **Fire damage sources:**
  - Flames, heat, water.
- **Fire suppression:**
  - Gas-charged systems (nitrogen, argon, CO_2).
  - Loud alarms before discharge.
- **Guidelines**:
  - Hand-held extinguishers near exits.
  - Annual fire extinguisher training.
  - Monthly extinguisher checks.
  - Override false alarms.
  - Emergency phone access.

The Environment (cont..)

Fire (cont..)

- **Sprinkler systems:**
  - Computers may survive if power is cut.
  - Dry-pipe systems preferred.
- **Water recovery:**
  - Dry equipment fully.
  - Clean circuit boards if minerals present.
- **Modern guidance:**
  - Water sprinklers may outperform gas systems.

The Environment (cont..)

Smoke

- **Smoke damage:**
  - Abrasive particles cause disk crashes.
  - Toxic smoke from electrical fires (e.g., video monitors).
- **Tobacco smoke:**
  - Harms people and computers.
  - Causes keyboard failure.
- **Guidelines:**
  - No smoking.
  - Smoke detectors above/below floors and ceilings.

The Environment (cont..)

Dust

- **Dust effects**:
  - Abrasive, conductive.
  - Causes shorts and erratic behavior.
- **Guidelines**:
  - Dust-free rooms.
  - Clean air filters.
  - Use HEPA/ULPA vacuums.
  - Keyboard dust covers (avoid overheating/static).

The Environment (cont..)

Earthquake

- Earthquake risk is widespread.
- Historical examples:

  - San Francisco (1906), New Madrid fault.

Guidelines:

- Avoid high surfaces.
- Secure shelves.
- Place computers under strong tables.
- Avoid windows.
- Bolt/tie computers (also deters theft).

The Environment (cont..)

Explosion

- Risks from gas or solvents.
- **Guidelines**:
  - Store solvents safely.
  - Off-site backups.
  - Keep systems away from windows.
  - Use ruggedized systems if needed.

The Environment (cont..)

Temperature Extremes

- Optimal range: 50–90°F (10–32°C).

Risks:

- Overheating damages components.
- Cold causes thermal shock.

Guidelines:

- Temperature alarms.
- Adequate airflow (6–12 inches).
- Allow transported systems to acclimate.

The Environment (cont..)

Bugs (biological)

- Origin of term “bug” (Grace Murray Hopper, Mark I).
- Insects damage:
  - Power supplies.
  - Wiring insulation.
- Prevent insect infestation.

The Environment (cont..)

Electrical Noise

- **Sources**:
  - Motors, fans, transmitters.
- **Electrical surges**:
  - Vacuum cleaner example.
- **Guidelines**:
  - Isolated circuits.
  - UPS and line filters.
  - Static mats.
  - Keep transmitters ≥5 feet away.

The Environment (cont..)

Lightning

- Causes magnetic and power surges.

Guidelines:

  - Unplug during storms.
  - Keep backups away from steel structures.
  - Avoid outdoor copper cabling.
  - Use conduits for outdoor cables.

The Environment (cont..)

Vibration

Effects:

- Loosened boards.
- Disk misalignment.

Guidelines:

- Rubber/foam mats.
- Avoid placing printers on computers.
- Laptops are more vibration-resistant.

The Environment (cont..)

Humidity

- **Benefits**:
  - Reduces static.
- **Risks**:
  - Too dry → static damage.
  - Too humid → condensation.
- Optimal: >20% RH, below dew point.
- Use humidity alarms if needed.

The Environment (cont..)

Water

- **Dangers**:
  - Electrical shorts.
  - Trace melting.
- **Sources**:
  - Flooding, sprinklers, plumbing failures.
- **Guidelines**:
  - Water sensors at multiple heights.
  - Avoid basements.
  - Automatic power cutoffs.

The Environment (cont..)

Environmental Monitoring

- Continuous monitoring of temperature and humidity.
- One recorder per 1,000 sq ft.
- Regular log review.

Preventing Accidents

Food and Drink

- Liquids destroy keyboards and consoles.
- Food oils damage media and screens.
- Rule: **No food or drink near computers.**

Physical Access

Raised floors and dropped ceilings

- Intruders can bypass locked rooms.
- **Guidelines**:

  - Walls must extend above ceilings and below floors

Entrance through air ducts

- Large ducts enable entry.
- **Guidelines**:
  - Small ducts.
  - Welded screens.
  - Motion detectors (paranoid option).

Glass walls

- **Risks**:
  - Easy breakage.
  - Shoulder surfing.
- **Guidelines**:
  - Avoid glass.
  - Use translucent blocks.
  - Useful for guarded areas.

Vandalism

- **Motivations**:
  - Revenge, politics, riots, entertainment.
- Often fast and destructive.

Ventilation holes

- MIT case: Coca-Cola poured into vents.
- Prevention:
  - No food/drink.
  - Guards or CCTV.

Network cables

- Vulnerable to cuts.
- Fiber optics:
  - Harder to repair, attractive targets.
- Protection:
  - Steel conduits.
  - Shielded, pressurized conduits.
- Redundancy alone is insufficient.

Network connectors

- High-voltage attacks possible.
- **Example**:

  - Thin-wire Ethernet plugged into 110VAC outlet

Defending Against Acts of War and Terrorism

- Non-military systems are targets.
- High-risk sectors need extra protection.
- Best defense:
    - Hot backups
    - Mirrored disks
    - Geographically distributed servers

Preventing Theft

Physically secure your computer

- Tie-down devices deter theft.

RAM theft

- Common and hard to detect.
- Figure 14-1:
  - Illustrates RAM modules being removed from a computer.
- Symptoms:
  - Slower performance.
- RAM and CPU chips are high-value items.

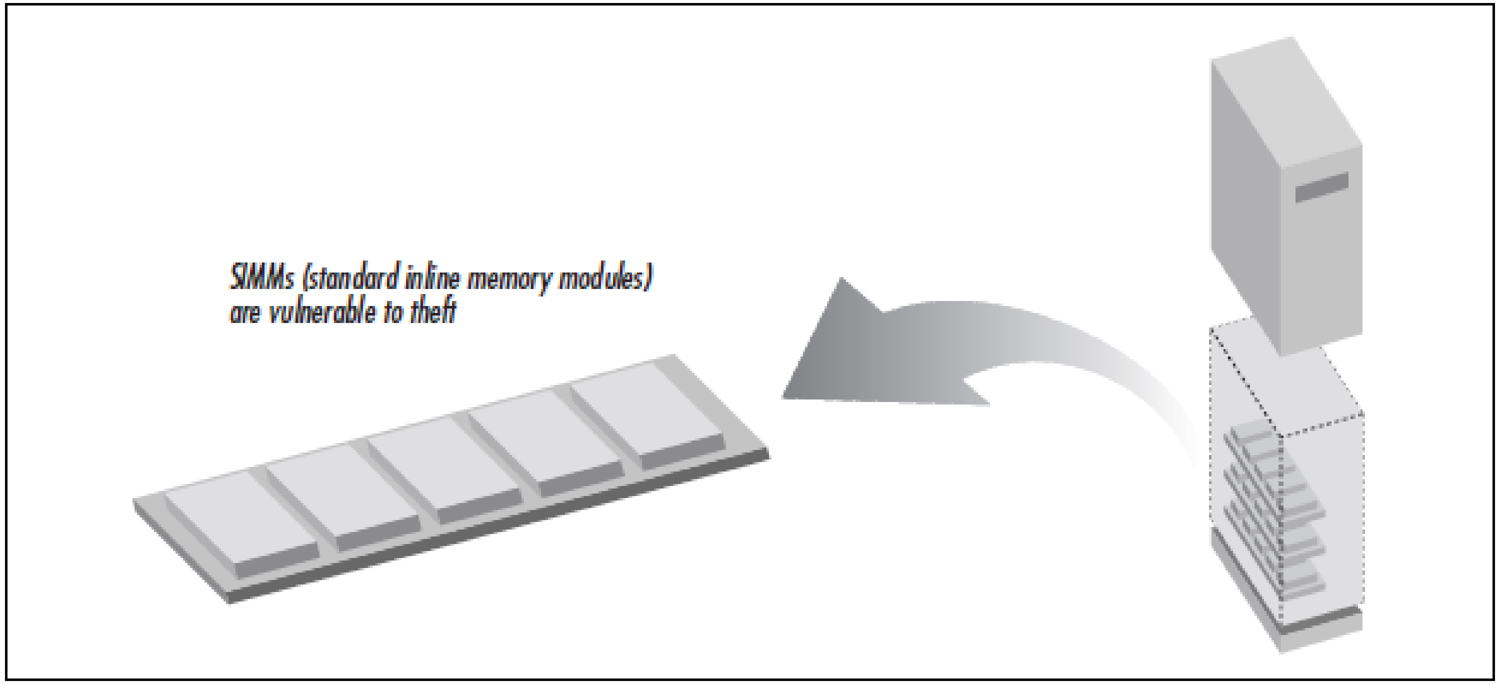


Figure 14-1. There are many recent cases of theft of all or part of computer RAM. RAM is easy to resell and all but untraceable.

Encryption

- Stolen systems expose data.
- Encryption makes stolen data useless.
- Strong encryption recommended for all sensitive data.

Laptops and portable computers

- High theft risk (especially airports).
- Prevention:
    - Engraving ownership details.
    - Property tags.
- **Figure 11-3:**
    - Shows Secure Tracking of Office Property tag.

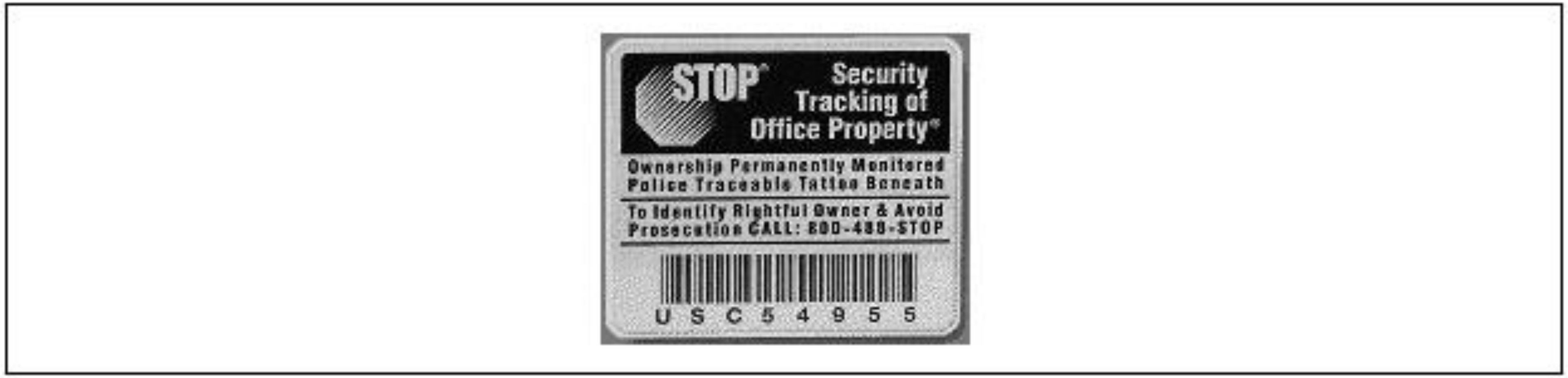

STOP®
Security Tracking of Office Property®
Ownership Permanently Monitored
Police Traceable Tattoo Beneath
To Identify Rightful Owner & Avoid
Prosecution CALL: 800-488-STOP
U S C 5 4 9 5 5

Figure 11-3. The Security Tracking of Office Property (STOP) tag is a simple and effective way to label your laptop (reprinted with permission)

Laptops and portable computers (cont..)

- Encryption tools:
  - Windows 2000 Encrypted File System.
  - PGP Disk.
- Competitive intelligence often targets laptops.

Protecting Your Data

- Strong link between **physical security and data privacy**
- If hardware is stolen, data is compromised
- Many attacks bypass physical security
- Focus: **Data attacks & protection mechanisms**

Data Attacks Overview

- Electronic eavesdropping
- Wiretapping
- Network-based eavesdropping
- Wireless LAN attacks
- Radio/TEMPEST monitoring
- Keyboard monitoring
- Backup theft
- Improper media disposal
- Unattended terminals

Eavesdropping

Electronic eavesdropping is one of the most dangerous forms of data piracy. With relatively simple equipment, an attacker can record:

- Every keystroke
- Information displayed on the screen
- Data sent to printers

The victim is typically unaware of the attack, unknowingly exposing:

- Sensitive information
- Passwords
- Operational procedures

In many cases, detection occurs only after the stolen data is misused, by which time serious damage has already occurred. Although eavesdropping cannot always be detected, **careful security practices** can reduce the risk.

Eavesdropping

Protection Against Eavesdropping

- **Encryption** is the most effective defense.
- Assume communications are being monitored.
- Encrypt all data transmissions by default.

Wiretapping

Wiretapping exploits the fact that electrical wires can easily leak information.

Attackers can:

- Splice directly into cables
- Use induction loops without physical contact
- Monitor telephone lines, modems, and RS-232 communications
- Intercept LAN traffic

Advanced intelligence agencies can even monitor **underwater fiber-optic cables** by analyzing emissions from amplifiers and repeaters.

Wiretapping

Guidelines for Preventing Wiretapping

- Regularly inspect data-carrying wires for damage
- Use shielded or armored cables
- Route cables through steel conduits
- In high-security environments:
  - Pressurize conduits with gas
  - Use pressure sensors to detect tampering

  (These methods are expensive to implement and maintain.)

Eavesdropping over Local Area Networks (Ethernet and twisted pair)

Ethernet and twisted-pair LANs are highly vulnerable. An attacker can intercept traffic by:

- Connecting a packet monitor to an unused network port

Security Measures

- Disable unused Ethernet ports in wiring closets
- Do not leave live network ports in unused offices

Role of Switches

- Ethernet switches limit packet broadcasting
- Improve security over shared Ethernet
- However, skilled attackers can still monitor switched networks
- Switches should **not** be relied upon as the sole security mechanism

Eavesdropping over Local Area Networks (Ethernet and twisted pair)

Network Monitoring

- Periodically scan for unauthorized hosts
- Monitor unknown MAC addresses
- Configure hubs/switches to:
  - Raise alarms
  - Disable ports on MAC/IP mismatch
- Use MAC address filtering and port lock-down

Eavesdropping on 802.11 Wireless LANs

Wireless LANs are inherently insecure.

- WEP encryption is weak
- Attackers can impersonate authorized users
- Wireless traffic is easily intercepted

Protection Measures

- Avoid wireless LANs in high-security environments
- If required:
  - Place access points outside the firewall
  - Use additional encryption (VPN or SSL)

Eavesdropping by Radio and Using TEMPEST

All electronic equipment emits radio frequency (RF) radiation.

- Emissions can be analyzed to reconstruct processed data
- Known as **radio eavesdropping**

TEMPEST

- A certification system measuring susceptibility to RF monitoring
- TEMPEST-certified equipment:
  - Better shielding
  - Larger and more expensive

Alternative Approaches

- TEMPEST-certified rooms or buildings
- Conductive shielding in walls
- Reduction of monitor emissions using special fonts (e.g., **Soft TEMPEST**)

Fiber Optic Cable

Fiber optic cable offers improved protection:

- Harder to tap than copper cable
- Tapping usually requires cutting the cable
- Less interference and grounding issues

Limitations:

- Optical “vampire” taps exist
- Fiber is fragile
- Repairs are difficult

Keyboard Monitors

Keyboard monitors are hardware devices placed between the keyboard and computer.

- Capture every keystroke
- Undetectable by software
- Require physical access to retrieve data
- Typically inexpensive and widely available

Protecting Backups

Backup media is highly vulnerable.

- OS security protections do not apply to tapes
- Anyone with physical access can read backup data

Backup Protection Guidelines

- Never leave backups unattended
- Use bonded messengers
- Sanitize old backup media
- Encrypt backups
- Secure cryptographic keys carefully

Verify Your Backups

Backups degrade over time due to:

- Environmental conditions
- Magnetic print-through

Best Practices

- Test recent and archived backups
- Periodically restore sample backups
- Spin and rewind tapes annually to reduce print-through
- Verify backups at least once per year

Protect Your Backups

- Backups face the same hazards as live systems
- Store backups at a separate physical location
- Geographic separation improves survivability

Sanitizing Media Before Disposal

Deleting files does not erase data.

- Data remnants remain recoverable

Hard Disk Challenges

- Hidden and reserved disk storage
- Requires disk-specific sanitization software
- Risk of firmware-level attacks

Tape and Optical Media

- Use bulk erasers for tapes
- Overwrite multiple times:
  - Zeros
  - Ones
  - Random data

Sanitizing Printed Media

Printed materials often contain sensitive information:

- Source code
- Design documents
- Phone books
- System configurations

Improper disposal enables:

- Social engineering
- Corporate espionage

Sanitizing Printed Media (cont..)

Dumpster Diving

- Attackers recover sensitive data from trash
- Can occur off-site after trash removal

Protection Measures

- Use shredders
- Train users on proper disposal
- Consider on-site incineration where permitted

Protecting Local Storage

Many devices store data unknowingly:

- Printers
- Fax machines
- Modems
- Terminals

These devices often lack:

- Password protection
- Encryption

Printer Buffers and Output

- Printers store documents in memory
- COPY buttons can reproduce sensitive data
- Network printers may contain hard disks
- Unclaimed printouts are vulnerable to theft

X Terminals

- May contain RAM or hard disks
- Often lack encryption

Security Guidelines

- Power off after use
- Password-protect storage
- Erase disks before servicing

Function Keys

- Can store keystroke sequences
- Storing passwords is dangerous
- Physical access compromises credentials

Unattended Terminals

Logged-in unattended systems allow:

- File theft
- Network attacks
- Identity misuse

Countermeasures

- Automatic logout
- Screen locking
- Shell autologout variables
- Secure screensavers

Key Switches

- Prevent booting into single-user mode
- Firmware passwords provide added security
- Physical access remains the primary risk

Personnel

People pose significant security risks.

- Insiders, contractors, and cleaning staff
- Inadequate background checks increase exposure

Controls

- Background investigations
- Bonding
- Security awareness training
- Incident response education

Story: A Failed Site Inspection

A company believed it had “nothing to lose,” yet a brief inspection revealed:

- Fire hazards
- Unprotected networks
- Poor access controls
- Theft opportunities
- Sabotage risks

Downtime costs were estimated at **millions per hour**, proving the organization had far more to lose than expected.

2. Host Security for Servers

Host Security: Definition and Background

- Host security refers to the protection of the computer system on which a web server runs.
- Historically treated as a standalone discipline within computer security.
- Extensive literature exists focusing on operating system and user-level protection.

Historical Context (1980s–Early 1990s)

- Host security was critical in multi-user time-sharing systems.
- **Common environments:**
  - **Universities**: Preventing students from accessing each other’s coursework.
  - **Government systems**: Segregating “Secret” and “Top Secret” information.
- **Traditional concerns:**
  - Protecting the operating system from users
  - Protecting users from each other
  - Implementing auditing and monitoring mechanisms

2. Host Security for Servers (cont..)

Shift in the 1990s

- Rise of personal computers and distributed systems.
- False assumption: exclusive computer use reduced security needs.
- **Reality**:
  - Distributed systems are equally or more vulnerable.
- Reasons for reduced emphasis:
  - Increased complexity and cost of securing distributed environments.
  - Preference for ease of deployment over security

2. Host Security for Servers (cont..)

Renewed Importance Due to the Web

- Web servers expose host systems to external attackers.
- If attackers gain OS-level control:
  - They can access files
  - Monitor communications
  - Modify the web server itself
- **Key principle**: A compromised operating system cannot provide secure services.

Scope of Discussion

- No step-by-step guide provided due to constraints.
- **Focus**:
  - Common host security problems
  - Methods to minimize risks

Current Host Security Problems

- Many issues identified in **RFC 602 (1973)** still exist.
- **Common problems:**
  - Poor server hardening
  - Weak or reused passwords
  - Password sniffing using packet sniffers
- **Motivations for attacks:**
  - Thrill-seeking
  - Financial gain
  - Ideological purposes

Dialup Access Issue

- Unauthorized dialups largely eliminated due to commercialization.
- **New risk:**
  - Easily obtained “**authorized” ISP trial accounts**
- Threat has shifted from **unauthorized users to misuse by authorized users.**

A Taxonomy of Attacks

Unsecured Dialups

- Study by Peter Shipley found:
    - Over 50,000 dialup modems
    - More than 2% allowed unrestricted access
- Affected systems included:
    - Fire departments
    - Bookstore order-entry systems
    - Medical records
- Attack methodology: **systematic dialing (wardialing)**

A Taxonomy of Attacks (cont..)

Remote exploits

- Allow compromise **without logging in.**
- Examples:
  - **Ping of Death** (Windows NT 4.0 crash)
  - **BIND DNS remote root exploit**
- Common technique:
  - **Buffer overflow**
    - Overwrites stack memory
    - Executes attacker-supplied machine code

A Taxonomy of Attacks (cont..)

Malicious programs

- **Back doors**: Hidden access services
- **Trojan horses**: Appear legitimate but perform malicious actions
- **Viruses**:
  - Modify existing programs
  - Carry viral payloads
- **Worms**:
  - Self-replicate over networks
  - Install back doors or drop viruses

A Taxonomy of Attacks (cont..)

Stolen usernames and passwords and social engineering

- Attackers escalate normal user privileges to **superuser/administrator**.
- Use of **stolen credentials** to avoid traceability.
- **Social engineering**:
  - Phone-based deception
  - Pretending to be employees or service representatives
  - Exploits human helpfulness

A Taxonomy of Attacks (cont..)

Phishing

- Automated social engineering via email.
- Targets:
  - Usernames and passwords
  - Credit card details
- Fake URLs redirect victims to attacker-controlled servers

Frequency of Attack

Growth of the Internet

- From **231 ARPANET computers (1981)** to millions today.
- Internet used for:
  - Commerce
  - Government
  - Communication

Increased Attacker Collaboration

- Thousands of organized attacker groups.
- Distribution of:
  - Vulnerability data
  - Exploit code
  - Attack tools (email, IRC, websites)

Frequency of Attack (cont..)

Automation and Scale

- Automated scanning and exploitation tools.
- High-speed connections enable attacks on **millions of systems rapidly**.

Honeynet Project Findings

(Often shown using time-to-compromise graphs)

- Average compromise time:
  - **72 hours** for Red Hat 6.2 (June 2001)
- Windows 98 with file sharing:
  - Scanned hourly
  - Compromised within a day
- Some systems compromised within **15 minutes**.

Understanding Your Adversaries

Script kiddies

- Typically, children or teenagers.
- Use pre-written scripts and tools.
- Dangerous due to:
    - Lack of understanding of consequences
- Case studies:
    - Gibson Research DDoS attack (13-year-old)
    - “Mafiaboy” attacks (age 16)

Understanding Your Adversaries (cont..)

Industrial spies

- Black market for stolen data.
- Activities:
  - Extortion
  - Selling trade secrets
- Illegal in many countries.

Ideologues and national agents

- Hacktivism:
  - Political or ideological motivations
  - Website defacement
- Possible state-sponsored attacks.
- Can affect third-party ISPs.

Understanding Your Adversaries (cont..)

Organized crime

- Targets financial and sensitive data.
- Activities include:
  - Fraud
  - Money laundering
  - Illegal trade coordination
- Global reach via the Internet.

Rogue employees and insurance fraud

- Insider threats:
  - Trojan horses
  - Logic bombs
- Motivations:
  - Revenge
  - Malice
  - Insurance scams

What the Attacker Wants

Compromised systems are used for:

- Launching further attacks
- Distributed denial-of-service (DDOS)
- Running covert servers (e.g., **IRC** rendezvous points (Internet Relay Chat))
- Network surveillance
- Hosting contraband or stolen data

Reasons compromised systems are valuable

- High-speed connectivity
- Obfuscation of attacker identity
- Multi-jurisdiction attack paths

Tools of the Attacker’s Trade

nc (netcat)

- “Swiss Army knife” for TCP/IP.
- Functions:
  - Data transmission
  - Port scanning
  - Server creation

trinoo (trin00)

- Distributed DoS attack server.
- Hidden presence.
- Unix-based versions available.

Tools of the Attacker’s Trade (cont..)

Back Orifice and Netbus

- Windows Trojan horses.
- Capabilities:
  - Keystroke logging
  - File access
  - Remote command execution

bots

- Distributed attack agents.
- Used for:
  - DDOS
  - IRC control (Internet Relay Chat)
- Can remain dormant.

root kits

- Provide superuser access.
- Hide attacker presence.
- Modify system utilities and logs.

Securing the Host Computer

Security Through Policy

- Security cannot rely solely on technical checklists.
- Network services inherently expose systems.
- Focus should be on policy-driven security practices.

Poor Security Practices (Nine Key Issues)

- Lack of security planning
- Cost-driven purchases
- Plaintext password transmission
- Improper use of security tools
- Unpatched software
- Poor threat monitoring
- Inadequate logging
- Weak backups
- Insufficient monitoring

Securing the Host Computer (cont..)

Role of Policy

- Defines allowed and disallowed actions.
- Guides:
  - Users
  - Administrators
  - Designers

Standards and Guidelines

Policy should define:

- Access authorization
- Security responsibilities
- Allowed content
- External access rules
- Testing requirements
- Incident response
- Policy updates
- External communication authority

Keeping Abreast of Bugs and Flaws

- Rapid global dissemination of vulnerability information.
- Administrators must:
  - Monitor vendor bulletins
  - Apply patches promptly
- Sources:
  - Vendor mailing lists
  - FIRST teams (e.g., CERT/CC)
  - Security mailing lists (bugtraq, nt-security)

Patch Management

- Verify authenticity (digital signatures, checksums).
- Avoid unofficial patches.
- Beware of malicious or poorly written fixes.

Choosing Your Vendor

- Security often overlooked in purchase decisions.
- Factors affecting security:
  - Vendor code quality
  - User base size
- High-usage platforms attract attackers.
- Risk of:
  - Buggy software
  - Beta/pre-beta deployments

Choosing Your Vendor (cont..)

Evaluation Criteria

- Vendor security reputation
- Patch responsiveness
- Design philosophy
- Feature minimalism
- Historical vulnerability trends

Procurement Requirements

- Proof of secure development practices
- Test documentation
- Vulnerability response policies
- Notification procedures
- Past security advisories

Installation I: Inventory Your System

- Document:
  - Hardware serial numbers
  - RAM, processors, options
- Store inventory securely in multiple locations.
- Software inventory:
  - Vendor
  - Version
  - Activation codes (secured)
- Retain:
  - Packaging
  - Documentation
  - Inserts (often contain critical warnings)

Installation II: Installing the Software and Patches

- Check vendor websites for:
  - Patches
  - Release notes
- Install patches in **correct order**.
- Disconnect system from Internet during installation.
- Installation sequence:
  1. Base OS
  2. OS patches
  3. Applications
  4. Application patches
- Maintain a detailed installation log.

Installation II: Installing the Software and Patches (cont..)

Backup Strategy

- First full backup after installation.
- Second backup after customization.
- Store backups and media securely.
- Restrict physical access.
- Consider removing removable drives.

Minimizing Risk by Minimizing Services

- One of the most effective ways to secure a web server is to **minimize the number of services** running on the host system.
- Each additional network service introduces its own **security risks and attack surfaces.**
- By disabling **nonessential services**, administrators reduce the number of possible entry points for attackers.
- Even services considered “safe” today may later be found vulnerable.
- **Example (BIND vulnerability, 2001):**
  - Berkeley Internet Name Daemon (BIND) flaw allowed remote superuser access.
  - Systems running name servers on web servers were compromised.
  - Systems that had disabled name services were not affected.
- **Key principle:** If you don’t need a service, **disable it.**

Making a Pre-Mac OS X Your Web Server

- Pre–Mac OS X systems (OS 7, 8, 9) offer **inherent security advantages**.

These systems:

  - Lack a command-line interpreter, making remote execution difficult.
  - Do not enable many network services by default.
  - Have historically stable and well-written code from Apple.

- **Available Macintosh web servers:**
  - **MacHTTP** – free, simple administration.
  - **WebStar** – commercial version by StarNine Technologies.
  - **WebStar Pro** – SSL-enabled WebStar.
  - **Apple Personal Web Server** – included with Mac OS 9 and some OS 8 versions.
- **Mac OS X:**
  - Based on FreeBSD (Unix-like OS with deep roots in **Berkeley Software Distribution**).
  - Expected to have Unix-like security characteristics.

Operating Securely

- Security **degrades over time** due to:
  - Installation of new software.
  - Increased system complexity.
  - Disabled security features for convenience.
  - Newly discovered vulnerabilities.
- Security consultants often provide **temporary improvements** without long-term maintenance.
- **Conclusion**: A secure system must be **continuously maintained**, not just initially deployed.

Keep Abreast of New Vulnerabilities

- Vulnerabilities are now disclosed **rapidly and publicly**.
- Exploits often appear **within hours** of disclosure.
- Administrators must respond quickly to apply patches.
- **Firewalls and IP filtering** can limit exposure but:
  - Firewalls themselves may have vulnerabilities.
  - Some attacks exploit allowed protocols.
- **Key takeaway:** Continuous vigilance is essential.

Logging

- Logging records system and network activity.
- Unix and Windows systems allow flexible logging:
  - Single or multiple files.
  - Remote logging to other machines or devices.
- **Importance of logs:**
  - Aid in incident recovery.
  - Reveal attack methods.
  - Provide forensic evidence.
- Logs should be:
  - Enabled on all servers.
  - Reviewed regularly.
- **Commonly logged parameters:**
  - External and internal network utilization.
  - CPU load.
  - Disk usage.
- Logs also help in **capacity planning**.
- Web servers are a notable exception, often maintaining separate logs.

Setting up a log server

- Attackers often **erase or modify logs** after gaining access.
- Solution: Use a **secured log server**.
- A log server:
  - Collects logs from other systems.
  - Offers no services and no user accounts.
  - Is the most secure system on the network.
- Can be placed:
  - Inside the firewall.
  - Outside the firewall.
  - Or both (dual log servers).
- Log servers **supplement**, not replace, local logging.

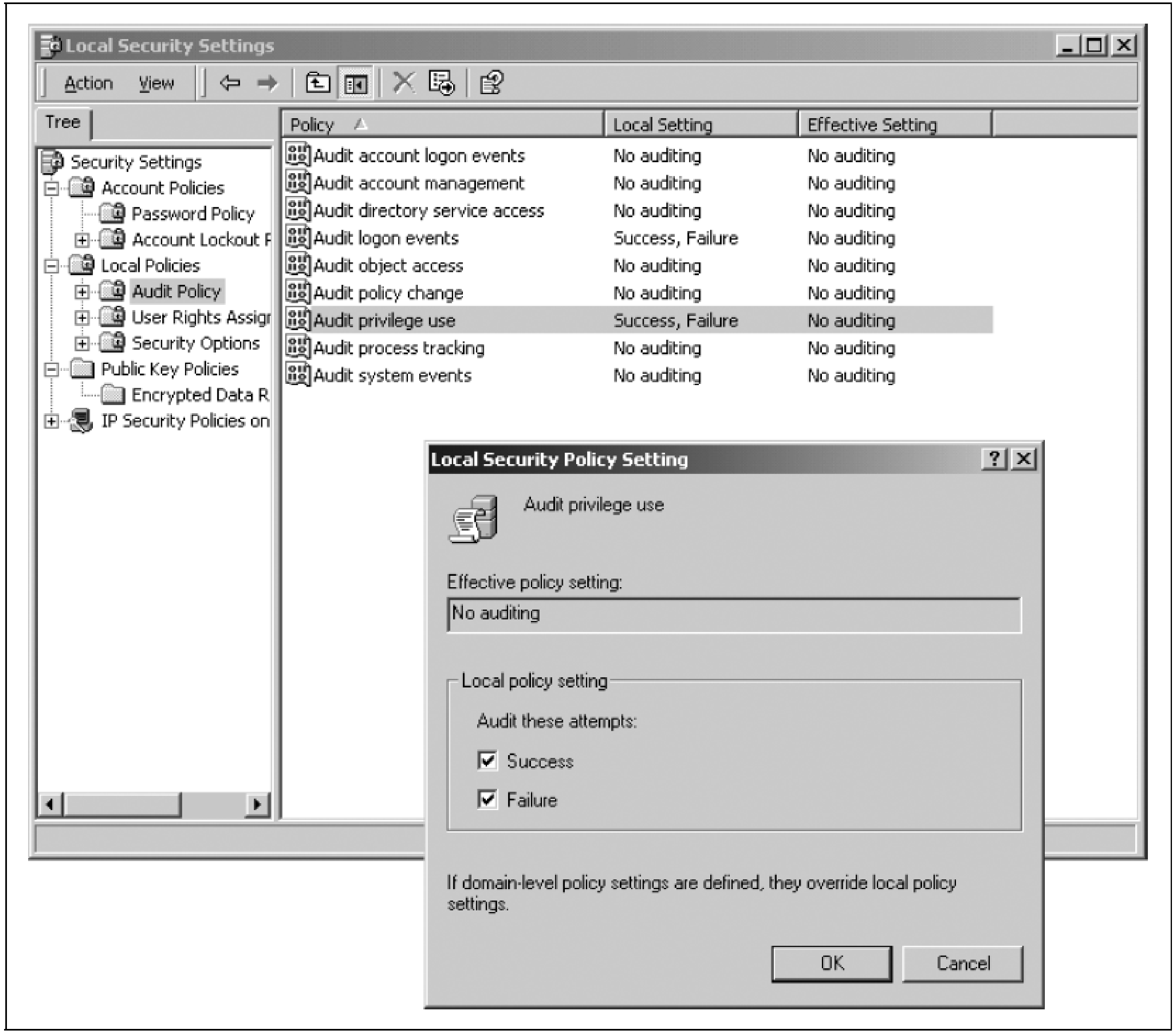
Logging on Unix

- Unix logging uses:
  - **Facilities** (source of message: kern, auth, news, etc.).
  - **Priorities** (severity: info, alert, crit).
- Configuration file: **/etc/syslog.conf**
  - Defines where log messages are sent.
- Log maintenance:
  - Logs must be **rotated and pruned**.
  - Tool: **newsyslog**
  - Configuration file: **/etc/newsyslog.conf**

Logging on Windows 2000

- Controlled by the **Windows logging service**.
- Auditing is disabled by default on some versions.
- Auditing should be enabled to monitor:
  - Login attempts.
  - IP services.
- Excessive logging can generate large volumes of data.
- Logs are pruned automatically.
- **Enabling auditing:**
  - Use Local Security Policy → Local Policies → Audit Policy.
  - Refer to Figure 15-1 showing the Audit Policy interface.

Logging on Windows 2000



Policy	Local Setting	Effective Setting
Audit account logon events	No auditing	No auditing
Audit account management	No auditing	No auditing
Audit directory service access	No auditing	No auditing
Audit logon events	Success, Failure	No auditing
Audit object access	No auditing	No auditing
Audit policy change	No auditing	No auditing
Audit privilege use	Success, Failure	No auditing
Audit process tracking	No auditing	No auditing
Audit system events	No auditing	No auditing

Local Security Policy Setting

Audit privilege use

Effective policy setting:
No auditing

Local policy setting

Audit these attempts:

☑ Success

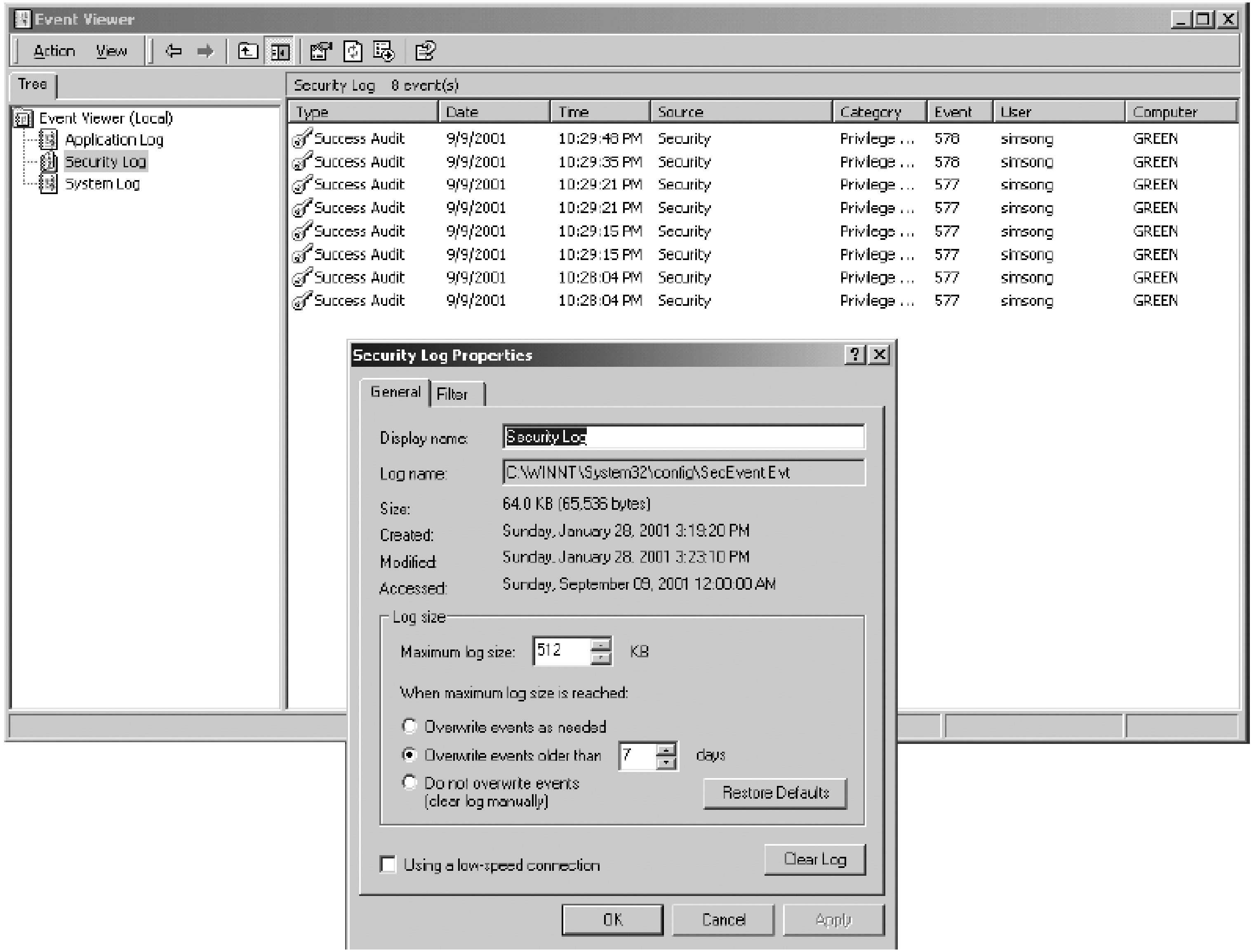
☑ Failure

If domain-level policy settings are defined, they override local policy settings.

Figure 15-1. Enable auditing from the Local Secure Policy Setting application.

Logging on Windows 2000

- **Viewing logs:**
  - Use Event Viewer.
- Retention time can be adjusted (see Figure 15-2).



Security Log 8 event(s)

Type	Date	Time	Source	Category	Event	User	Computer
Success Audit	9/9/2001	10:29:48 PM	Security	Privilege ...	578	simsong	GREEN
Success Audit	9/9/2001	10:29:35 PM	Security	Privilege ...	578	simsong	GREEN
Success Audit	9/9/2001	10:29:21 PM	Security	Privilege ...	577	simsong	GREEN
Success Audit	9/9/2001	10:29:21 PM	Security	Privilege ...	577	simsong	GREEN
Success Audit	9/9/2001	10:29:15 PM	Security	Privilege ...	577	simsong	GREEN
Success Audit	9/9/2001	10:29:15 PM	Security	Privilege ...	577	simsong	GREEN
Success Audit	9/9/2001	10:28:04 PM	Security	Privilege ...	577	simsong	GREEN
Success Audit	9/9/2001	10:28:04 PM	Security	Privilege ...	577	simsong	GREEN

Security Log Properties

Display name: Security Log

Log name: C:\WINNT\System32\config\SecEvent.Evt

Size: 64.0 KB (65,536 bytes)

Created: Sunday, January 28, 2001 3:19:20 PM

Modified: Sunday, January 28, 2001 3:23:10 PM

Accessed: Sunday, September 09, 2001 12:00:00 AM

Log size

Maximum log size: 512 KB

When maximum log size is reached:

○ Overwrite events as needed

● Overwrite events older than 7 days

○ Do not overwrite events (clear log manually)

□ Using a low-speed connection

Figure 15-2. Run the Event Viewer application to view the contents of the log.

Backups

- Backups are copies of data stored on long-term media.

- **Security roles of backups:**
  - Recovery from hardware failures.
  - Restoration after accidental deletion.
  - Recovery from break-ins.
  - Damage assessment via file comparison.
- **Backup risks:**
  - Backup integrity must be verified.
  - Backup servers may control client systems.
  - Unencrypted backups can be intercepted.
  - Backup media must be physically secured.
  - ACL misuse in NT environments can expose all files.
- **Best practices:**
  - Regular backups.
  - On-site and off-site storage.
  - Strong protection of backup media.

Using Security Tools

- Security tools help **evaluate and improve security posture**.
- Tools may be free or commercial.
- **Five categories:**
    - Snapshot tools
    - Change-detecting tools
    - Network scanners
    - Intrusion detection systems
    - Network recording and logging tools
- Attackers use similar tools; administrators should too.

Snapshot tools

- Perform static audits of system configuration.
- Example checks:
  - File permissions (e.g., /etc/passwd).
- **Tools**:
  - **COPS** – historical Unix tool.
  - **Tiger** – modern Unix tool (Texas A&M).
  - **Windows tools**: KSA, NAT, ScanNT, L0phtCrack.
- Should be run **weekly or monthly**.
- Output must be stored securely.

Change-detecting tools

- Detect unauthorized system changes after compromise.
- Help identify:
  - Backdoors.
  - Tampering.
- **BSD/OS daily insecurity report:**
  - Compares /etc files using diff.
  - Vulnerable if comparison files are compromised.
- **Tripwire**:
  - Stores cryptographic checksums.
  - Supports Unix and Windows.
  - Can report to central console.
  - Open-source version available.
- One of the most widely used intrusion detection tools historically.

Network scanning programs

- Scan systems for known network vulnerabilities.
- **Tools**:
  - **SATAN** – historical, modular scanner.
  - Commercial scanners (ISS, Axent, Network Associates).
  - Windows analysis tools from SomarSoft.
- Regular scanning helps administrators identify weaknesses before attackers do.

Intrusion detection systems

- IDS act as **burglar alarms** for computer systems.
- Detect signs of intrusion during runtime.
- **Types**:
  - Host-based IDS.
  - Network-based IDS.
- **Examples**:
  - Tripwire
  - Dragon
  - Cisco Secure IDS
  - Realsecure
  - Shadow
- Mostly commercial solutions.

Virus scanners

- Antivirus tools are essential for **Microsoft platforms**.
- Major vendors:
  - Network Associates.
  - Symantec.
- Unix/Linux:
  - Very few viruses.
  - Integrity tools like Tripwire are sufficient.
- Mac OS:
  - Rare virus infections.
  - Mostly macro-based threats.
- Majority of viruses target **Windows environments.**
- Frequent updates are required.

Network recording and logging tools

- Record **all network traffic** for later analysis.
- Useful for forensic investigations.
- Require large storage capacity.
- **Examples**:
  - NFR
  - NetVCR
  - Silent Runner
  - NetIntercept

Secure Remote Access and Content Updating

- Web content is usually created on desktops and uploaded.
- File transfer introduces authentication risks.
- FTP sends credentials in plaintext.

The Risk of Password Sniffing

- Password sniffing captures unencrypted credentials.
- Affects protocols such as:
  - Telnet
  - FTP
  - POP3 / IMAP
  - HTTP

Using Encryption to Protect Against Sniffing

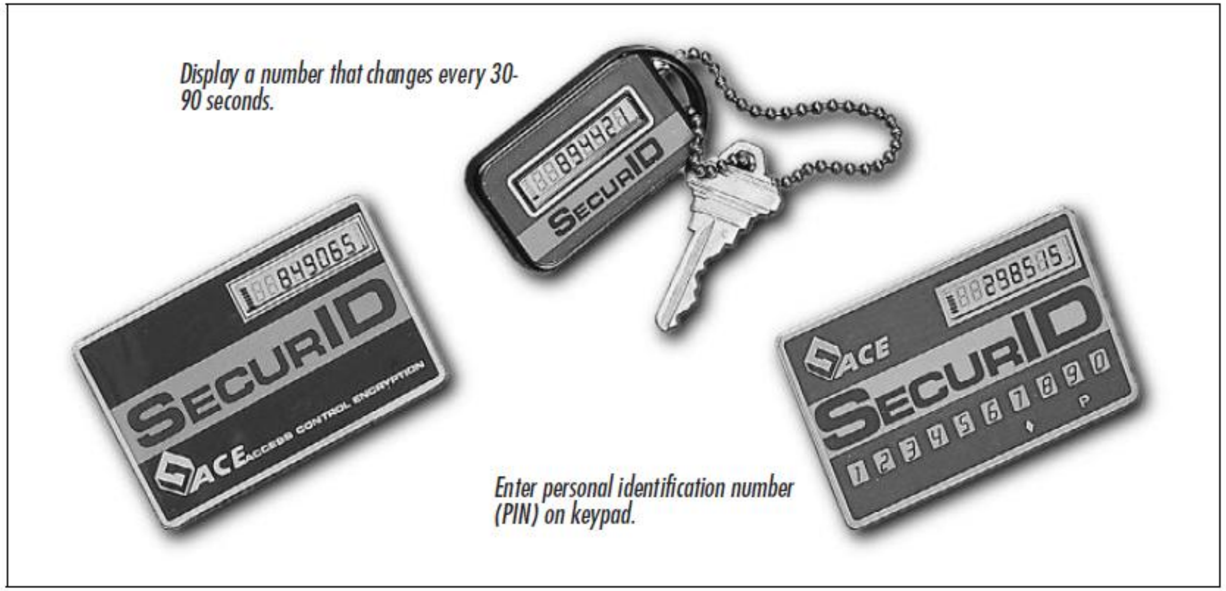

Display a number that changes every 30-90 seconds.
SecurID
ACE Access Control Encryption
Enter personal identification number (PIN) on keypad.

Figure 15-3. Security Dynamics' SecurID card (reprinted with permission)

Use a token-based authentication system

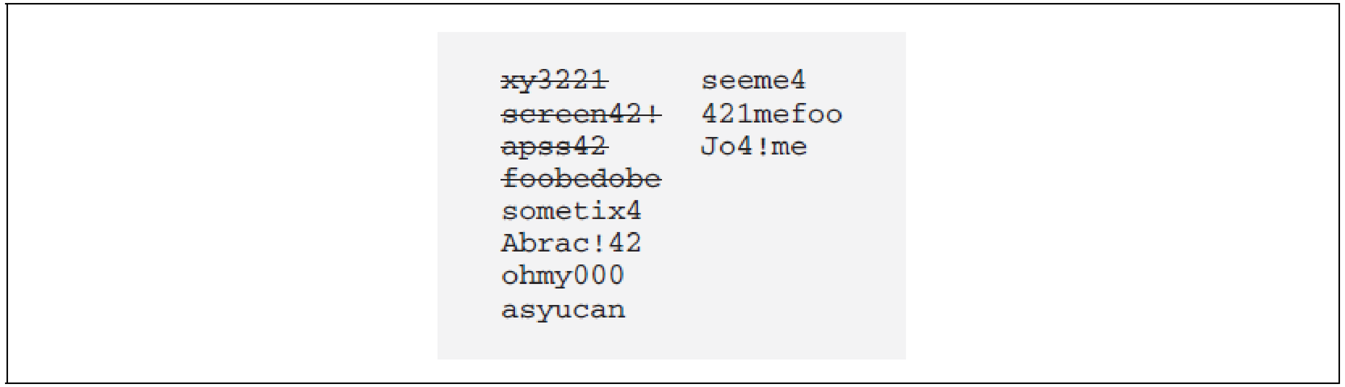
- Example: SecurID (see Figure 15-3).
- Generates one-time passwords.

Use a nonreusable password system

- Example: S/Key (see Figure 15-4).
- Pre-generated password lists.

Use a system that relies on encryption

- **Examples**:
  - Kerberos
  - SSH / SCP
  - SSL / TLS
- Protects against sniffing and session hijacking.



~~xy3221~~
~~screen42!~~
~~apss42~~
~~foobedobe~~
sometix4
Abrac!42
ohmy000
asyucan

seeme4
421mefoo
Jo4!me

Figure 15-4. S/Key uses nonreusable passwords

Secure Content Updating

- Web servers should ideally be behind firewalls.
- VPNs provide the safest remote update method.
- **Update methods include:**
  - scp/ssh
  - FTP
  - rcp/rdist
  - NFS
  - SMB
  - Physical transfer

scp/ssh

- Secure, encrypted file transfer.
- Supports recursive directory copying.
- Uses public key authentication.
- Does not delete obsolete files by default.
- Synchronization tools may be required.

FTP

- Widely supported.
- Weak authentication.
- Vulnerable to sniffing.
- Can be enhanced using S/Key or SSH tunneling.

Unix rcp or rdist

- Can be secured using Kerberos or SSH.
- Supports IP-based authentication.
- Vulnerable to IP spoofing but less risky than plaintext passwords.

NFS

- Allows centralized content management.
- Filesystems should be mounted read-only.
- Performance impact possible.
- Suitable for multiple web servers.

Using SSH and FTP Together

- SSH tunnels FTP control traffic.
- Protects usernames and passwords.
- Data traffic remains unencrypted.
- Reduces overhead.

SMB

- Enables Windows file sharing.
- Requires careful firewall filtering.
- Disable guest accounts.
- Restrict administrative access.

Physical transfer

- No network exposure.
- Requires physical access.
- Suitable for high-security environments.

Dialup Modems

- Modems present hidden back doors.
- Many lack authentication.
- Organizations must:
  - Establish modem policies.
  - Conduct telephone scans.
- **Scanning tools:**
  - PhoneSweep
  - TeleSweep
  - THL-SCAN
  - Toneloc
- Telephone firewalls (e.g., TeleWall) provide strong protection.

Firewalls and the Web

- Firewalls **contain attacks**, not prevent them.
- Used for:
  - Protocol control.
  - Traffic filtering.
- Overreliance can weaken internal security.

Types of Firewalls

Packet filtering

- Router-based filtering.
- Fast and inexpensive.
- Does not inspect payloads.

Proxy

- Breaks direct connections.
- Uses intermediary servers.
- Proxy vulnerabilities possible.

Network Address Translation

- Hides internal IP addresses.
- Enables IP reuse.
- Simplifies ISP changes.

Virtual Private Networks

- Allow secure tunneling.
- Can be exploited if endpoints are compromised.

Protecting LANs with Firewalls

- Firewalls block dangerous traffic like ICMP Echo.
- Internal threats still remain.

Protecting Web Servers with Firewalls

- Limit traffic to required ports (80, 443).
- Isolate web server from internal network.
- Refer to **Figure 15-5** illustrating firewall isolation.
- VPNs can be used for secure content updates.

Protecting LANs with Firewalls

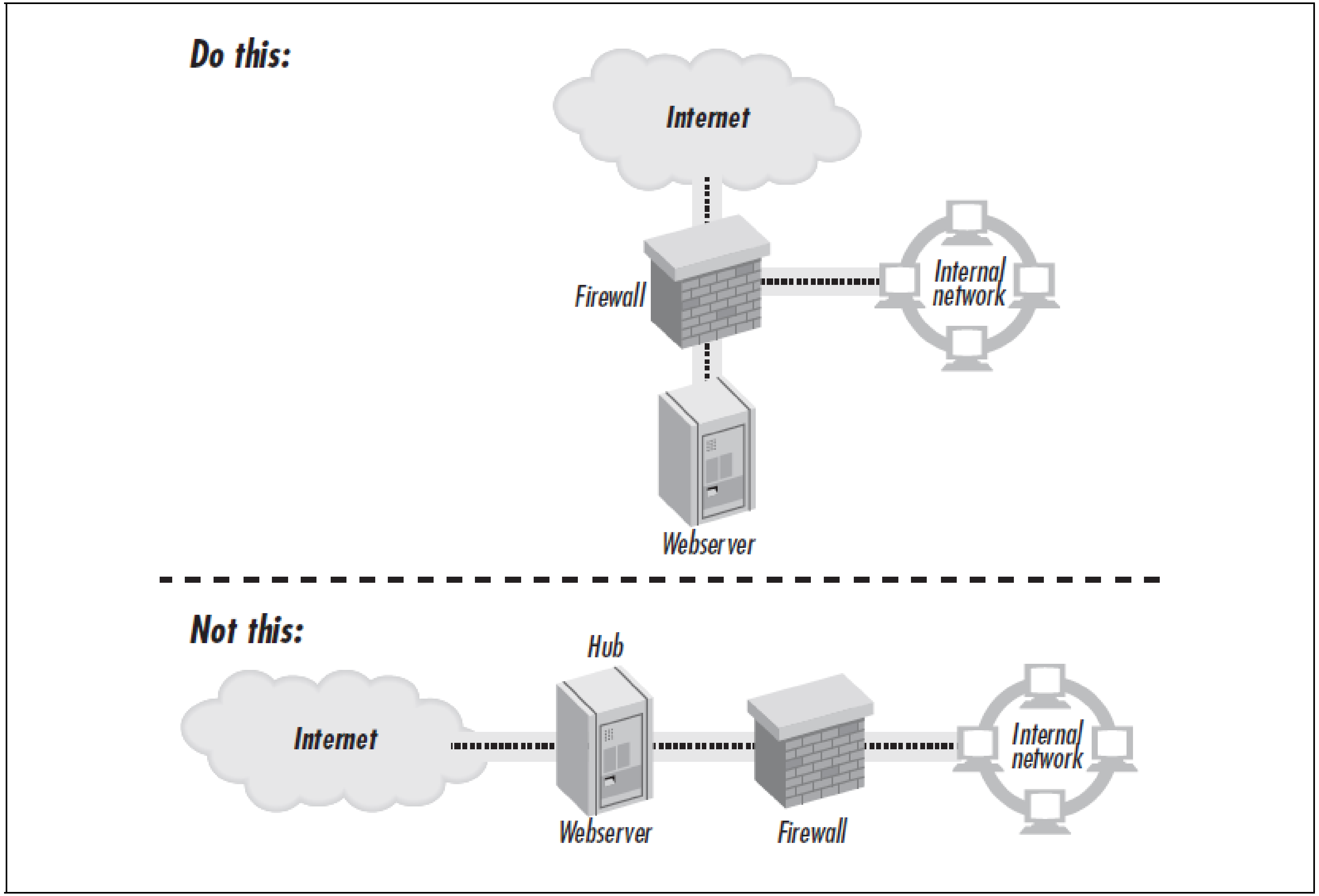


Figure 15-5. For high security, use a firewall to protect your web server from attackers on the Internet. Position the firewall so that it also protects your own organization from the web server.

3. Securing Web Applications

- Web servers are commonly used to display **static content** such as brochures, FAQs, and catalogs.
- **Dynamic web applications** (e.g., shopping carts, personalized pages) require:
  - Customized code
  - Business logic execution
- This code executes **each time a web page is fetched**.
- Code usually runs as:
  - Scripts
  - Programs triggered by specific URLs
- Web servers combined with programming languages allow powerful applications.

3. Securing Web Applications (cont..)

- **Problem**: These programs may contain **hidden flaws.**
- Flaws are often not visible during normal operation.
- Attackers exploit these flaws to compromise:
  - Web servers
  - Underlying operating systems
- This chapter focuses on **secure programming techniques** for web applications.

A Legacy of Extensibility and Risk

- Web servers are highly extensible.
- Extensibility increases functionality, but also security risk.
- Four primary techniques are used to create web-based applications.

CGI

- **Common Gateway Interface (CGI)** was the first web extension mechanism.
- When a CGI URL is requested:
  - Web server launches a **separate process**
  - Captures program output
  - Sends results to the browser

- Parameters are passed via:
  - Environment variables
  - Standard input

- CGI programs can:
  - Perform database queries
  - Run financial calculations
  - Enable chat systems

- Early web innovations (search engines, tracking systems) used CGI.
- **Risk**: Any executable program can be run.

Plug-ins, loadable modules, and Application Programmer Interfaces (APIs)

- Second extension technique.
- Uses modules written in **C or C++.**
- Modules are loaded into the web server's **address space**.
- Advantages:
  - Faster than CGI
  - No new process per request
- Disadvantages:
  - Difficult to write safely
  - A single bug can crash the entire web server
- Bugs affect both:
  - Web server
  - Host operating system

Embedded scripting languages

- Third technique for adding programmability.
- Scripts are embedded directly into web pages.
- An interpreter runs the script **before sending output**.
- Faster than CGI.
- Examples:
  - Microsoft ASP
  - PHP
  - Server-side JavaScript
  - mod_perl
- Widely used for dynamic web applications.

Embedded web server

- Web server functionality is embedded directly into the application.
- No separate web server process is required.
- Common in specialized systems and appliances.

- These extension techniques allow **any program to run**.
- Security consequences include:
  - Running vulnerable programs
  - Allowing outsider access
  - Modifying or deleting critical files

Limiting Damage from Web Applications

- Two methods reduce potential damage:
    1. **Secure program design and inspection**
    2. **Restricted execution environments**
- On multiuser systems:
    - Web servers run as restricted users (e.g., nobody, httpd)
    - CGI and API programs inherit these privileges
- Some operating systems lack privilege separation:
    - Windows 3.1
    - Windows 95/98/ME
    - Mac OS 7–9
- These systems cannot restrict CGI program access effectively.

Programs That Should Not Be CGIs

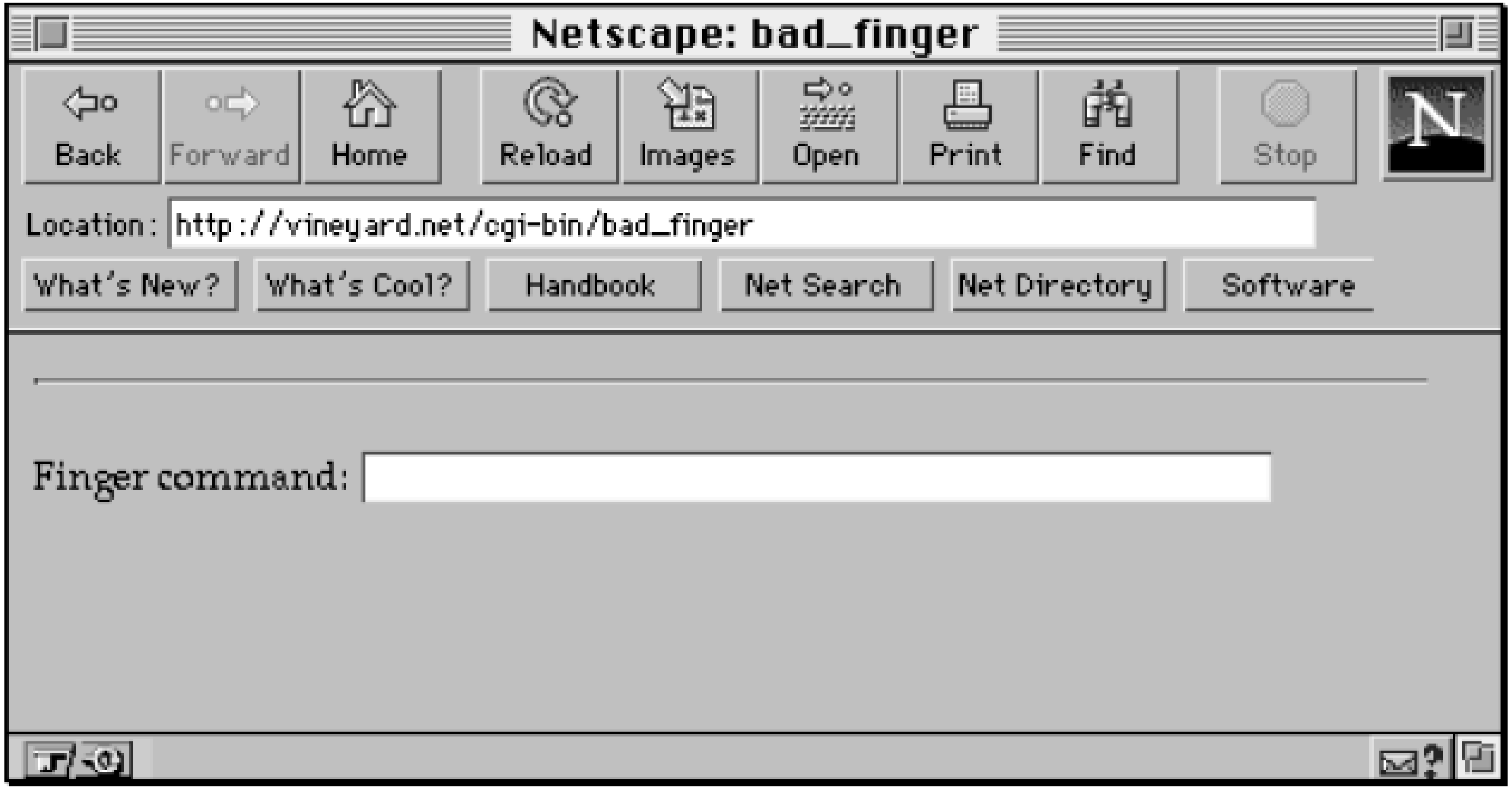
- Interpreters and shells should **never** be placed in cgi-bin.
- **Examples**:
  - Perl interpreter (PERL.EXE) on Windows
- Attackers can run **arbitrary commands** if such programs exist.
- Search engines can locate misconfigured servers automatically.
- Default scripts may remain installed even after upgrades.
- **Example**: **phf script**
  - Distributed with NCSA and early Apache servers
  - Allowed attackers to retrieve system files
- Demonstrates **unintended side effects.**

Unintended Side Effects

- CGI script in **Example 16-1** is discussed.
- **Script contains:**
  - A safe form-handling function
  - A finger gateway program
- **Normal usage:**
  - Displays an HTML form
  - Accepts a user ID
- **Figure 16-1:**
  - Shows the finger form displayed in a web browser.
- **Figure 16-2:**
  - Shows expected output for a valid finger request.
- **Hidden flaw:**
  - Allows attackers to execute arbitrary commands.

- Security flaws can remain dormant for years.
- Some flaws may be intentional **back doors.**

Unintended Side Effects

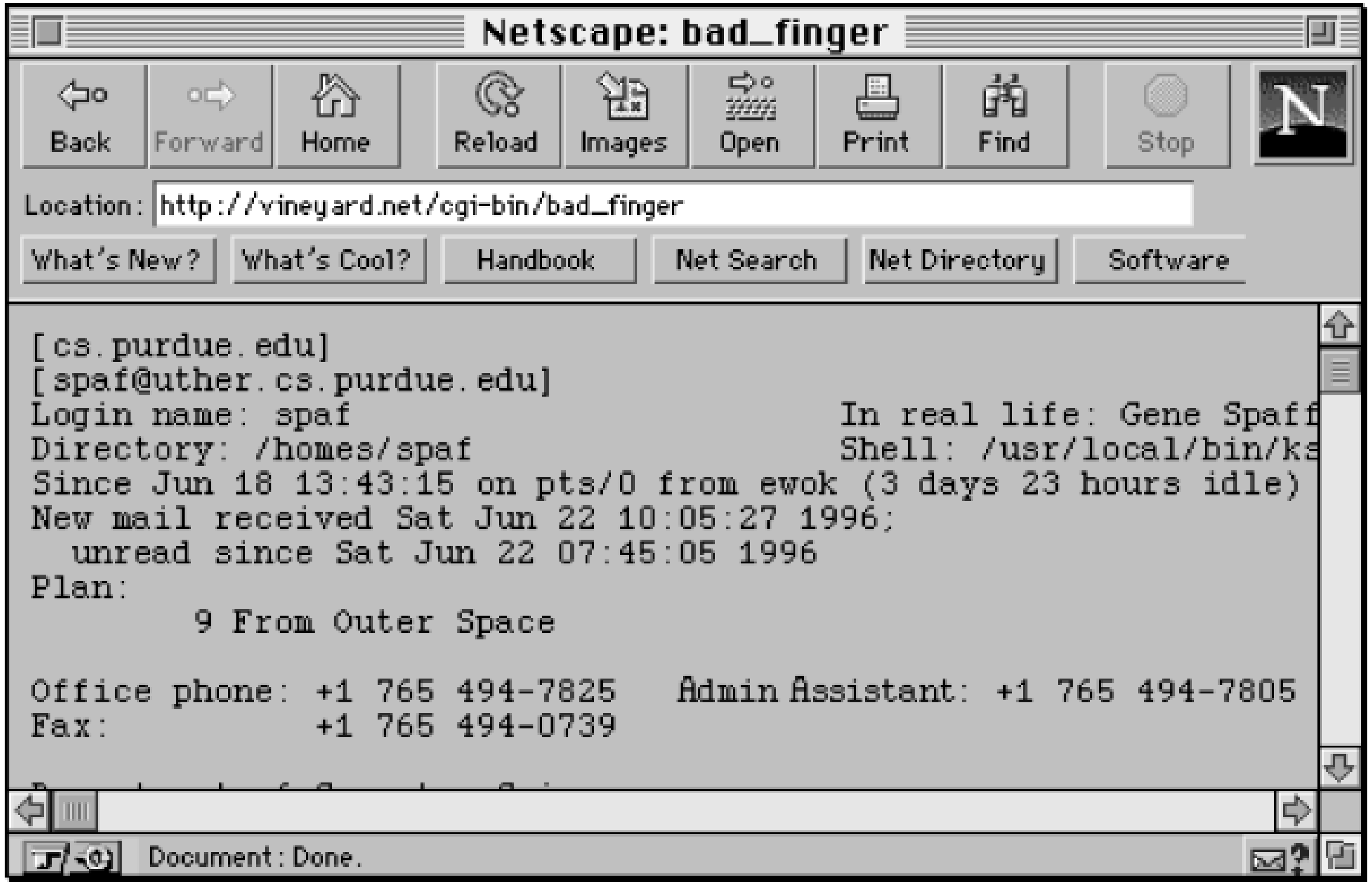


Location: http://vineyard.net/cgi-bin/bad_finger

Finger command:

Figure 16-1. The finger gateway

Unintended Side Effects



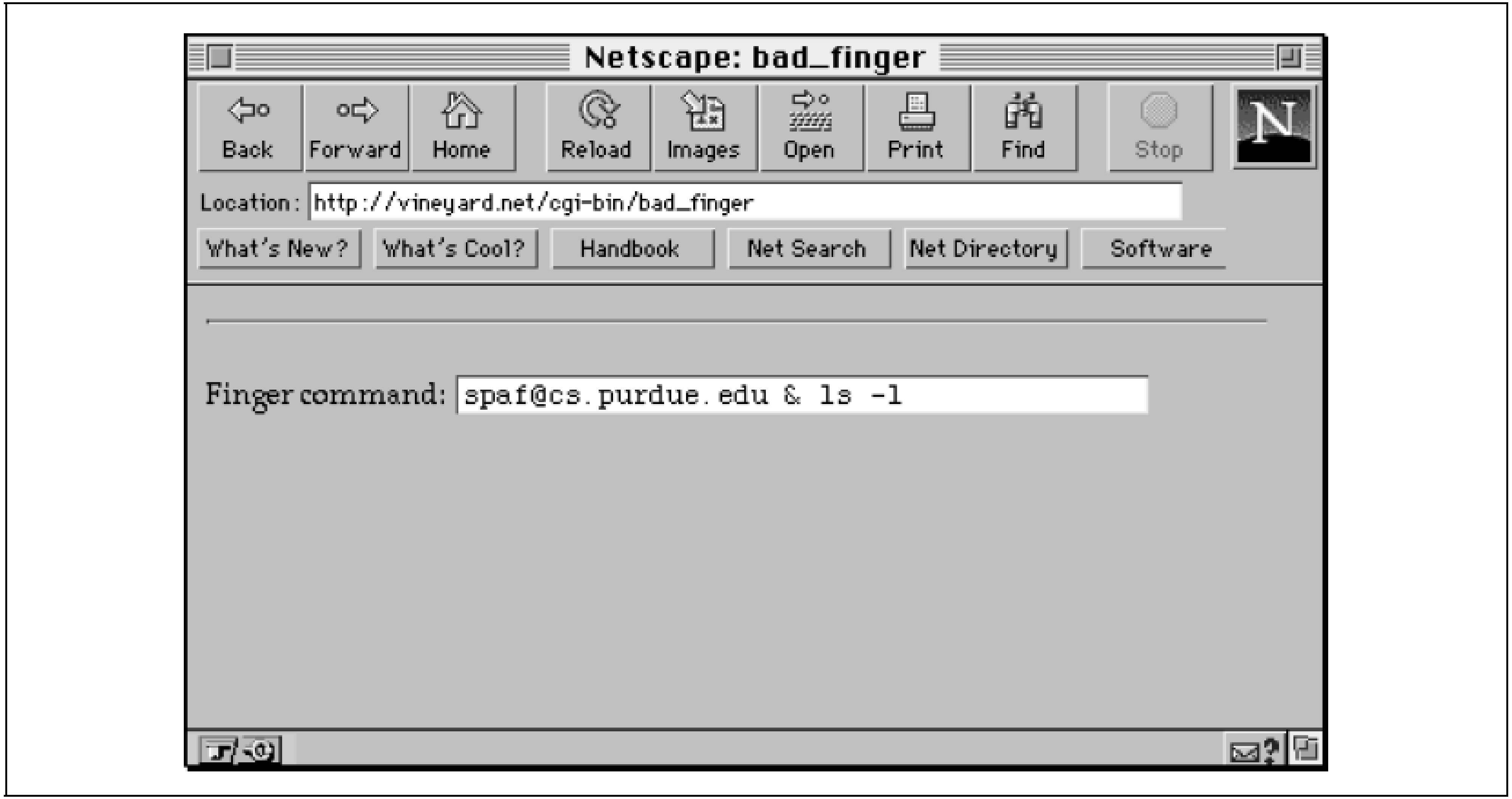
```
[cs.purdue.edu]
[spaf@uther.cs.purdue.edu]
Login name: spaf                          In real life: Gene Spaff
Directory: /homes/spaf                    Shell: /usr/local/bin/ks
Since Jun 18 13:43:15 on pts/0 from ewok (3 days 23 hours idle)
New mail received Sat Jun 22 10:05:27 1996;
  unread since Sat Jun 22 07:45:05 1996
Plan:
        9 From Outer Space

Office phone: +1 765 494-7825   Admin Assistant: +1 765 494-7805
Fax:          +1 765 494-0739
```

Figure 16-2. The form displayed by the finger script

The problem with the script

- Vulnerable line:
- print `/usr/bin/finger $input{'command'}`;
- Uses Perl backquotes, which invoke the **Unix shell**.
- Shell interprets special characters.
- Normal execution:
- /usr/bin/finger spaf@cs.purdue.edu
- Unix shell allows multiple commands per line.
- Attacker input:
- spaf@cs.purdue.edu & /bin/ls -l
- **Figure 16-3:**
  - Shows malicious input entered into the form.

The problem with the script



Location: http://vineyard.net/cgi-bin/bad_finger

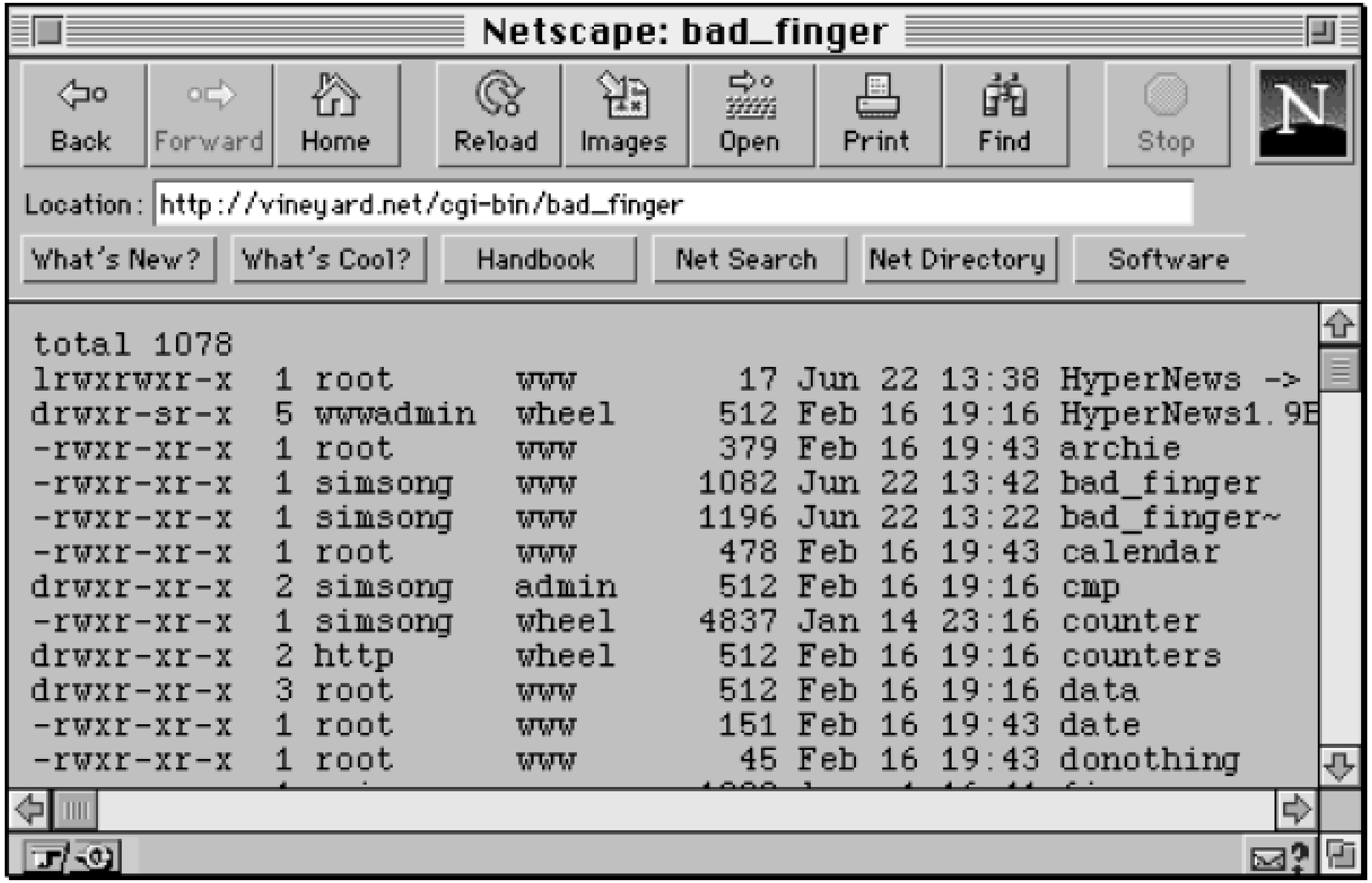
Finger command: spaf@cs.purdue.edu & ls -l

Figure 16-3. Attacking the bad_finger script

The problem with the script (cont..)

- **Figure 16-4:**
  - Shows directory listing output returned by the script.
- Potential attacker actions:
  - View confidential files
  - Delete data
  - Launch denial-of-service attacks
  - Gain remote shell access
- Key lesson: **Never allow arbitrary command execution.**

The problem with the script (cont..)



Location: http://vineyard.net/cgi-bin/bad_finger

```
total 1078
lrwxrwxr-x  1 root      www        17 Jun 22 13:38 HyperNews ->
drwxr-sr-x  5 wwwadmin  wheel     512 Feb 16 19:16 HyperNews1.9E
-rwxr-xr-x  1 root      www       379 Feb 16 19:43 archie
-rwxr-xr-x  1 simsong   www      1082 Jun 22 13:42 bad_finger
-rwxr-xr-x  1 simsong   www      1196 Jun 22 13:22 bad_finger~
-rwxr-xr-x  1 root      www       478 Feb 16 19:43 calendar
drwxr-xr-x  2 simsong   admin     512 Feb 16 19:16 cmp
-rwxr-xr-x  1 simsong   wheel    4837 Jan 14 23:16 counter
drwxr-xr-x  2 http      wheel     512 Feb 16 19:16 counters
drwxr-xr-x  3 root      www       512 Feb 16 19:16 data
-rwxr-xr-x  1 root      www       151 Feb 16 19:43 date
-rwxr-xr-x  1 root      www        45 Feb 16 19:43 donothing
```

Figure 16-4. Output from the bad_finger script under attack

Fixing the problem

- Never trust user input.
- Filter input to allow **only legal characters**.
- Use **whitelisting**, not blacklisting.

Example:

  - Accept only alphanumeric characters, @, ., and –
- This blocks shell metacharacters such as:
  - &, ;, '
- Using character selection is safer than filtering disallowed characters.
- Input rules depend on:
  - Data type
  - Shell and program behavior

Avoiding the Shell Entirely

- Use Perl’s system() function instead of backquotes.
- Prevents shell invocation.
- Improves security and performance.
- Directly executes the command with arguments.

Rules to Code By

- Most security flaws are programming bugs.
- Secure programs are also more reliable.

General Principles for Writing Secure Scripts

- Design before coding.
- Review design with another programmer.
- Write and test small sections.
- Check all values provided by the user.
- Validate arguments passed to system functions.
- Check all system call return codes.
- Use internal consistency checks.
- Include extensive logging.
- Avoid logging sensitive data.
- Keep critical code small.
- Review code from an attacker’s perspective.
- Use full pathnames.

General Principles for Writing Secure Scripts (cont..)

- Set the working directory explicitly.
- Test with expected and unexpected input.
- Be aware of race conditions.
- Disable core dumps.
- Avoid world-writable directories.
- Do not trust source IP addresses.
- Implement load limiting.
- Use execution time limits.
- Set CPU usage limits.
- Avoid plaintext reusable passwords.
- Conduct peer code reviews.
- Reuse trusted, audited code.

The Seven Design Principles of Computer Security

- Least privilege
- Economy of mechanism
- Complete mediation
- Open design
- Separation of privilege
- Least common mechanism
- Psychological acceptability

Securely Using Fields, Hidden Fields, and Cookies

- Web applications split code between:
  - Server
  - Browser
- Attackers can:
  - Modify form data
  - Bypass JavaScript
  - Send forged requests
- Browser-stored data must always be **validated on the server**.

Using Fields Securely

- Filter every field.
- Validate length.
- Verify selection list values.
- Always revalidate on the server.

Hidden Fields and Compound URLs

- Hidden fields store data in browser memory.
- Used for:
  - Session tracking
  - Shopping carts
- URLs can embed parameters directly.
- Problems:
  - Back button issues
  - Shared computers
  - Log file exposure
  - User manipulation
- Must defend against modified submissions.

Using Cookies

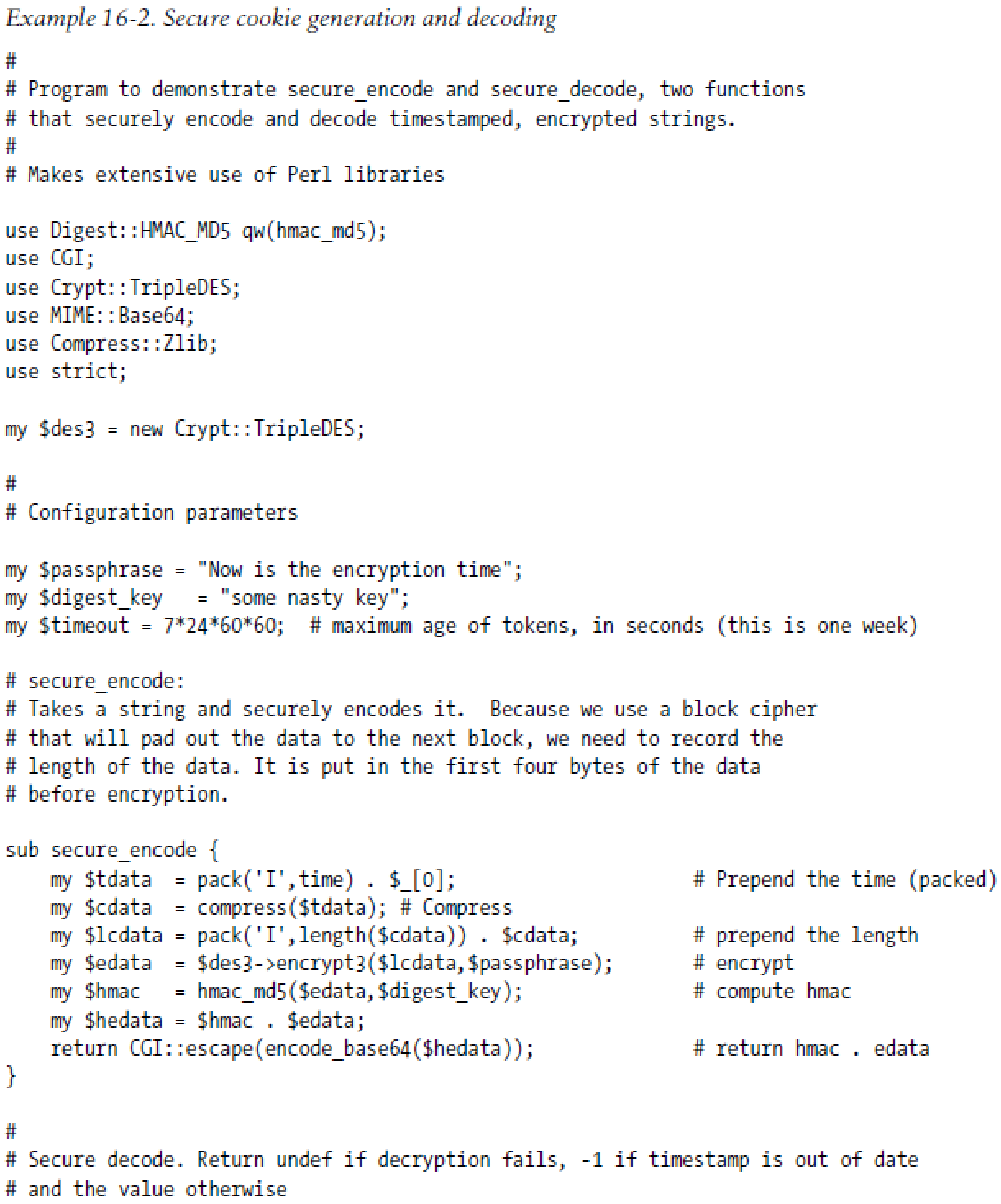
- Cookies store client-side state.
- Users can modify cookies.
- Problems include:
  - Reuse after expiration
  - Long-term storage
  - User distrust

Using Cryptography to Strengthen Hidden Fields, Compound URLs, and Cookies

- Cryptography:
  - Protects confidentiality
  - Detects tampering
- Human-readable data replaced with encrypted blocks.
- Process includes:
  - Marshalling
  - Timestamping
  - Compression
  - Encryption
  - HMAC
  - Base64 encoding
- Prevents replay and modification attacks.

Example 16-2. Secure cookie generation and decoding

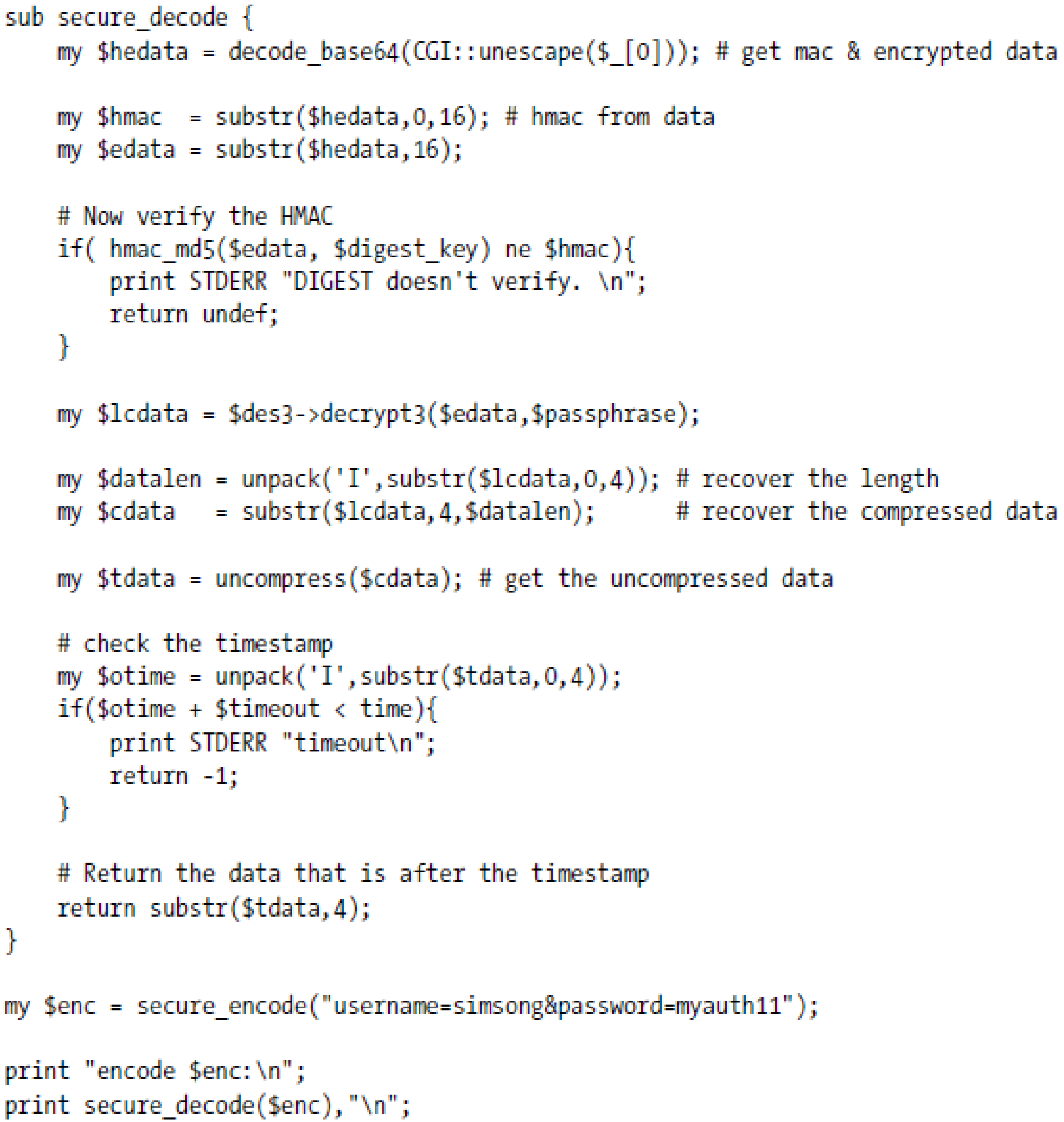
- Demonstrates secure_encode() and secure_decode().
- **Uses**:
  - HMAC-MD5
  - TripleDES
  - Compression
  - Base64 encoding
- Efficient even on slow hardware.
- Shows cryptography can be **practical and fast**.



Example 16-2. Secure cookie generation and decoding

```
#
# Program to demonstrate secure_encode and secure_decode, two functions
# that securely encode and decode timestamped, encrypted strings.
#
# Makes extensive use of Perl libraries

use Digest::HMAC_MD5 qw(hmac_md5);
use CGI;
use Crypt::TripleDES;
use MIME::Base64;
use Compress::Zlib;
use strict;

my $des3 = new Crypt::TripleDES;

#
# Configuration parameters

my $passphrase = "Now is the encryption time";
my $digest_key   = "some nasty key";
my $timeout = 7*24*60*60;  # maximum age of tokens, in seconds (this is one week)

# secure_encode:
# Takes a string and securely encodes it.  Because we use a block cipher
# that will pad out the data to the next block, we need to record the
# length of the data. It is put in the first four bytes of the data
# before encryption.

sub secure_encode {
    my $tdata  = pack('I',time) . $_[0];                     # Prepend the time (packed)
    my $cdata  = compress($tdata); # Compress
    my $lcdata = pack('I',length($cdata)) . $cdata;          # prepend the length
    my $edata  = $des3->encrypt3($lcdata,$passphrase);       # encrypt
    my $hmac   = hmac_md5($edata,$digest_key);               # compute hmac
    my $hedata = $hmac . $edata;
    return CGI::escape(encode_base64($hedata));              # return hmac . edata
}

#
# Secure decode. Return undef if decryption fails, -1 if timestamp is out of date
# and the value otherwise
```

Example 16-2. Secure cookie generation and decoding (continued)



```
sub secure_decode {
    my $hedata = decode_base64(CGI::unescape($_[0])); # get mac & encrypted data

    my $hmac  = substr($hedata,0,16); # hmac from data
    my $edata = substr($hedata,16);

    # Now verify the HMAC
    if( hmac_md5($edata, $digest_key) ne $hmac){
        print STDERR "DIGEST doesn't verify. \n";
        return undef;
    }

    my $lcdata = $des3->decrypt3($edata,$passphrase);

    my $datalen = unpack('I',substr($lcdata,0,4)); # recover the length
    my $cdata   = substr($lcdata,4,$datalen);      # recover the compressed data

    my $tdata = uncompress($cdata); # get the uncompressed data

    # check the timestamp
    my $otime = unpack('I',substr($tdata,0,4));
    if($otime + $timeout < time){
        print STDERR "timeout\n";
        return -1;
    }

    # Return the data that is after the timestamp
    return substr($tdata,4);
}

my $enc = secure_encode("username=simsong&password=myauth11");

print "encode $enc:\n";
print secure_decode($enc),"\n";
```

Rules for Programming Languages

Rules for Perl:

To secure Perl scripts, especially CGI programs:

1. Use Perl’s Tainting Features
   - Enable tainting with -T at the beginning of scripts.
   - Tainting marks all user-supplied variables as “tainted.”
   - Tainted variables cannot be used in unsafe operations (e.g., file opening, system calls).
   - Untaint variables using Perl string match operations.
   - Figure/Example: Example 16-2 shows secure cookie generation and decoding, using functions secure_encode() and secure_decode().
2. Set PATH Environment Variable
   - Must be a known safe value before calling system().

Rules for Programming Languages (cont..)

3. Filenames
    - Perl ignores tainting for read-only files; always untaint filenames used in writing operations.
4. SUID Scripts
    - Use Perl’s emulation mode to handle SUID scripts safely on older Unix systems.
5. PATH Security
    - Always set the program’s PATH variable, even if not running SUID or Unix.
6. Interpreter and Libraries Security
7. Ensure Perl interpreter and libraries are modifiable only by the administrator.

Security-Related CGI/API Variables

- HTTPS_RANDOM: 256-bit random value for each CGI invocation (Netscape).
- REMOTE_HOST: Hostname of client machine (e.g., dialup10.vineyard.net).
- REMOTE_USER: Authenticated username (e.g., simsong).
- REMOTE_ADDR: Client IP address (e.g., 204.17.195.47).
- AUTH_TYPE: Type of authentication (e.g., Basic).

Rules for C

- Writing secure C programs is **harder than Perl** because C lacks automatic memory management.
- **Perl advantage**: smaller, modular code; automatic memory handling.
- **C advantage**: speed, especially for CGI programs.

Security Guidelines for C

1. Check buffer boundaries when manipulating strings.
2. Use caution with unsafe library calls:
   - sprintf(), scanf(), sscanf(), vsprintf(), realpath(), getopt(), getpass(), etc.
3. Watch for functions returning pointers to static storage; attackers can overflow buffers.
4. Use ANSI C compiler with function prototypes. Consider analysis tools like Purify.
5. Enable compiler warnings:
   - GNU C: -Wall
   - MS VC++: /W4
   - Replace unsafe functions:

Avoid	Use instead
gets()	fget()
strcpy()	strncpy()
strcat()	strncat()

Security Guidelines for C (cont..)

6. **File creation:**
   - New files: use O_EXCL | O_CREAT.
   - Existing files: omit O_CREAT.
   - Temporary files: tmpfile() or mkstemp() (avoid mktemp(); vulnerable to race conditions).

Rules for the Unix Shell

- Avoid writing CGI scripts with **sh, csh, ksh, bash, tcsh** except for trivial scripts.
- Security issues are abundant; easy to make mistakes.

Using PHP Securely

Introduction to PHP

- Server-side scripting language, originally Personal Home Page → PHP3 → PHP Hypertext Preprocessor.
- Runs on **Unix/Windows** with **Apache/IIS**.
- **Advantages**:
  - Fast execution; interpreter built into web server.
  - No special directory/executable required.
  - Error display directly on web page.
  - Database connection caching (MySQL).
  - Powerful: open files, network connections, execute programs.

Using PHP Securely (cont..)

Example PHP Script:

```
<html><head><title>PHP Test</title></head>
<body>
<?php
echo "Hello World!<p>";
?>
</body></html>
```

- PHP code enclosed in <?php ... ?>.
- Variables: begin with $, untyped, auto-substituted in double-quoted strings.

Controlling PHP

- **php.ini or Apache httpd.conf** controls behavior.
- **Example**: enabling **PHP3 safe mode** in /htdocs but not /staffdocs.

```
<Directory /htdocs/>
php3_safe_mode on
</Directory>
<Directory /staffdocs/>
php3_safe_mode off
</Directory>
```

Understanding PHP Security Issues

- **Shared hosting:** users may access others' files.
- **Lax variable protections:** default globals, hidden backdoors, downloaded scripts.

PHP Installation Issues

- Recommended as **Apache module** (faster).
- If installed as executable: place outside web hierarchy (/usr/local/bin/php).

PHP Variables

- Global variables include:
  - CGI environment variables (HTTP_USER_AGENT, DOCUMENT_ROOT)
  - GET, POST, Cookie, Server variables
  - Variables in libraries
- **Danger**: variable shadowing; attackers can override expected values.

Example: Global Variable Attack

- $MAILDIR normally /var/spool/mail
- URL ?MAILDIR=/etc/passwd overrides variable.
- **Solution**: manually initialize variables:

```
$authorized = 0;
if(validate_user($user,$pass)) {
  $authorized = 1;
}
```

- Best practice: set register_globals = off.

Database Authentication

- Avoid hardcoding usernames/passwords in scripts.
- Better: store passwords in secure file and read them.

```
$fp = fopen("/usr/local/adm/dbpasswords/http", "r");
$pass = fgets($fp,14);
fclose($fp);
mysql_pconnect("mysql.vineyard.net","http",$pass);
```

URL fopen()

- PHP can open URLs with fopen().
- Risk: attacker can manipulate include files via globals.
- **Example**: main.php includes loadlanguage.php; attacker sets $langDir to external URL.

Hiding PHP Scripts

- Keep scripts private; ensure always processed by PHP.
- Avoid exposing debugging variables (debug, showerrors).
- Web server configuration can hide PHP:

```
AddType application/x-httpd-php .bop .foo .133t
# or parse all HTML with PHP
AddType application/x-httpd-php .htm .html
```

PHP Safe Mode

- Disables dangerous functions based on script location.
- Useful for shared servers (ISPs).
- Restrictions include:
  - File operations limited to UID of script owner.
  - system() only executes scripts in safe_mode_exec_dir.
  - dl(), backticks, shell_exec() disabled.

Scripts with Additional Privileges

- Avoid SUID/SGID unless necessary.
- Scripts running with higher privileges are common security risks.

PHP Configuration File Settings

Shaun Clowes' Recommendations for Securing PHP Environments:

- **set register_globals=off**
  - Prevents users from setting variables in PHP scripts.
- **set safe_mode=on**
  - Enables PHP safe mode, improving security.
  - Especially recommended for ISP environments.
  - Quote: "This is a great option for ISP environments… but it can also be a complete pain in the neck."
- **set open_basedir**
  - Restricts PHP to a specified directory hierarchy.

PHP Configuration File Settings (cont..)

- **set display_errors=off, log_errors=on**
    - Writes errors to a log file instead of the web browser.
    - Makes debugging harder but prevents attackers from reverse-engineering scripts.
    - Recommendation: On development systems, display_errors=on; on production, display_errors=off.
- **set allow_url_fopen=off**
- Prevents PHP from opening URLs when expecting files.

Writing Scripts with Additional Privileges

1. Use SUID root carefully:
   - Needed only for tasks requiring superuser access (e.g., modifying /etc/passwd).
   - For restricted database access, create a special Unix user and SUID scripts to that user.
2. Separate SUID functionality:
   If superuser access is rarely needed, isolate SUID operations in a separate program with controlled interface.
3. Revoke privileges quickly:
   - Use SUID/SGID early in the program and return effective/real UID/GID to normal immediately after use.
4. Avoid shell scripts for SUID:
   - Especially csh and derivatives.
5. Use separate users/groups per application:
   - Prevents abuse amplification.

Writing Scripts with Additional Privileges (cont..)

6. Use setuid() and setgid() functions to bracket privileged code:
7. setuid(0); // Become superuser to open master file
8. fd = open("/etc/masterfile", O_RDONLY);
9. setuid(-1); // Revoke superuser
10. if(fd<0) error_open(); // Handle errors
11. Use full pathnames for all file operations.
12. Use chroot() for further restriction:
    - Changes root directory to limit process access.
    - Example: Restrict program to /usr/local/logs:
    - chroot("/usr/local/logs");
    - Recommended only for CGI programs, not API modules.
    - Easier to implement in Perl than C.

Connecting to Databases

- CGI scripts often connect to external databases for:
  - User preferences
  - Shopping carts
  - Order processing
- **Security concerns:**
  - Each script execution may open a new connection, or use persistent connections.
  - Database-backed websites are powerful but can reduce overall security if attackers execute arbitrary SQL.
  - Example: Theft of credit card numbers due to insecure database access.

Protect Account Information

- Databases require **username/password authentication**.
- Common but unsafe practice: Hard-coding credentials in scripts.

  Problems:

  - Scripts can be viewed by attackers → credentials exposed
  - Multiple scripts may require the same credentials → redundancy
  - Changing credentials requires editing multiple scripts → risk of mistakes

Better approach: Store credentials in a separate file, read them at runtime.

- $fp = fopen("/usr/local/adm/dbpasswords/http", "r");
- $pass = fgets($fp,14);
- fclose($fp);
- mysql_pconnect("mysql.vineyard.net","http",$pass);

Use Filtering and Quoting to Screen Out Raw SQL

- Always filter user input to ensure only allowable characters.
- Properly quote user data before sending to SQL server.
- Unsafe example:
- $name = param('name');
- sql_send("insert into names (name) value ('$name');");
  - Input "Simson Garfinkel')"; delete from names; results in:
  - insert into names (name) value ('Simson Garfinkel')"; delete from names; ');
  - Executes insertion, deletion, and generates a SQL error.

Use Filtering and Quoting to Screen Out Raw SQL (cont..)

Safe approach:

- Use a quote function:
- sub squote {
-   my $ret = $_[0];
-   $ret =~ s/\'/\\\'/g;
-   return '\'' . $ret . '\'';
- }
- $qname = squote(param('name'));
- sql_send("insert into names (name) value ($qname);");
- Or use variable binding with precompiled SQL queries:
- $func = sql_compile("insert into name (name) value (@)");
- $name = param('name');
- sql_bind($func,1,$name);
- sql_exec($func);

Protect the Database Itself

- **Network security:**
  - Use firewalls to prevent outside access.
  - Recommended: Separate Ethernet adapters and firewall appliance between web server and database (Figure 16-5).

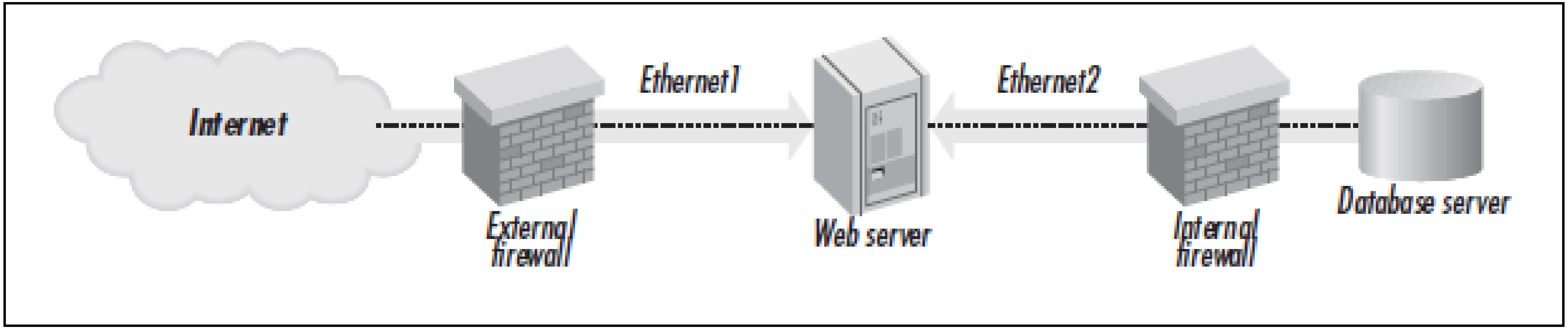


Figure 16-5. Connecting a database server and a web server to the Internet and your internal network with multiple firewalls.

Protect the Database Itself (cont..)

- **Limit logins:**
  - Only system administrators and DB admins should have login access.
- **Physical and maintenance security:**
  - Ensure database server is backed up, physically secure, and maintained like other critical servers.