WEB SECURITY
UNIT - II

Outline:

1. Privacy and Security for Users
    - The Web's War on Your Privacy
    - Privacy-Protecting Techniques
    - Privacy-Protecting Technologies
    - Backups and Antitheft
2. Web Server Security
    - Physical Security for Servers
    - Host Security for Servers
    - Securing Web Applications

Privacy and Security for Users

1. The Web's War on Your Privacy

- The Web is **bidirectional**: users watch content, while web servers **monitor user activity**.
- Each page visit is typically **recorded by a computer**, contributing to a growing **user profile**.
- Visiting different types of sites (e.g., parenting, consumer electronics) allows systems to infer **interests and behavior**.
- **Registration with email addresses** often leads to marketing emails and "special offers."

Web Opportunities vs. Privacy Costs

- The Web benefits consumers but also empowers:
    - Marketers
    - Sales organizations
    - Tricksters and criminals
- Unlike billboards (which measure traffic volume), **web advertising collects detailed personal data**:
    - Location
    - Internet access type
    - Browsing history
    - Email addresses
- Web ads are **measurable and analyzable**, but this requires **extensive data collection**, reducing privacy.

Technology and Privacy

- The Internet was designed to **transfer information**, not protect privacy.

- Many **web technologies enable tracking**, often invisibly.
- This chapter introduces privacy threats.
- Later chapters (9 and 10) discuss **privacy-protection techniques and software**.

Understanding Privacy

- Privacy definitions vary.
- Merriam-Webster defines privacy as:
    - Being apart from observation
    - Freedom from unauthorized intrusion

The Tort of Privacy

- In 1890, **Warren and Brandeis** argued for a legal **right to privacy**.
- Privacy protects individuals from:
    - Undesired publicity
    - Exposure of personal matters without consent
- Key principles:
    - Truth is **not a defense**
    - Lack of malice is **not a defense**

Four Privacy Torts in U.S. Law

1. **Privacy intrusion**
    - Intruding into a private sphere.
2. **Disclosure of private facts**
    - Publishing private information with no public interest.
3. **Portrayal of information in false light**
    - True or false information that misleads.
4. **Appropriation**
    - Using a person's name or likeness commercially without permission.

- These torts do not fully address **computer-age privacy threats**.

Informational Privacy (Westin)

- Alan Westin (1967) defined **informational privacy** as:
  Control over when, how, and to what extent information is communicated.
- This definition is especially relevant to the **Web**.
- Most modern privacy violations involve **loss of control over personal data**.

Personal, Private, and Personally Identifiable Information
Types of Information

- **Personal information**
    - Name, birth date, education, family.
- **Private information**
    - Personal information not generally known.
    - Some protected by law (education records, bank records).
    - Privacy depends on **context**.

- **Personally identifiable information (PII)**
    - Data that reveals identity (name, account number).
- **Anonymized information**
    - Modified so identities cannot be discerned.
- **Aggregate information**
    - Statistical summaries (e.g., Census tract data).

Triangulation

- Combining anonymized or aggregate data can reveal identities.
- Example: Zip code + birthday can uniquely identify individuals.
- Even "aggregate" questions may request **PII unintentionally**.

User-Provided Information

- Users often provide:
    - Name, address, credit card details
    - Email addresses and phone numbers
- Merchants store data in **user accounts** with usernames/passwords.
- Security questions (e.g., mother's maiden name) are used for password recovery.
- **Figures 8-1 and 8-2** illustrate:
    - Registration forms
    - Account information collection processes
- U.S. law places **few restrictions** on how websites use collected data.
- Privacy policies are **voluntary**.

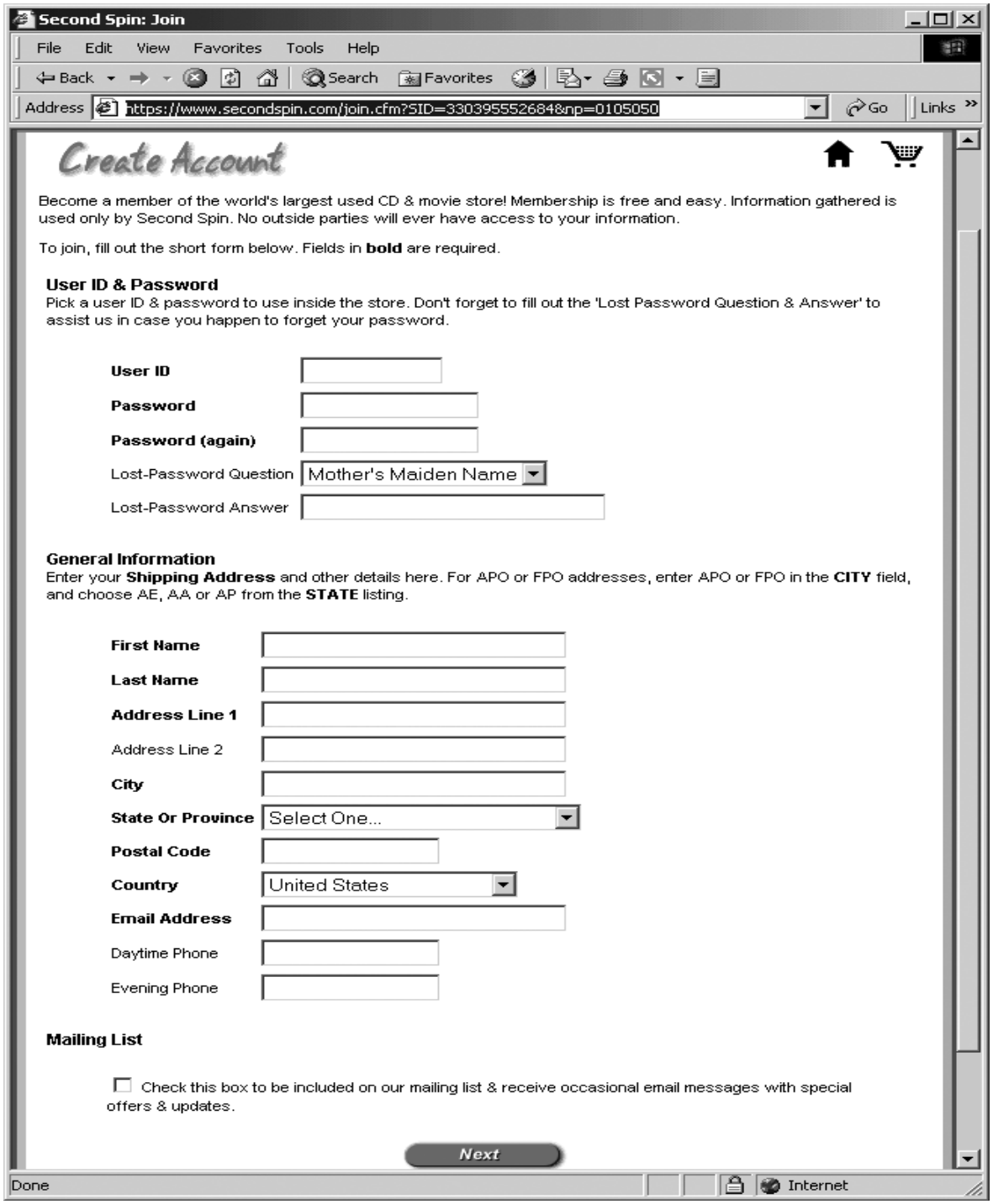

Figure 8-1. By far, the greatest kind of personal information on the Web today is the information provided by consumers when they register at web sites.

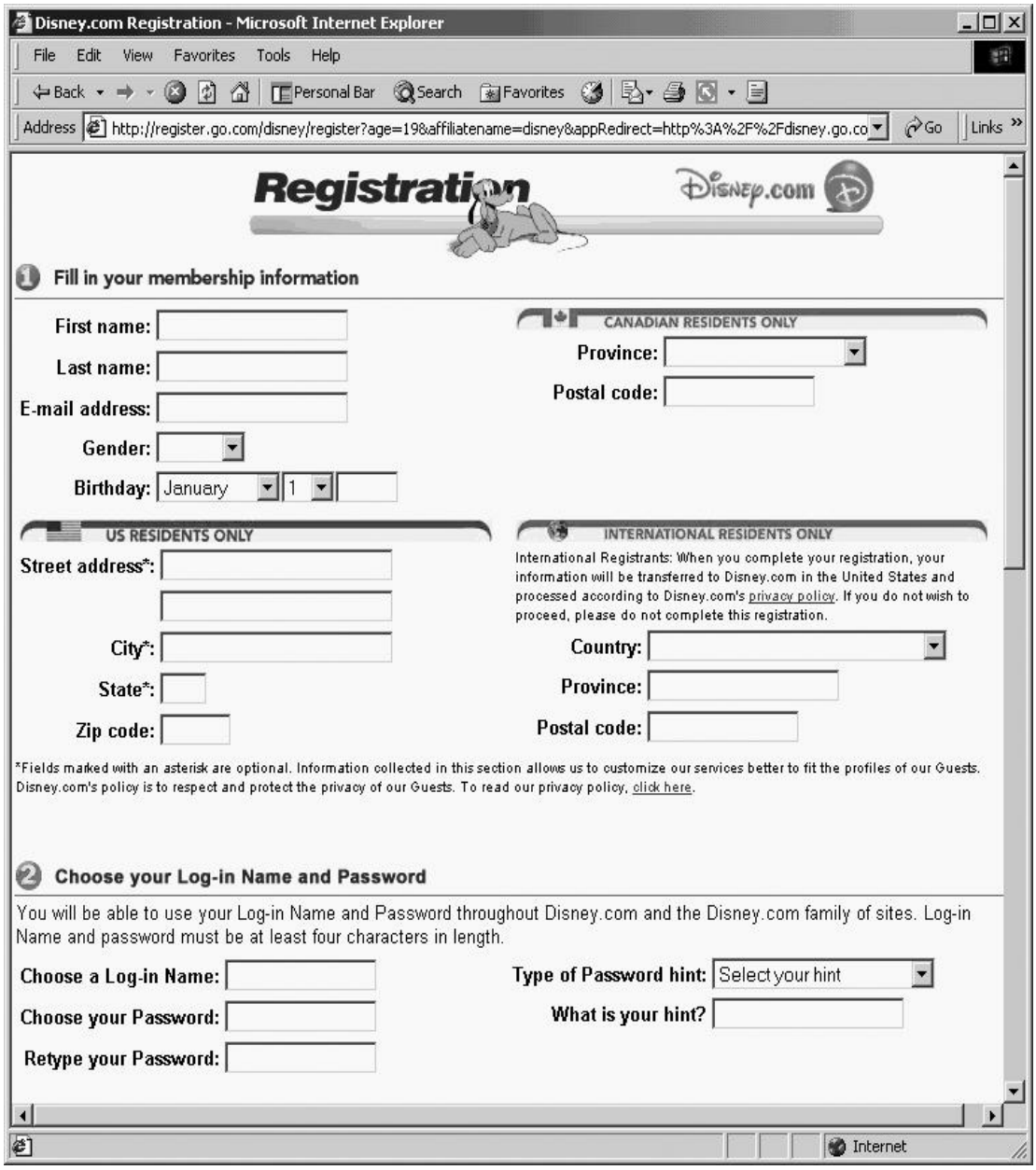

Figure 8-2. Disney's registration page for adults asks for name, email address, gender, and birthday, in addition to mailing address. Many people are surprised how identifying even simple demographic information can be. For example, in many cases a person can be uniquely identified by day of birth (without the year) and Zip code.

Log Files

- Log files record network and user activity.
- Created for:
  - Debugging
  - Maintenance
  - Marketing
  - Government investigations
- Users generally **cannot know** what is logged.

Retention and Rotation

- **Rotation**: automatic deletion of old logs.
- Some systems retain logs until manually deleted.
- Backup systems (e.g., magnetic tape) may preserve logs for years.
- Example web server logs show:
    - access_log
    - access_log.1
    - access_log.2.gz
- These indicate **compressed and rotated logs**.

Web Logs

- Each page request generates entries on:
    - Web servers
    - Databases
    - Firewalls
    - Proxies
- Logs can be subpoenaed or misused.
- Most logs are **never reviewed**, yet store extensive data.

What's in a Web Log?

- IP address and hostname
- Timestamp
- Requested URL
- Browser type
- Refer link
- Errors
- Authentication usernames
- Logs can be cross-correlated to identify users.

Example 8-1

- Shows a **typical web server log**
- Demonstrates:
    - IP tracking
    - Browser identification
    - Refer links

Example 8-1. A sample web server log

```
free-dial-77.freeport.mwci.net - - [09/Mar/1997:00:04:11 -0500] "GET /awa/issue2/
    Woodstock.gif HTTP/1.0" 200 26385
"http://www.vineyard.net/awa/issue2/Wood.html" "Mozilla/2.0 (compatible; MSIE 3.01;
    Windows 95)" ""
free-dial-77.freeport.mwci.net - - [09/Mar/1997:00:04:27 -0500] "GET /awa/issue2/
    WoodstockWoodcut.gif HTTP/1.0" 200 54467
"http://www.vineyard.net/awa/issue2/Wood.html" "Mozilla/2.0 (compatible; MSIE 3.01;
    Windows 95)" ""
crawl4.atext.com - - [09/Mar/1997:00:04:30 -0500] "GET /org/mvcc/ HTTP/1.0" 200 10768 "-"
    "ArchitextSpider" ""
www-as6.proxy.aol.com - - [09/Mar/1997:00:04:34 -0500] "GET /cgi-bin/imagemap/mvol/cat2.
    map?31,39 HTTP/1.0" 302 - "http://www.mvol.com/" "Mozilla/2.0 (Compatible; AOL-IWENG
    3.0; Win16)" ""
www-as6.proxy.aol.com - - [09/Mar/1997:00:04:40 -0500] "GET /mvol/photo.html HTTP/1.0" 200
    6801
"http://www.mvol.com/" "Mozilla/2.0 (Compatible; AOL-IWENG 3.0; Win16)" ""
www-as6.proxy.aol.com - - [09/Mar/1997:00:04:48 -0500] "GET /mvol/photo2.gif HTTP/1.0" 200
    12748
"http://www.mvol.com/" "Mozilla/2.0 (Compatible; AOL-IWENG 3.0; Win16)" ""
free-dial-77.freeport.mwci.net - - [09/Mar/1997:00:05:07 -0500] "GET /awa/issue2/Wood.html
    HTTP/1.0" 200 37016
"http://www.altavista.digital.com/cgi-bin/query?pg=q&what=web&fmt=.&q=woodstock" "Mozilla/
    2.0 (compatible; MSIE 3.01; Windows 95)" ""
free-dial-77.freeport.mwci.net - - [09/Mar/1997:00:05:07 -0500] "GET /awa/issue2/
    Sprocket1.gif HTTP/1.0" 200 4648
"http://www.vineyard.net/awa/issue2/Wood.html" "Mozilla/2.0 (compatible; MSIE 3.01;
    Windows 95)" ""
free-dial-77.freeport.mwci.net - - [09/Mar/1997:00:05:08 -0500] "GET /awa/issue2/
    Sprocket2.gif HTTP/1.0" 200 5506
"http://www.vineyard.net/awa/issue2/Wood.html" "Mozilla/2.0 (compatible; MSIE 3.01;
    Windows 95)" ""
www-as6.proxy.aol.com - - [09/Mar/1997:00:05:09 -0500] "GET /mvol/peter/index.html HTTP/1.
    0" 200 891 "http://www.vineyard.net/mvol/photo.html" "Mozilla/2.0 (Compatible; AOL-
    IWENG 3.0; Win16)" ""
```

The Refer Link Field

- Automatically sends the **previous URL**.
- Used to:
  - Measure advertisement effectiveness
  - Track navigation paths
- Can leak:
  - Search queries
  - Form data
- GET vs POST:
  - GET embeds data in URLs → higher privacy risk

Obscuring Web Logs

- **Proxy servers** hide user IP addresses.
- Proxies do not guarantee anonymity.
- Proxy logs can still identify users.

RADIUS Logs

- RADIUS authenticates dial-up users.
- Logs include:
  - Username
  - IP address
  - Session time
  - CALLER-ID
- Played a key role in identifying the **Melissa worm author**.

Mail Logs

- Track:
  - Sender and recipient
  - Time
  - Message ID
- Content usually not logged.
- Useful for identifying:
  - Communication patterns
  - Mailing list membership

DNS Logs

- DNS servers can log every query.
- Reveal:
  - Websites accessed
  - User behavior patterns
- Useful for maintenance and surveillance.

Understanding Cookies

- Cookies are **ASCII text blocks** stored by browsers.
- Sent automatically with each request.
- Introduced by Netscape Navigator 2.0.
- Used to maintain **state** across HTTP sessions.

The Cookie Protocol

- Cookies are set using **Set-Cookie headers**.
- Key attributes:
  - expires
  - domain
  - path
  - secure
- Cookies are sent back via HTTP headers.

Example Cookies

- HotBot sends multiple cookies (Table 8-1).
- Includes **third-party cookies** (e.g., .lycos.com).
- Cookies used for:

- Tracking
- Visitor counting
- Advertising profiles

Table 8-1. Cookies sent by www.hotbot.com at 8:10 a.m. EST on April 21, 2001

Cookie #	Content	Domain	Expires	Path
1	lubid=01000008C73351C5086C3AE177A40000351200000000	.lycos.com	18-Jan-2038 08:00:00 GMT	/
2	p_uniqid=aD3QMJX/K93Z		21-Dec-2012 08:00:00 GMT	/
3	remotehost=secondary=chi%2Emegapath&top=net		21-May-2001 07:00:00	/
4	HB%5FSESSION=BT=lowend&BA=false&VE=&PL=Unknown&MI=u&BR=Unknown&MA=0&BC=1			/

Cookie Uses

- Store user data directly
- Or store identifiers linked to databases
- Widely used for **advertising analytics**

Cookies and Privacy

- Cookies can:
  - Weaken privacy (profiling, tracking)
  - Improve privacy (store preferences locally)
- Example privacy-protecting cookie:
  - DigiCrime virus counter

Cookie Jars

- Cookies stored:
  - In memory
  - On disk if persistent
- Netscape:
  - cookies.txt (Example 8-2)
- Internet Explorer:
  - Individual files (Figure 8-3, Example 8-3)

Example 8-2. A sample Netscape cookies file

```
# Netscape HTTP Cookie File
# http://www.netscape.com/newsref/std/cookie_spec.html
# This is a generated file!  Do not edit.
.techweb.com     TRUE  /wire/news FALSE 942169160 TechWeb 204.31.228.79.852255600 path=/
.hotwired.com    TRUE  /   FALSE 946684799 p_uniqid  yQ63oN3ALxO1a73pNB
.talk.com        TRUE  /   FALSE 946684799 p_uniqid  y46RXMoBwFwD16ZFTA
.packet.com      TRUE  /   FALSE 946684799 p_uniqid  y86ijMoA9MhsGhluvB
.boston.com      TRUE  /   FALSE 946684799 INTERSE   stl-mo8-10.ix.netcom.
com20748850376179639
.netscape.com    TRUE  /   FALSE 1609372800 MOZILLA  MOZ-ID=DFJAKGLKKJRPMNX[-]MOZ_VERS=1.
    2[-]MOZ_FLAG=2[-]MOZ_TYPE=5[-]MOZ_CK=AJpzO85+6OjN_Ao1[-]
.netscape.com    TRUE  /    FALSE 1609372800 NS_IBD   IBD_
    SUBSCRIPTIONS=INCO05|INCO10|INCO17|INCO18|INCO20|INCO21|INCO22|INCO34|INCO46
www.xmission.com   FALSE  /  FALSE  946511999  RoxenUserID  0x7398
ad.doubleclick.net FALSE  /  FALSE  942191940  IAF  22348bb
.focalink.com     TRUE    /  FALSE  946641600  SB_ID  ads01.28425853273216764786
gtplacer.globaltrack.com  FALSE  /  FALSE 942105660  gtzopyid  85317245
.netscape.com     TRUE    /  FALSE  1585744496  REG_DATA  C_DATE_REG=13:06:51.304128 01/
    17/97[-]C_ATP=1[-]C_NUM=0[-]
www.digicrime.com   FALSE    FALSE  942189160  DigiCrime  virus=1
```

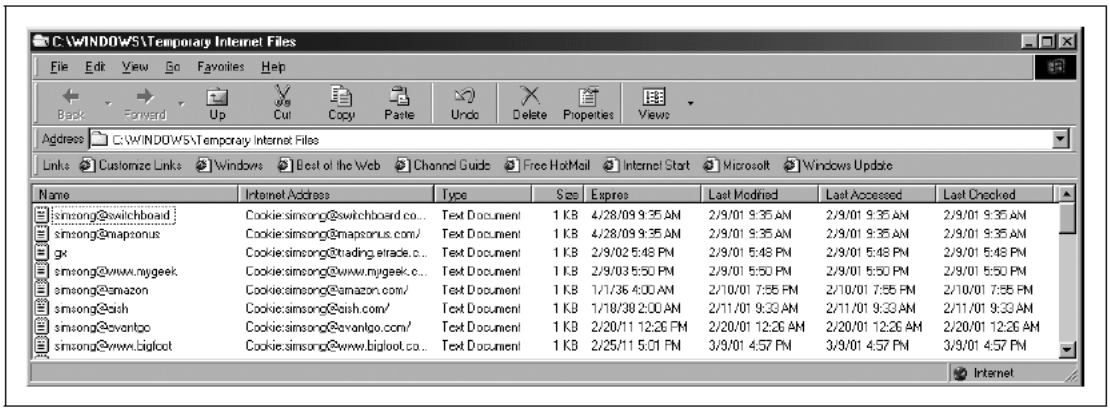

Figure 8-3. Internet Explorer stores cookies in files in the Cookies directory. You can delete a cookie by clicking on the cookie with the mouse and hitting the "Delete" key.

Example 8-3. The contents of an Internet Explorer Cookies file.

```
SITESERVER
ID=94e349397f0ba875c43fac4e1497ed69
caregroup.org/
0
642859008
31887777
514252192
29395648
*
```

Cookie Security

- Cookies can be edited by users.
- Secure cookies use:
    - Random IDs
    - Cryptographic MACs
- Examples compare insecure vs more secure cookies.
- Chapter 16 discusses secure cookie creation.

Disabling Cookies

- Browsers allow:
  - Accept all
  - Reject all
  - Prompt user
- Cookies already accepted cannot be selectively blocked.
- Advanced techniques include:
  - File permission tricks
  - Proxy filters
- **Figure 8-4** shows cookie management interface.

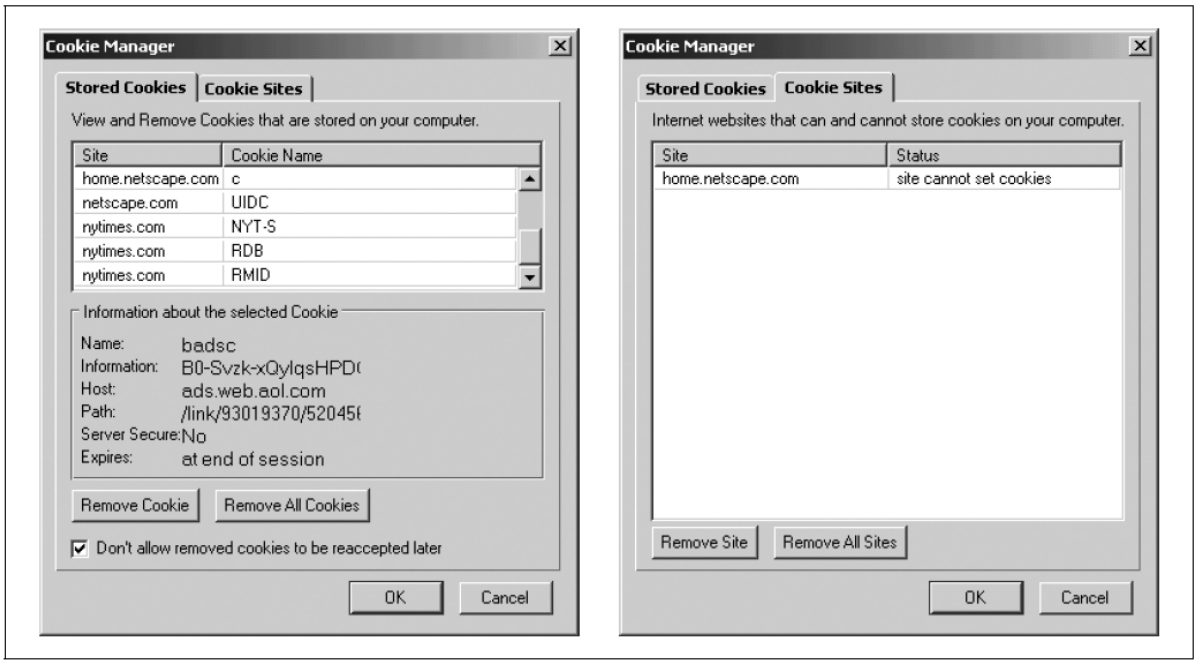

Figure 8-4. Netscape 6.0's Cookie Manager allows cookies to be controlled on a site-by-site basis

Web Bugs

- Introduced publicly in 2000 by the Privacy Foundation.
- Small invisible images (1×1 GIF).
- Also called:
  - Clear GIFs
  - Beacon GIFs

Web Bugs on Web Pages

- Example bugs from Quicken.COM:
  - Doubleclick
  - MatchLogic
- Enable third-party tracking without ads.
- Allow cross-database correlation.
- **Figure 8-5** shows web bug in Yahoo Profile.

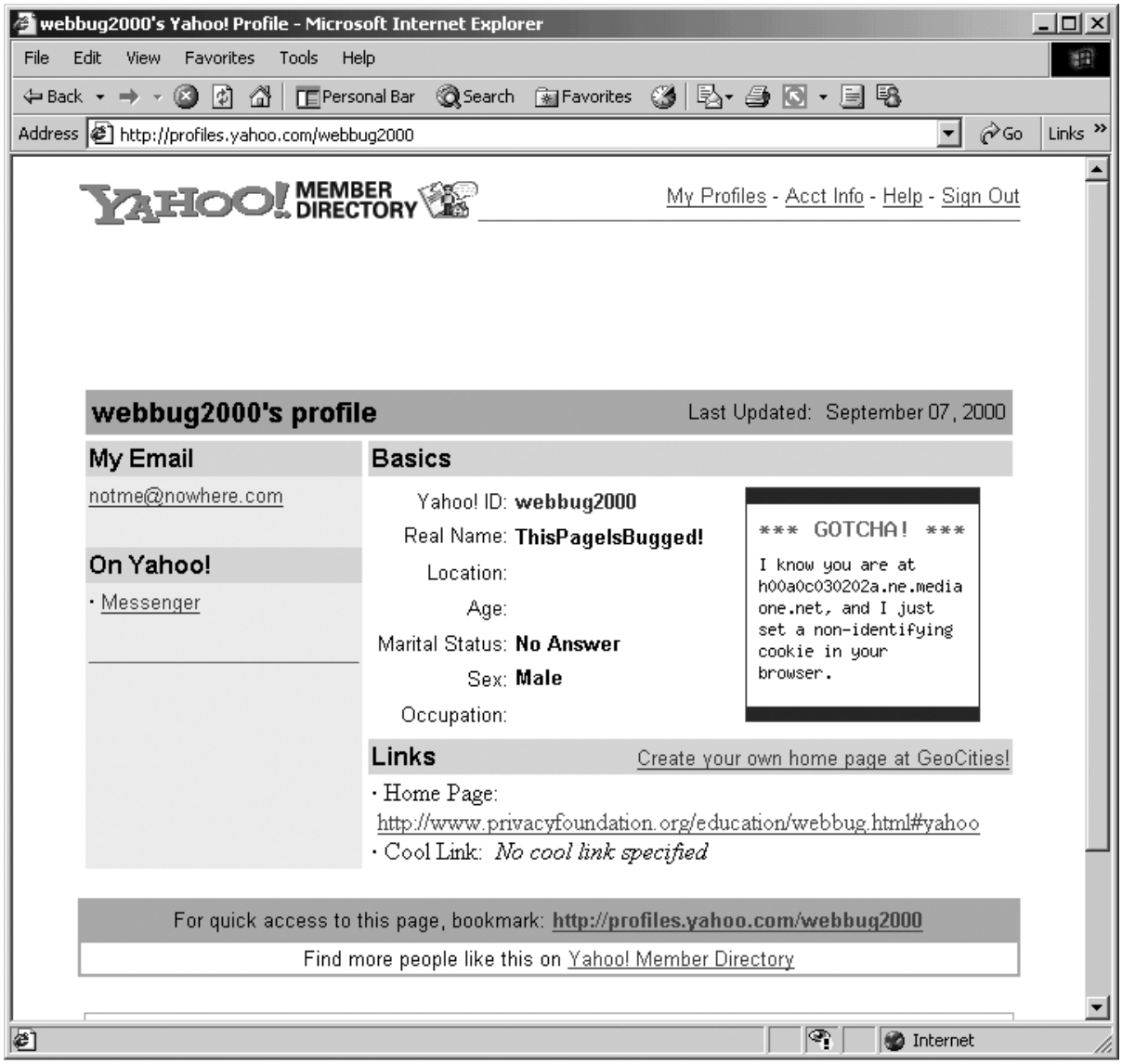

Figure 8-5. A Yahoo profile that was bugged with a web bug by the Privacy Foundation

Web Bugs in Email Messages and Word Files

- Used to detect:
  - Email reading
  - Forwarding
- Can be embedded in:
  - HTML email
  - Usenet messages
  - Microsoft Word documents

Uses of Web Bugs

- Usage statistics
- Cross-site tracking
- User profiling
- Cookie synchronization
- Ad verification

- Email tracking
- Detecting copyright infringement

2. Privacy-Protecting Techniques

- The Internet allows extensive **collection of personal information**.
- This chapter focuses on **practical techniques** to protect privacy.
- Techniques are mostly:
    - Simple
    - Common-sense
    - Immediately applicable
- Key areas covered:
    - Choosing a good service provider
    - Using strong passwords
    - Cleaning online traces
    - Avoiding spam and junk email
    - Protecting against identity theft
- Chapter 10 discusses **software-based privacy tools**.
- Legal aspects of privacy are discussed in **Chapter 24**.

Choosing a Good Service Provider

- The **most important privacy decision** is selecting a trustworthy ISP.
- ISPs can monitor:
    - Every website visited
    - Emails sent and received
    - Online behavior patterns
- Dial-up ISPs can infer:
    - When you are home
    - Travel or vacation periods
- ISPs may learn:
    - Workplace location (via email access)
    - User demographics

ISP Monitoring Practices

- Some ISPs:
    - Monitor activity for maintenance and research
    - Sell browsing data for revenue
- Monitoring may be:
    - Clearly disclosed
    - Or done silently
- New equipment allows ISPs to:
    - Monitor ads downloaded
    - Replace advertisements
    - Generate user-level statistics

ISP Privacy Policies

- Some ISPs enforce strict data-access rules.
- Others have **no privacy policies**.
- Many policies state:
  - "We can monitor anything we want."
- Privacy policies may use **vague legal language**.
- Some ISPs provide **no privacy policy at all**.
- Legal protections are limited; ISPs hold significant power.

Picking a Great Password

- Passwords are the **simplest form of authentication**.
- A password is a **shared secret** between user and computer.
- Strong passwords are a **first line of privacy defense**.
- Risks:
  - Easy-to-guess passwords
  - Reusing passwords across services

Why Use Passwords?

- Early personal computers:
  - Used by one person
  - Rarely required passwords
- Internet use introduced:
  - Email account passwords
  - Website account passwords
- Passwords prevent:
  - Unauthorized access
  - Viewing personal data
- Modern operating systems:
  - Windows
  - Macintosh
  - Unix
- Passwords control limited access to personal data.
- Users often receive **poor guidance** on password selection.

Bad Passwords: Open Doors

- Bad passwords are **easy to guess**.
- Crackers use:
  - Automated password-guessing programs
  - Lists of common passwords
- Weak passwords include:
  - Names (self, family, pets)
  - Dictionary words
  - Short passwords
  - Common substitutions (l → 1, E → 3)

    - Words backwards
- Examples of weak choices:
    - Movie characters
    - Landmarks
    - Phone numbers
    - Famous computer scientists
- Web-based services are **more vulnerable** due to:
    - High-speed guessing attempts
- Password rules vary widely across services.

Smoking Joes

- "Joe accounts":
    - Username and password are identical
- Extremely easy for attackers to exploit.
- Crackers often check for Joe accounts first.
- Making username lists public increases risk.

Good Passwords: Locked Doors

- Strong passwords:
    - Use uppercase and lowercase letters
    - Include digits and punctuation
    - Are at least 7–8 characters long
    - Are easy to remember
    - Can be typed quickly
- Suggested techniques:
    - Combine words with symbols (robot4my)
    - Use personal acronyms
- Once published, examples become **bad passwords**.

Bad Passwords

- Avoid:
    - Names (yours or others')
    - Birthdates
    - Social Security numbers
    - Usernames
    - Dictionary words
    - Keyboard patterns (qwerty)
    - Single-digit variations
- Eight-character random passwords provide:
    - Billions of combinations
    - Protection against brute-force attacks
- Longer passwords often fail due to:
    - System truncation at 8 characters

Writing Down Passwords

- Written passwords can be stolen.
- However:
  - A complex written password may be safer than a weak memorized one.
- If writing down passwords:
  - Do not label them as passwords
  - Do not include account names
  - Do not attach to computer
  - Disguise or scramble them
- Password-keeping programs (Table 9-1) can help.

Table 9-1. Recommended password keeper programs

Platform	Program	Location
PalmOS	GNU keyring	*http://gnukeyring.sourceforge.net/*
PalmOS	Strip	*http://www.zetetic.net*
Windows	Password Keeper 2000	*http://www.gregorybraun.com/PassKeep.html*
Windows	Password Safe	*http://www.counterpane.com/passsafe.html*
Macintosh 8.x, 9.x	Mac OS Keychain	Built in; see the Keychain Access control panel

Strategies for Managing Multiple Usernames and Passwords

- Reusing passwords increases risk:
  - One breach compromises many accounts
- System administrators may access stored passwords.
- Password restrictions differ by system.

Password Classes

- Divide passwords by security level:
  - Banking
  - Email
  - Low-security sites

Password Bases

- Modify a base password per service.
- Avoid obvious patterns.

Password Rotation

- Change passwords periodically.
- Can become confusing over time.

Password Keepers

- Store passwords securely in encrypted form.
- Built into browsers (Netscape, Internet Explorer).
- Available as:
  - Wallet programs
  - Stand-alone software
- PGP can be used to create a custom password safe.

Sharing Passwords

- Sharing passwords gives others:
  - Access to personal data
  - Ability to impersonate you
- Treat passwords like **house keys**.
- Best practices:
  - Share discreetly
  - Never email plaintext passwords
  - Change passwords after sharing ends

Resist Social Engineering Attacks

- Attackers trick users into revealing passwords.
- Common methods:
  - Fake ISP emails
  - Phone calls posing as IT staff
  - Requests to reset passwords
- Attacks succeed due to:
  - Desire to be helpful
  - Lack of security awareness

Beware of Password Sniffers and Stealers

Password Sniffers

- Capture unencrypted passwords in transit.
- Target protocols:
  - FTP
  - HTTP
  - POP
  - TELNET
  - RLOGIN
- Sniffers have been found on:
  - University networks
  - Corporate systems
  - ISP backbones
- Use **encrypted protocols** to reduce risk.

Keystroke Recorders and Keyboard Sniffers

- Record everything typed.
- Can be:
  - Software-based
  - Hardware-based
- **Figure 9-1**: KeyKatch device
  - Small hardware keystroke recorder
  - Undetectable without physical inspection
- Screen recorders capture display content.

- Programs like **Back Orifice 2000** enable remote spying.

Figure 9-1. The KeyKatch is a small device that attaches between a keyboard and a desktop computer and can record more than two million keystrokes (reprinted with permission)

Beware of Public Terminals

- Higher risk of spyware.
- Precautions:
  - Avoid confidential access
  - Use temporary webmail accounts

Cleaning Up After Yourself

- Internet use leaves **electronic footprints**.
- Computer forensics can reveal:
  - Visited websites
  - Downloaded data
- Cleanup reduces privacy risk.
- Automatic cleanup tools discussed in Chapter 10.

Browser Cache

- Cache stores previously visited pages.
- Improves speed but reduces privacy.
- Privacy protection methods:
  - Disable caching for SSL pages
  - Disable caching entirely
  - Manually delete cache

Managing Your Cache with Internet Explorer

- Use Internet Properties panel.
- Options:
  - Delete Files
  - View cached files
- **Figure 9-2**: Internet Properties panel
- **Figure 9-3**: Cache directory view
- **Figure 9-4**: ActiveX objects list

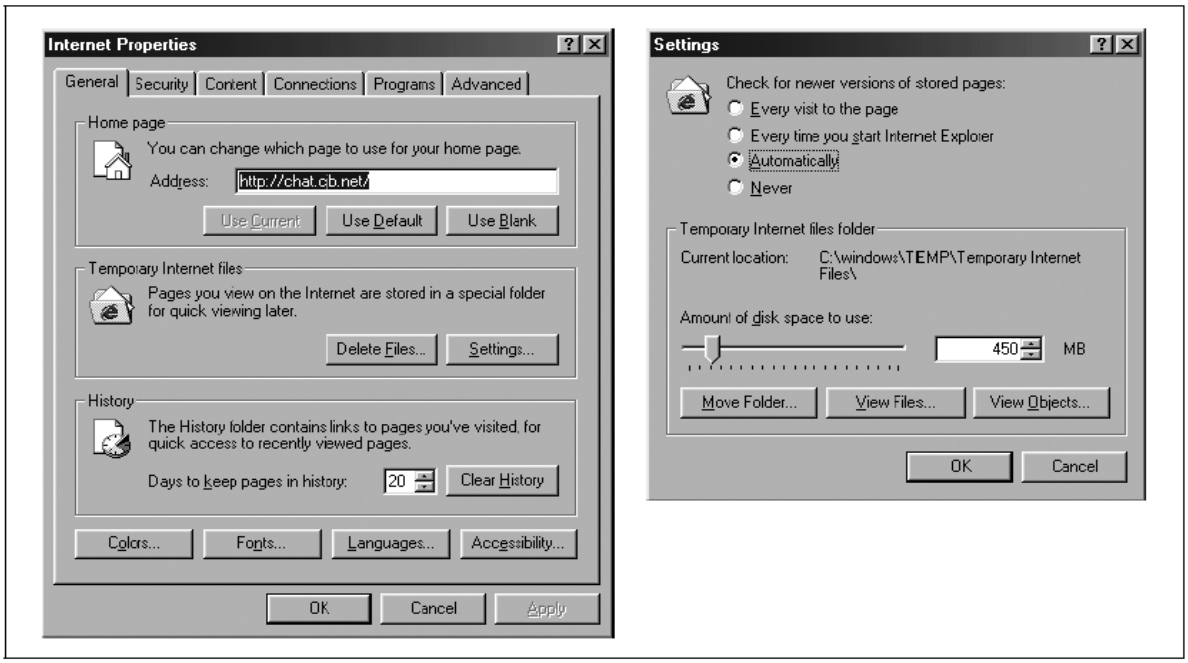

Figure 9-2. With the Internet Explorer "Internet Properties" control panel, you can control the browser's cache by clicking on the "Delete Files . . . " and "Settings . . . " buttons. If you click the "Settings . . . " button, the Settings panel will appear. Click the "View Files . . . " button to display the directory containing cookies and browser cache.

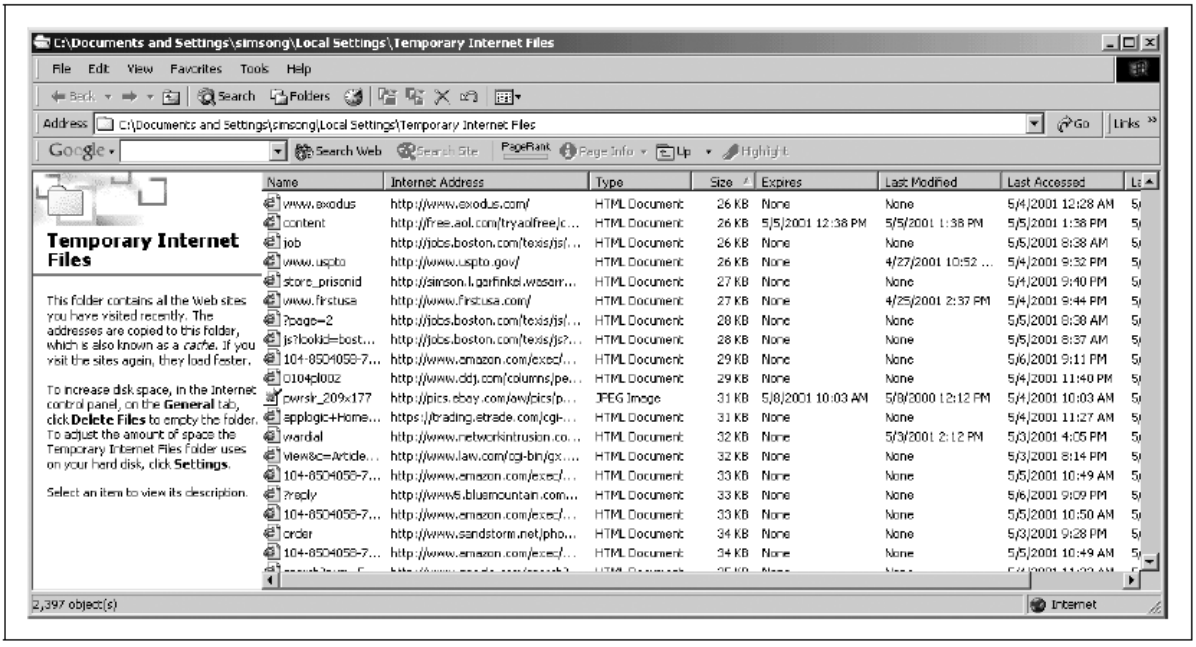

Figure 9-3. When you click on the View Files button, Internet Explorer opens up the Temporary Internet Files folder. This folder can contain cookies, JPEG files, and HTML documents. You can delete them as you wish without damaging your computer.

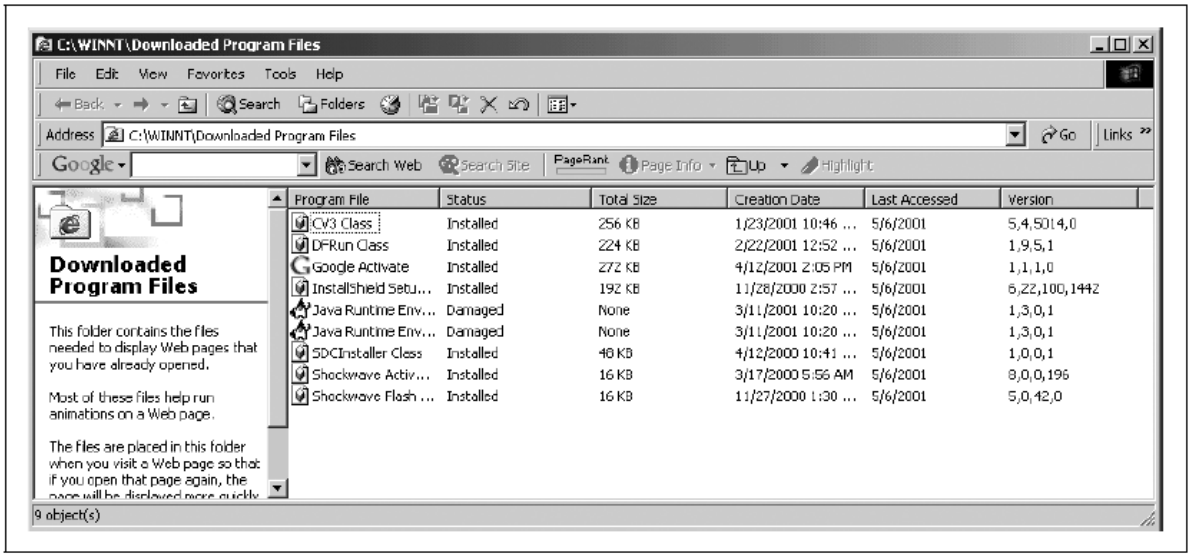

Figure 9-4. When you click the "View Objects..." button, Internet Explorer opens up the Downloaded Program Files folder. This folder will show you the ActiveX components that have been downloaded. In this example, components for several third-party programs have been downloaded. All except the Java runtime components are active.

Managing Your Cache with Netscape Navigator

- Preferences → Advanced → Cache
- Clear disk and memory cache
- **Figure 9-5**: Netscape cache settings

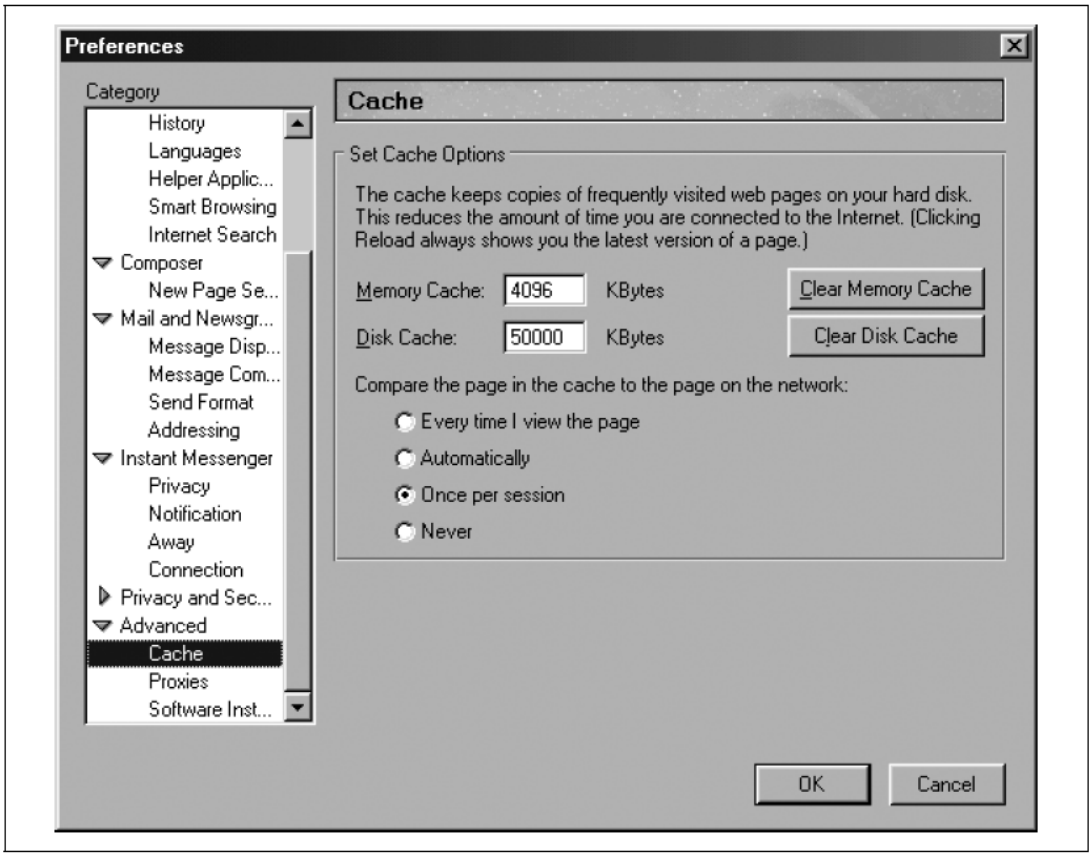

Figure 9-5. Buttons on Netscape Preferences panel allow you to clear the cache

Cookies

- Cookies store session and tracking data.
- Can expose personal information.

Crushing Internet Explorer's Cookies

- Cookies stored in History directory.
- Can be manually deleted.

Crushing Netscape's Cookies

- Stored in cookies.txt.
- Managed via Cookie Manager.
- **Figure 9-6**: Cookie Manager menu
- **Figure 9-7**: Stored cookies view

Figure 9-6. Netscape's Cookie Manager is accessed through the "Privacy and Security" submenu of the Tasks menu.

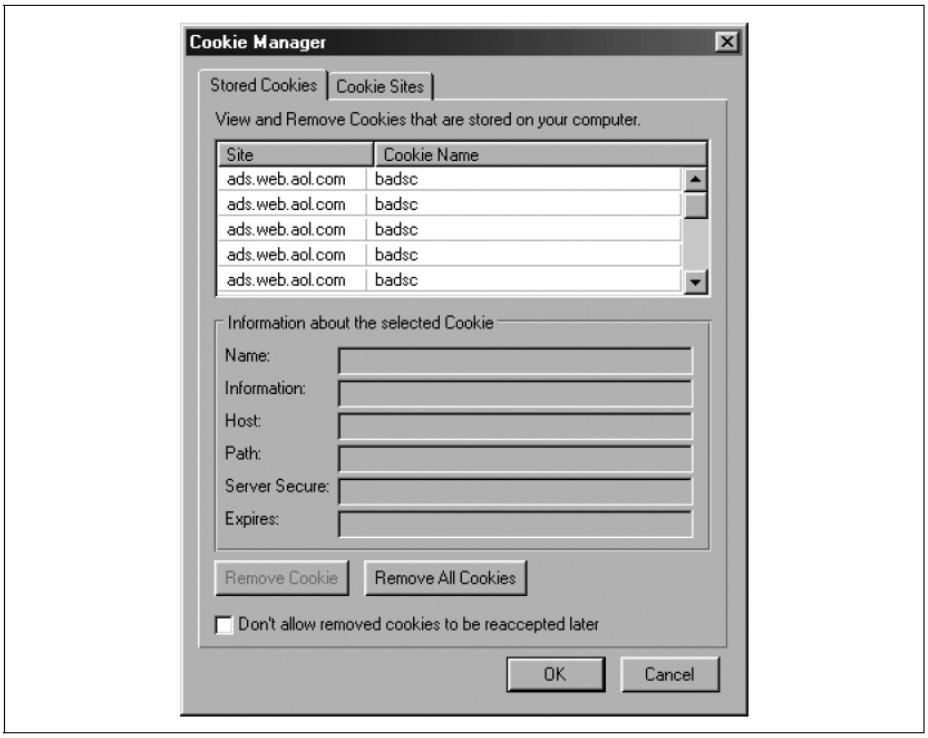

Figure 9-7. The Netscape Cookie Manager shows you which cookies have been accepted from which site. It allows you to block individual cookies, to delete a cookie, or to remove all cookies.

Browser History

- Browsers store visited URLs.
- Can reveal sensitive activity.
- Clearing history is recommended.

Clearing Internet Explorer's Browser History

- Stored in index.dat.
- Viewed as databases in Explorer.
- **Figure 9-8**: History database view
- **Figure 9-9**: Typed URLs registry

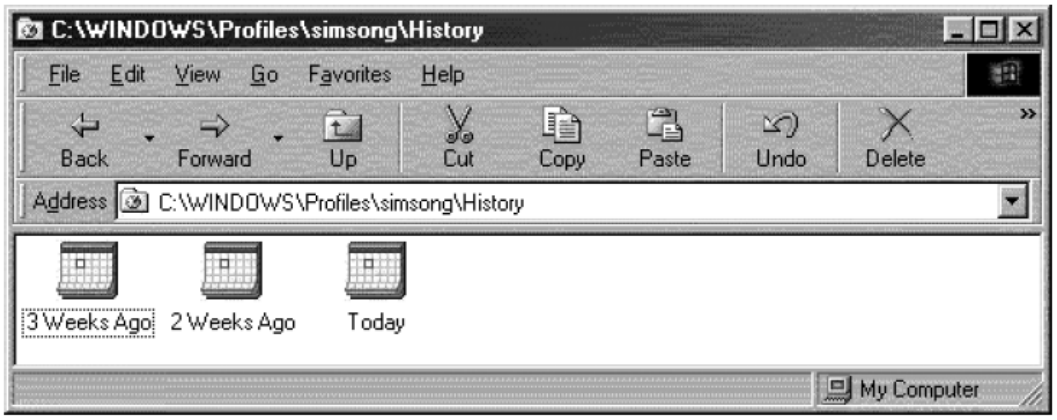

Figure 9-8. Internet Explorer has a shell extension that makes the file index.dat in the History folder appear as a set of tiny calendars. If you double-click on one of the little calendar icons, you will see the individual history records that it contains.

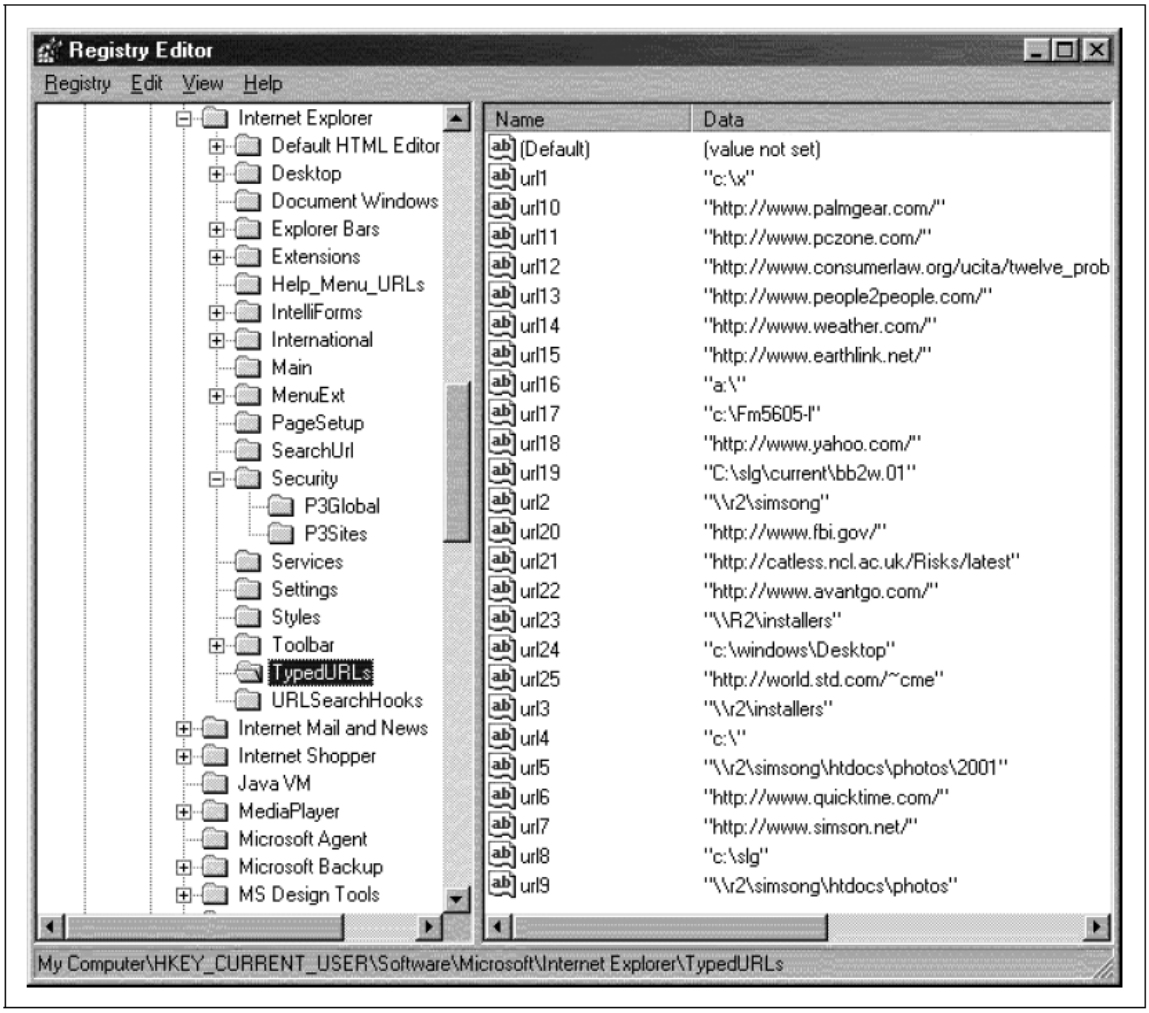

Figure 9-9. Internet Explorer stores the last typed URLs at the Registry Key HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\TypedURLs.

Clearing Netscape Navigator's Browser History

- Preferences → Navigator → History

Passwords, Form-Filling, and AutoComplete Settings

- Browsers store:
  - Form data
  - Usernames
  - Passwords
- Convenient but risky.
- **Figure 9-10**: AutoComplete prompt

Figure 9-10. Internet Explorer's AutoComplete system will remember fields that you recently entered into web forms. This feature can be very handy, but it can also reveal information to other people who have access to your computer.

Clearing AutoComplete with Internet Explorer

- Content tab → AutoComplete
- Clear forms and passwords
- **Figure 9-11**: AutoComplete settings

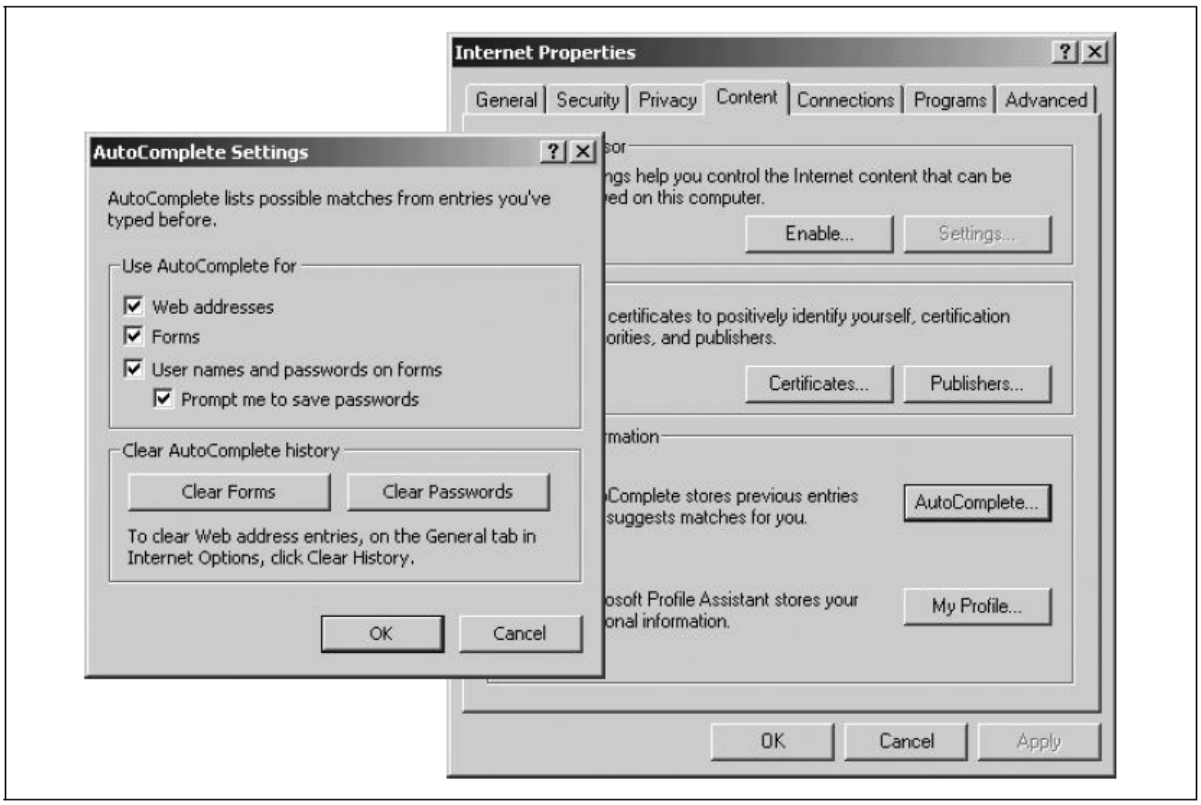

Figure 9-11. Internet Explorer's AutoComplete Settings panel allows you to control where AutoComplete is used. You can also clear AutoComplete information for forms and/or passwords.

Clearing Sensitive Information with Netscape Navigator

- Password Manager
- **Figure 9-12**: Clear sensitive information

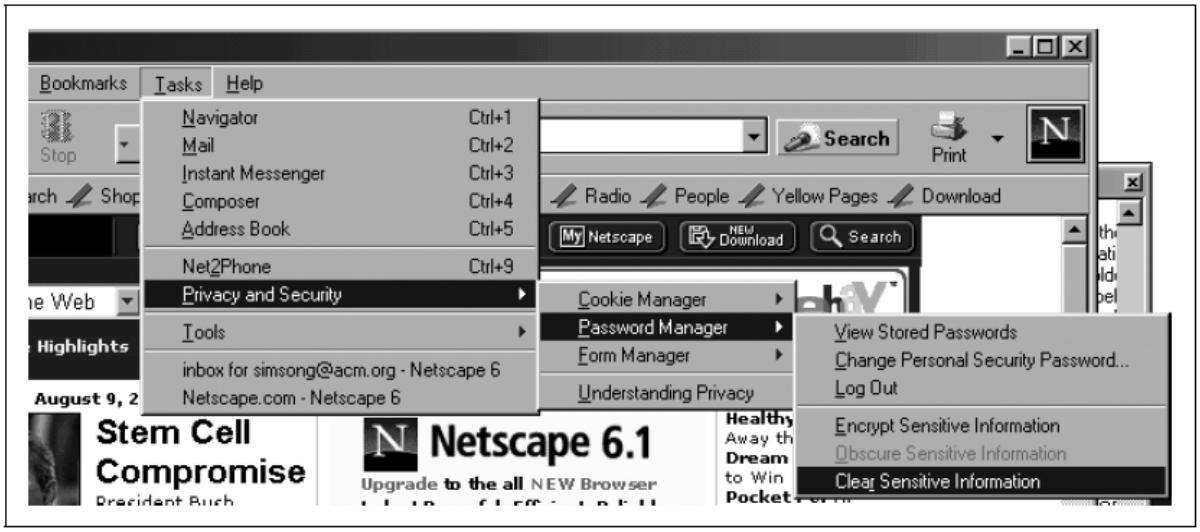

Figure 9-12. Netscape's Password Manager has an option that allows you to clear sensitive information that is stored on your computer.

Avoiding Spam and Junk Email

- Spam is a major privacy concern.

- Causes:
    - Time loss
    - Privacy invasion
    - ISP customer loss

Protect Your Email Address

- Avoid publishing email addresses.
- Remove listings from online directories.
- Avoid posting to:
    - Mailing lists
    - Usenet
- Choose uncommon usernames.

Use Address Munging

- Modify email addresses to confuse spam bots.
- Prefer mangling domain names.

Use an Antispam Service or Software

- Antispam services:
    - Filter messages externally
    - Use whitelists
- Examples:
    - BrightMail
    - SpamCop
- Antispam software:
    - Runs locally
    - Requires maintenance
- Examples:
    - SpammerSlammer
    - Spam Exterminator

Identity Theft

- Identity theft involves misuse of personal data.
- Example: **Stephen Shaw case**
- Consequences:
    - Financial loss
    - Credit damage
    - Emotional distress
    - Employment difficulties
- Often takes **years** to resolve.
- Common methods:
    - Stolen credit reports
    - Trash rummaging
    - Phishing scams

Protecting Yourself From Identity Theft

- Identity theft thrives due to weak identity verification.
- Consumers can take preventive steps.

Shred Your Trash

- Use strip or cross-cut shredders.
- Shred documents with personal data.

Monitor Your Credit Report

- Regularly check reports from:
  - Equifax
  - Experian
  - TransUnion
- Consider monitoring services.

Be Careful of Wallet Contents

- Do not carry:
  - Social Security card
  - Birth certificate
- Photocopy cards for records.

Cancel Unnecessary Credit Cards

- Reduce exposure.

Avoid Using SSNs as Account Numbers

- Request alternate identifiers.

Separate Online and Offline Credit Cards

- Use virtual card numbers when possible.

Don't Give Personal Information to Callers

- Verify identity of callers.

Use Passwords on Accounts

- Replace "mother's maiden name" with passwords.

If You Are the Victim of Identity Theft

- Report to:
  - Police
  - Secret Service
  - Postal Inspector
- Contact:
  - FTC Identity Theft Hotline
  - Banks and credit card companies
- Obtain and dispute all credit reports.
- Consider legal assistance.

3. Privacy-Protecting Technologies

- This chapter introduces **privacy-protecting technologies** used while browsing the Web.

- These technologies help **safeguard personal information** from advertisers, trackers, and unauthorized access.
- The chapter is organized by **program categories**, including:
    - Blocking ads
    - Crushing cookies
    - Anonymous browsing
    - Secure email
- For each category:
    - The basic concept is explained
    - Examples of programs are given
    - One or two programs are demonstrated
- Because the Internet changes rapidly:
    - Some programs may no longer exist
    - New programs may not be listed
- The chapter should be treated as a **survey of tools**, not a buyer's guide.

Blocking Ads and Crushing Cookies

- Web browsers are increasingly designed to **deliver advertisements**.
- Major browser companies (Microsoft, Netscape, Opera) profit from advertising.
- **Figure 10-1**: Shows browsers functioning as ad-delivery platforms.
- Unlike traditional advertising:
    - Internet ads can track users
    - Personal data can be collected and correlated
- Advertisers can:
    - Track browsing behavior
    - Build detailed user profiles
- The Internet's design also allows users to:
    - Defend against cookies
    - Block advertisements

Figure 10-1. On today's Internet, companies that develop web browsers have also created large web sites that are funded by advertising dollars. Browsers have become, in effect, programs for delivering advertisements to consumers.

Local HTTP Proxies

- Proxy servers relay requests between browsers and web servers.
- Commonly used in **corporate firewalls**.
- A **local HTTP proxy**:
  - Runs on the user's own computer
  - Proxies HTTP (web) traffic
- **Figure 10-2**: Shows a local HTTP proxy between the user and the Web.
- Because of its position, a local proxy can:
  - Monitor browsing activity
  - Preview web pages
  - Modify web content

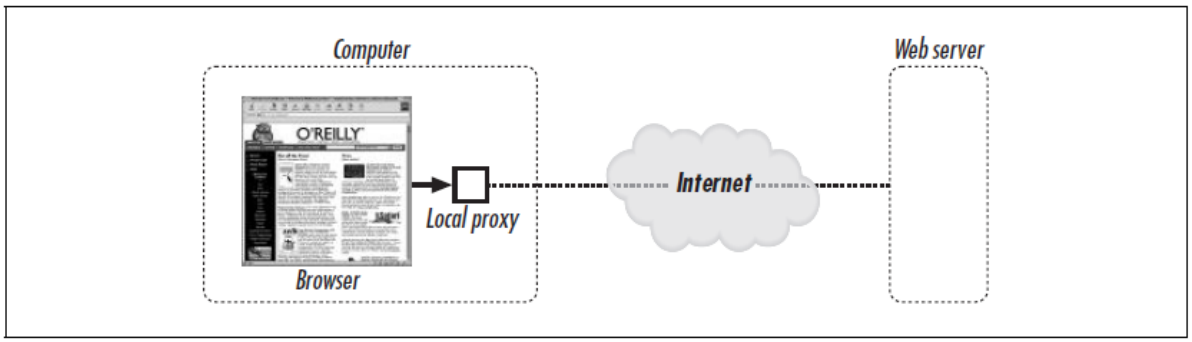

Figure 10-2. A local HTTP proxy sits between the browser on your computer and the rest of the Internet. Because of its position, a local proxy can change the way that you view the Web.

Capabilities of a Local HTTP Proxy

- Record all visited web pages
- Block or allow access to specific sites
- Insert or remove cookies
- Edit HTML content
- Modify downloaded images
- If controlled by the user:
  - Becomes a powerful privacy tool
- If controlled by others:
  - Can be invasive and dangerous

Using Ad Blockers

- Ad blockers often use **local HTTP proxies**.
- Two main ad-blocking methods:
  - Blocking ad-related URLs
  - Editing HTML to remove ads
- **Figures 10-3, 10-4, 10-5**: Demonstrate ad blocking and HTML modification.

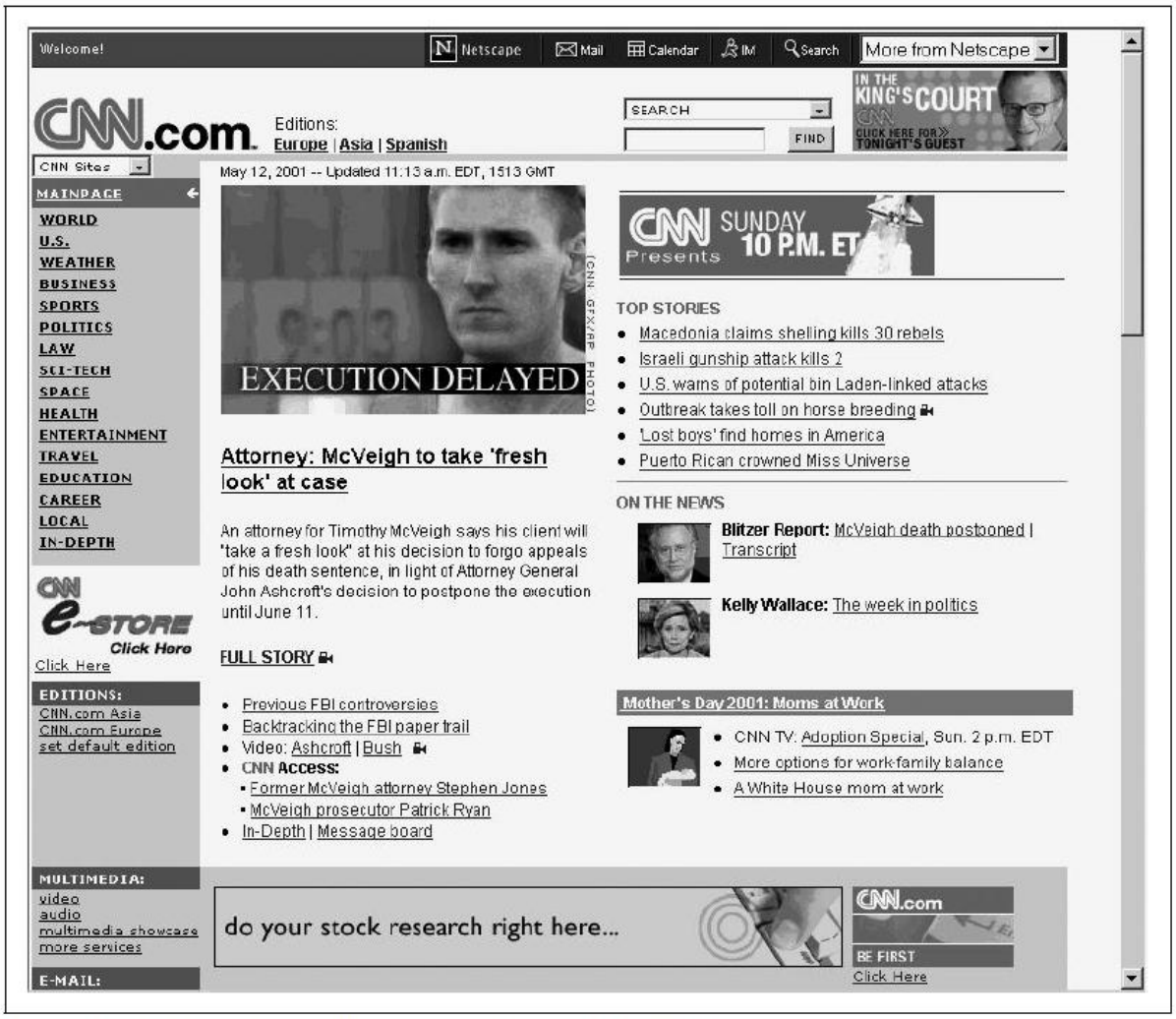

Figure 10-3. The CCN.com home page on May 12, 2001. This page contains four advertisements.

Figure 10-4. The CNN.com home page, as viewed through the Junkbuster HTTP proxy. Junkbuster recognizes the URLs of many advertisements and blocks them from being downloaded. The web browser displays these blocked images as boxes with an "X."

Figure 10-5. The CNN.com home page, as viewed through the AdSubtract HTTP proxy. AdSubtract edits the HTML as it travels from the web site to the browser and automatically removes the tags that cause some advertisements to be displayed. Notice how the page is automatically reformatted so that more content appears on the screen.

Reasons to Block Advertisements

- Ads are distracting
- Ads waste screen space
- Ads slow down page loading
- Ads often contain tracking technologies (cookies)

Disadvantages of Blocking Ads

- Some ads contain useful information
- Many websites rely on ad revenue
- Blocking ads may harm free content availability

Crushing Cookies

- Many ad blockers also **crush cookies**.
- Selective cookie blocking allows:

    - Blocking cookies from advertisers
    - Allowing cookies from trusted sites (banks, brokers)
- This balances **privacy and functionality**.

Additional Features of Ad Blockers

- Remove background music and images
- Disable JavaScript, Java, and ActiveX
- Stop animated GIFs
- Block pop-ups
- Prevent browser history manipulation
- Disable auto-refresh pages
- Block refer links
- Ad blockers can block **dozens of ads** after only a few pages.
- **Table 10-1**: Lists available ad-blocking programs.
- Most ad blockers:
    - Run on Windows
    - Some can protect entire local networks

Table 10-1. A survey of ad blocking programs

Program	Features	Comments
AdSubtract[a] *http://www.adsubtract.com*	Ad blocking Cookie management Sophisticated filtering	Windows only. Several different versions (some free) with different features available.
Internet Junkbuster Proxy *http://www.junkbuster.com*	Ad blocking Cookie management	Free. Windows and a variety of Unix systems.
Freedom Internet Privacy Suite *http://www.freedom.net*	Ad blocking Cookie management Form-filling Many other privacy features	Free. Windows and Linux. Optional subscription service provides for anonymous web browsing and email.
Norton Internet Security *http://www.symantec.com/sabu/nis/nis_pe/*	Ad blocking Cookie management Personal firewall Many other security features	Windows only.
WebWasher *http://www.webwasher.com*	Ad blocking Cookie management Sophisticated filtering	Windows only.

[a] Simson Garfinkel is on the advisory board of AdSubtract's publisher, InterMute.

Anonymous Browsing

- HTTP proxies **cannot hide IP addresses**.
- IP addresses:
    - Contain personal information
    - Allow tracking across websites
- IPs can be used to identify users via logs and legal orders.

Examples of IP Address Tracking

- **MIT Media Lab hostname**:
  - Linked directly to a single user
- **Media One cable modem hostname**:
  - Linked via ISP records
- **WebTV proxy server**:
  - Shared among many users
  - Logs still exist
- **Dial-up server hostnames**:
  - Reassigned over time
  - Usage records retained

IP Addresses in Email

- Web-based email often includes IP addresses in headers
- Example shown using **Hotmail headers**
- IP leakage may reveal location and identity

Simple Approaches to Protecting Your IP Address

Browse from a Public Terminal

- Public libraries and universities offer anonymity
- Institutions are often committed to user privacy

Use America Online

- AOL uses caching proxy servers
- User IP is hidden behind proxy names
- Example proxy hostnames listed
- Privacy depends on AOL policies and legal pressure

Use Your ISP's Web Cache or Proxy Server

- ISP proxies mask end-user IPs
- Remote servers see the proxy IP instead of the user's

Anonymous Web Browsing Services

- Provide stronger anonymity than simple methods
- Operate as **proxy services**
- **Figure 10-6**: Shows anonymous proxy architecture
- Key feature:
  - No log files kept
- Cannot comply with court orders for user activity

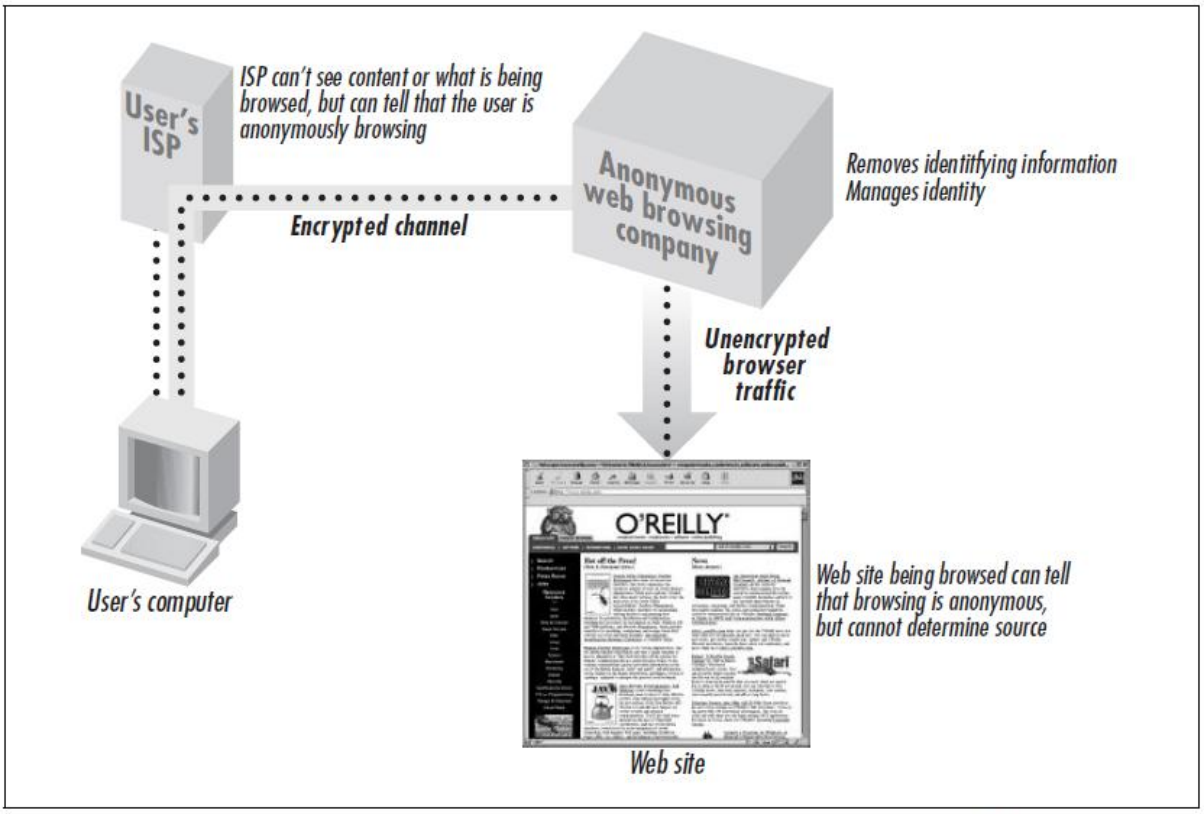

Figure 10-6. An anonymous web browsing service acts like a proxy server or a cache, except that no records are kept by the operator

Anonymizer.com

- One of the first anonymous browsing services
- Requires:
  - No software installation
  - No browser reconfiguration
- Users enter URLs on the Anonymizer website
- URLs are rewritten to maintain anonymity
- Example HTML rewriting shown
- **Figure 10-7**: Web page viewed through Anonymizer
- Offers:
  - Free ad-supported service
  - Paid service (~$5/month in 2001)
  - Secure encrypted tunnel option

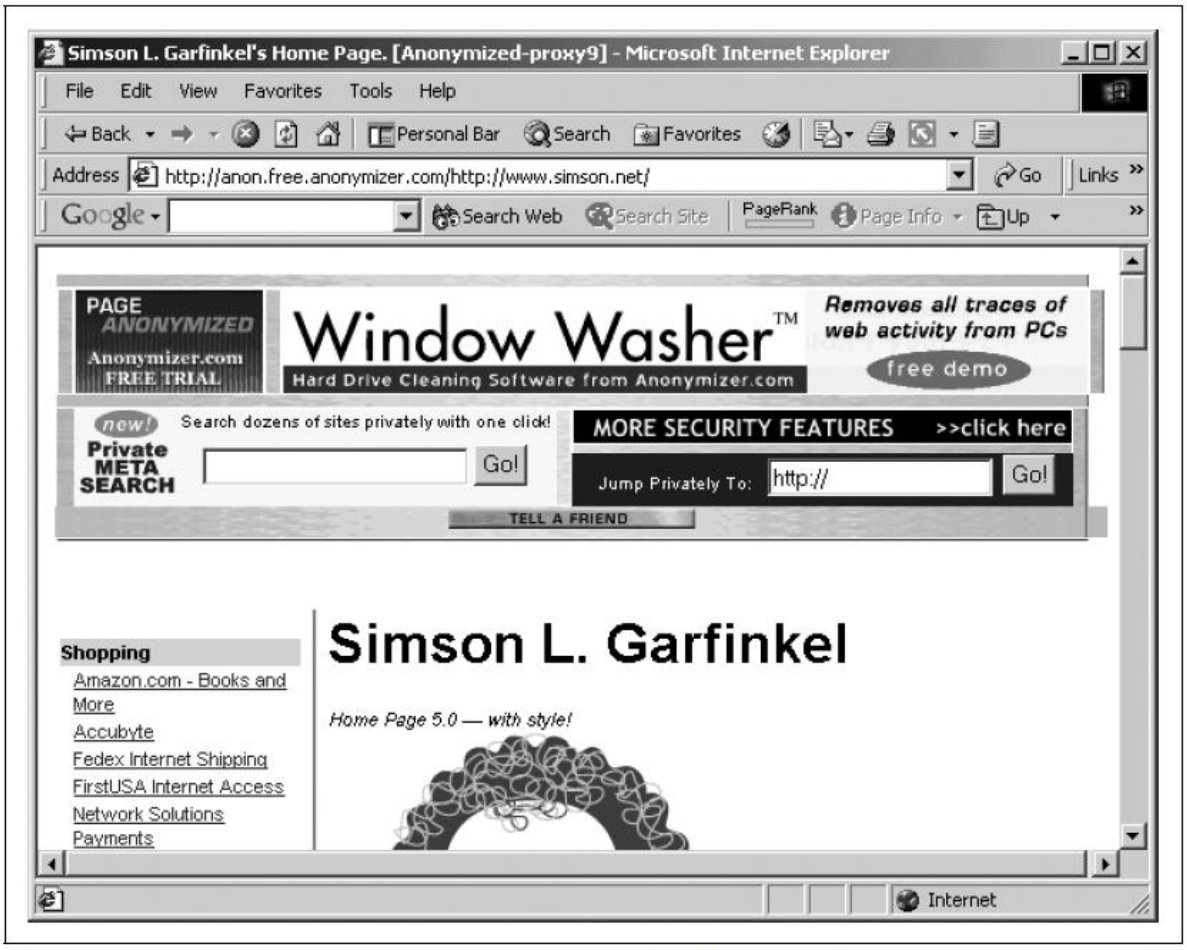

Figure 10-7. The Anonymizer web privacy service uses URL rewriting to provide anonymous web browsing

Freedom, by Zero Knowledge Systems

- Uses **multiple proxy servers** across countries
- **Figure 10-8**: Shows multi-hop encrypted routing
- Each packet:
  - Encrypted in multiple layers
  - Decrypted step-by-step by different servers
- Offers:
  - Anonymous browsing
  - Anonymous chat
  - Untraceable encrypted email
- Supports multiple identities (**nyms**)
- Each nym can:
  - Use separate cookies
  - Block cookies
- Cost: $49.95/year (includes 5 nyms)

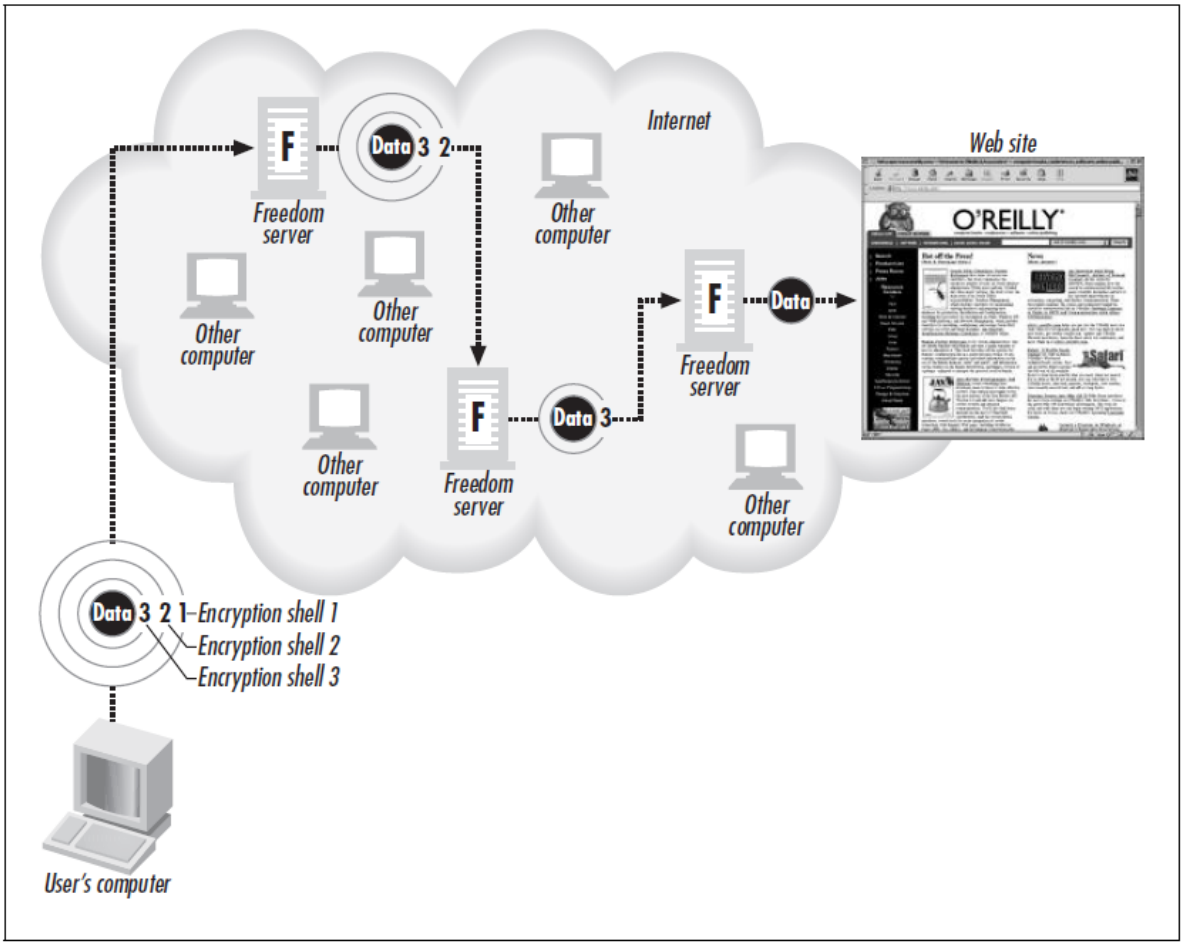

Figure 10-8. Each packet of data sent through the Freedom Network is encrypted with three distinct layers of encryption

safeWeb

- Similar to Anonymizer
- Key features:
  - Free service
  - SSL encryption
  - Customization options
- Supported by non-tracking ads
- **Figure 10-9**: safeWeb interface

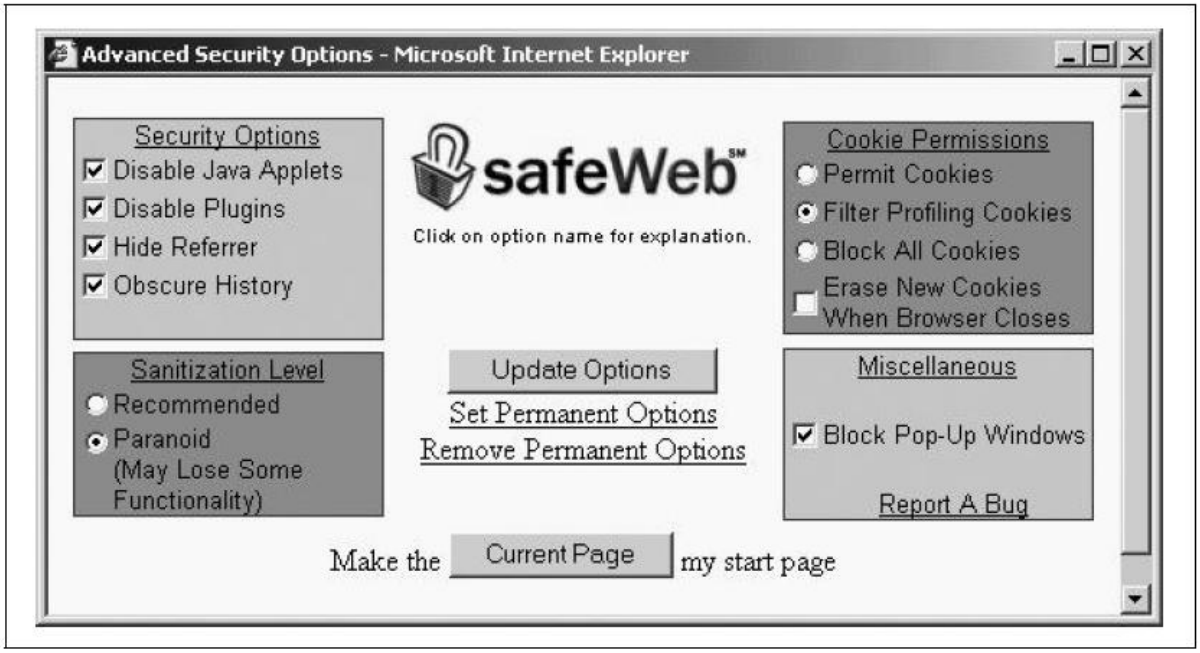

Figure 10-9. safeWeb's customization panel allows you to control how much information about your computer is revealed when you browse "anonymously"

Secure Email

- Email often carries highly sensitive information
- Standard email lacks security
- Threats include:
  - Identity leakage
  - ISP or employer monitoring
  - Misdelivery
  - Unauthorized access
  - Forwarding without consent
  - Message tampering
- These are **real, experienced risks**
- Can be mitigated using proper technologies

Hotmail, Yahoo Mail, and Other Web-Based Email Services

- Provide:
  - Free or low-cost email
  - Access from anywhere
- Useful for:
  - Semi-permanent addresses
  - Anonymous use with anonymous browsing
- Risks include:
  - Provider access to all emails
  - No end-to-end encryption
  - Exposure to subpoenas
  - Interception without SSL

    - Advertisements added to messages
- Useful for disposable or single-purpose email accounts

Hushmail

- Secure web-based email service
- **Figure 10-10**: Hushmail interface
- Encrypts messages so:
    - Even Hushmail staff cannot read them
- Encryption occurs on the user's computer
- **Figure 10-11**: Client-side encryption process
- Uses public-key cryptography
- Private key protected by user passphrase
- If passphrase is forgotten:
    - Account is unrecoverable
- Messages:
    - Automatically encrypted and decrypted
    - Not stored unencrypted on servers or disks
- Offers:
    - Free ad-supported version
    - Premium paid version

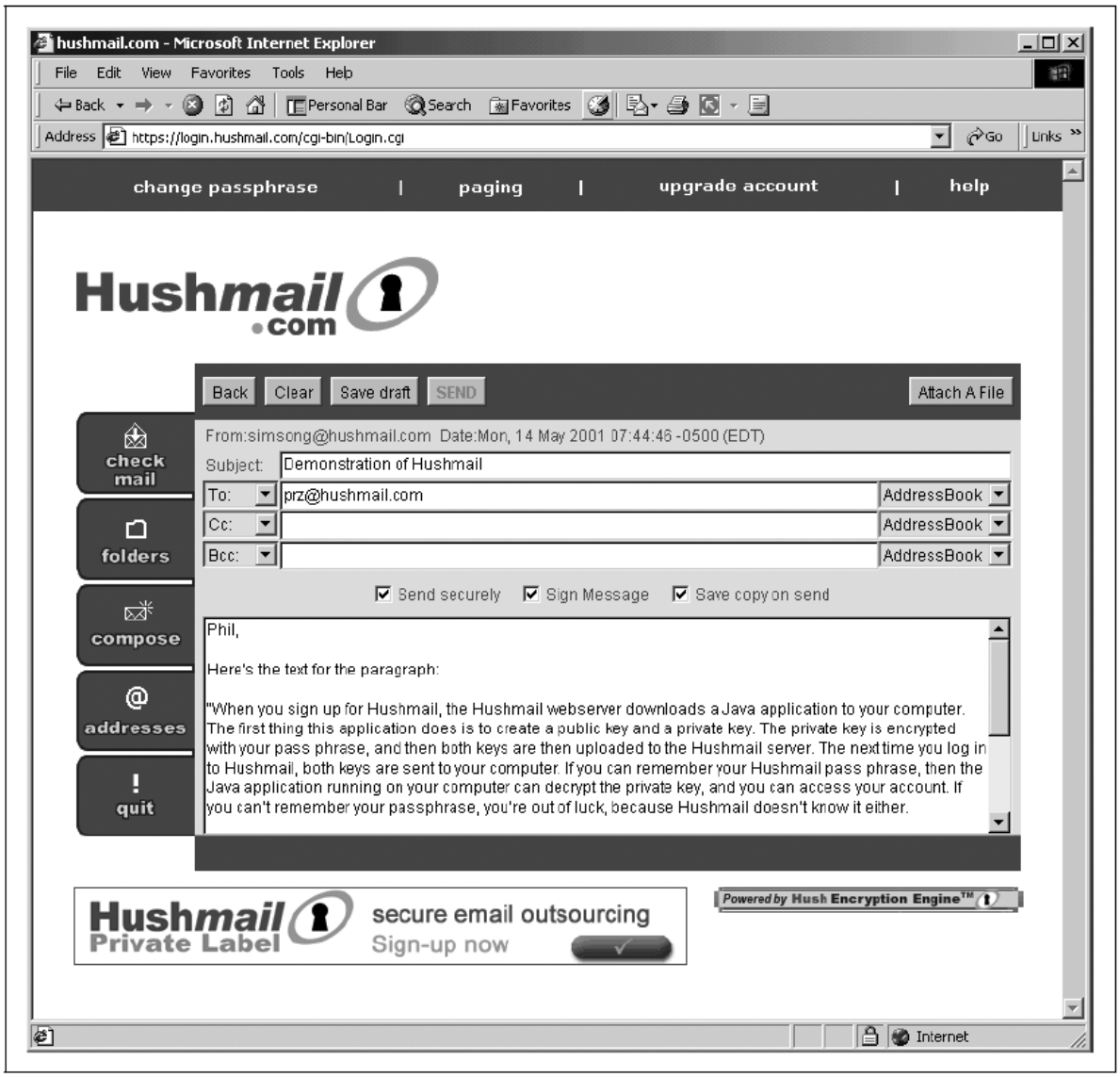

Figure 10-10. Hushmail looks like other web-based mail systems, but it is much more secure because all messages are encrypted and decrypted on the end user machines, rather than the server

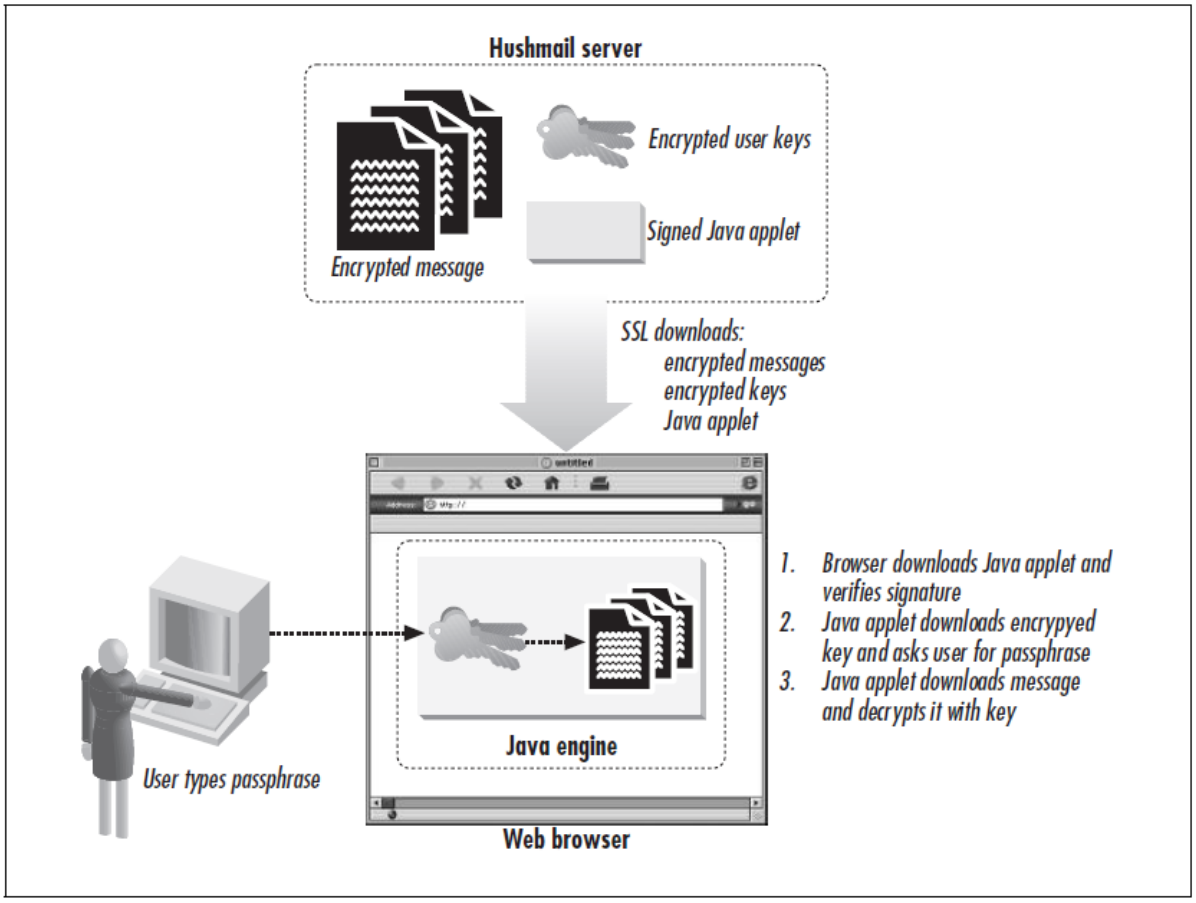

Figure 10-11. Encryption in the Hushmail system happens inside a browser on the end user's computer

Omniva's Self-Destructing Email

- Email copies exist in many locations
- **Figure 10-12**: Shows multiple email copies
- Email archives are valuable in:
  - Investigations
  - Litigation
- Omniva uses **time-limited cryptographic access**
- Sender chooses message expiration date
- **Figure 10-13**: Encryption and key distribution
- Encrypted messages are unreadable without keys
- **Figure 10-14 & 10-15**: Key-based access and expiration
- After expiration:
  - Key is deleted
  - Message becomes unreadable
- Does not prevent:
  - Printing
  - Manual copying
- Improves privacy and reduces long-term exposure

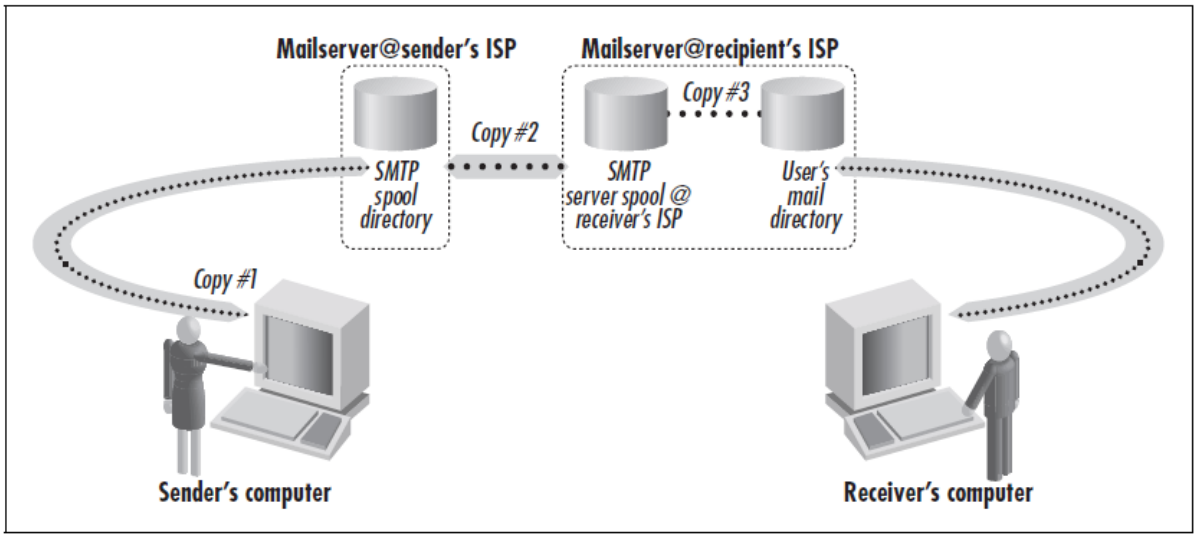

Figure 10-12. The typical email message is copied at least four times—and sometimes many more

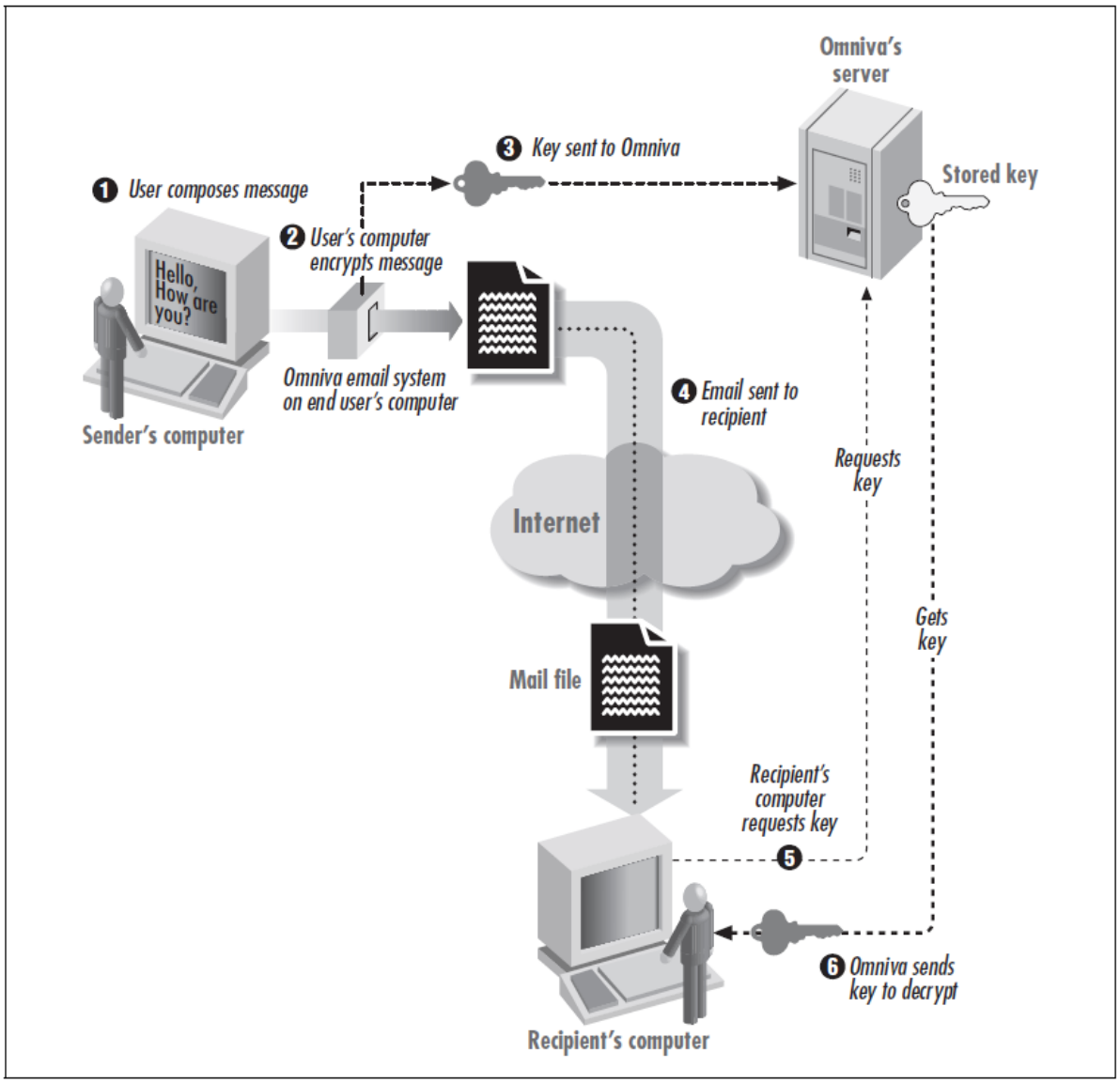

Figure 10-13. The Omniva email system relies on encryption and a central key server to assure that email messages will be unintelligible after their expiration date.

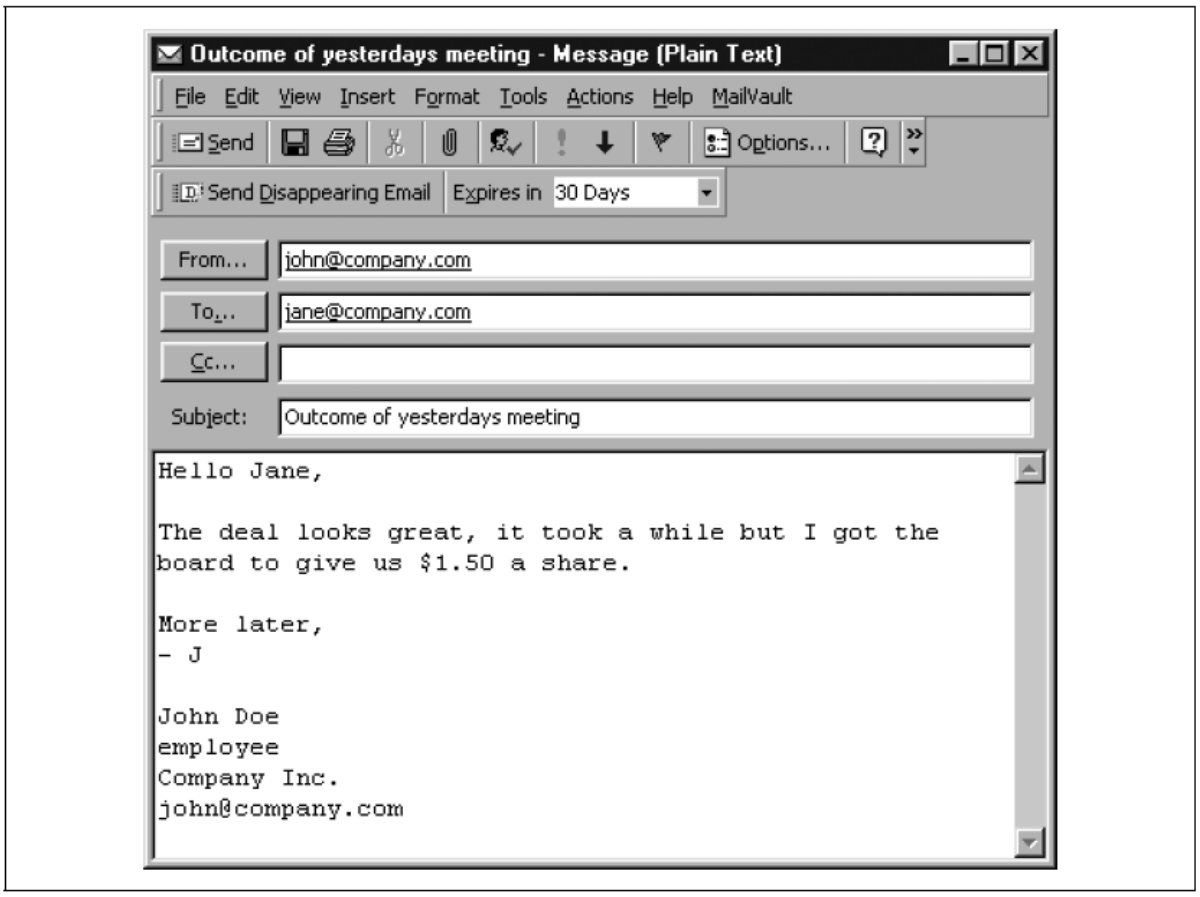

Figure 10-14. A message composed with the Omniva system message is given an expiration date

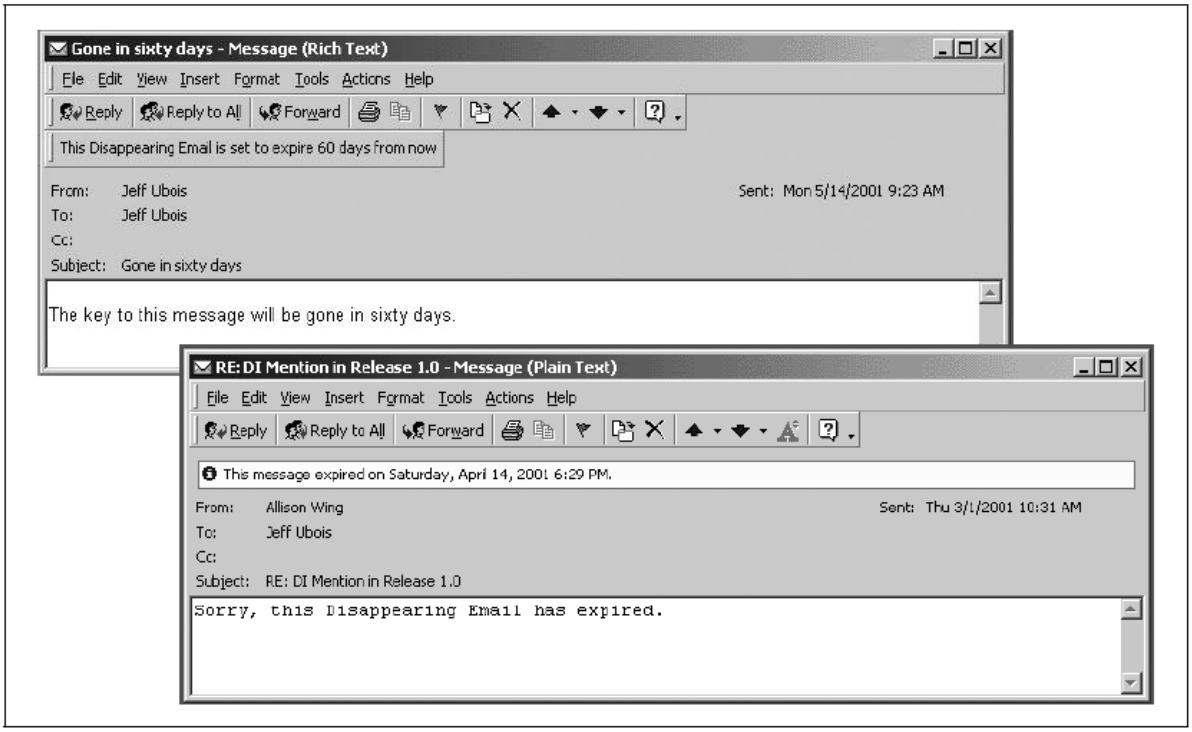

Figure 10-15. Viewing a message on Omniva Email depends on whether the message has expired. This example shows a message that hasn't expired (left) and another message that has (right).

4. Backups and Antitheft

- Focuses on:
  - Data loss
  - Theft
  - System recovery
- Threats include:
  - Hardware failure
  - Theft
  - Disasters
  - Human error

Using Backups to Protect Your Data

- Backups are **copies of important data**
- Can be:
  - Simple (Zip disk)
  - Complex (tape + restore floppy)
- Allow system restoration after loss

Make Backups!

- Failures are unpredictable
- Backups prevent permanent data loss
- Insurance replaces hardware, not data
- Essential for recovery after disasters

Why Make Backups?

Reasons

- Archival records
- User error
- System-staff error
- Hardware failure
- Software corruption
- Electronic break-ins
- Theft
- Natural and human-made disasters

What Should You Back Up?

- Two strategies:
  1. Back up only unique data
  2. Back up everything
- Recommended: **Back up everything**
- Simplifies restoration
- Protects against missing installation media

Types of Backups

Level-Zero Backup

- Initial full system backup

Full Backup

- Copies all files regularly

Incremental Backup

- Copies only changed files
- Common strategy:
  - Full backup biweekly
  - Incremental backup nightly
- **Figure 11-1**: Rotating backup tapes

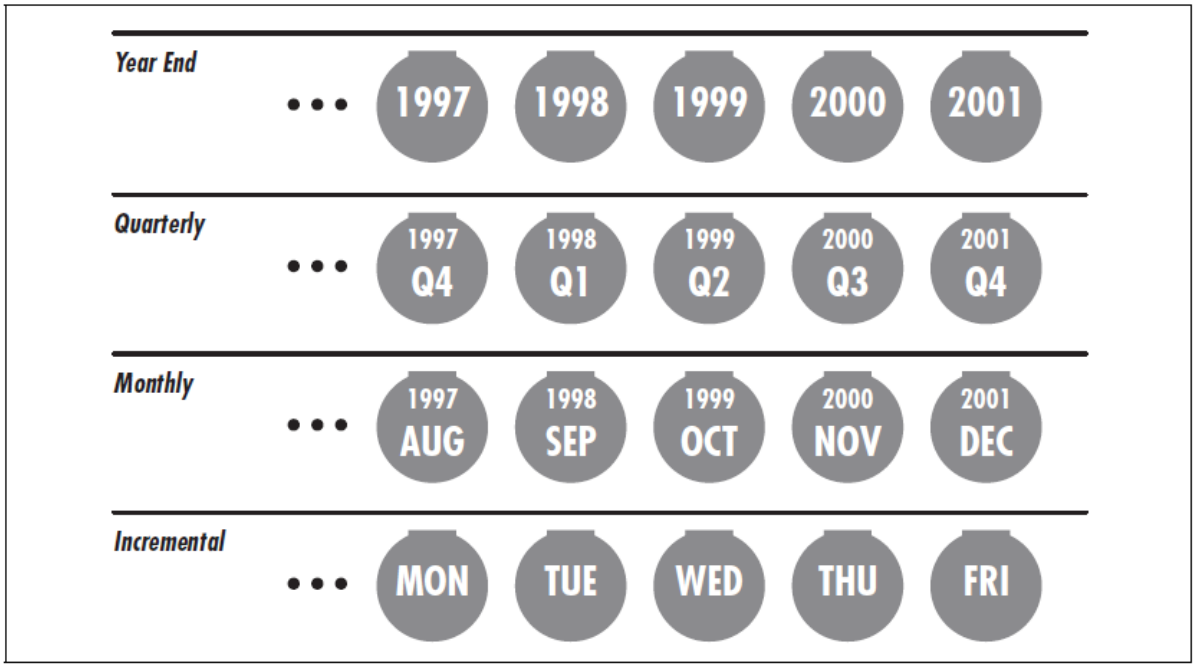

Figure 11-1. An incremental backup

Guarding Against Media Failure

- Use tandem backup sets (A and B)
- Protects against tape failure
- Periodically test restores
- Annual full restoration recommended

Security for Backups

Physical Security

- Remove media from drives
- Store backups off-site
- Use media-safe fireproof storage

Data Security

- Backups contain all data
- Encrypt backups

- Secure encryption keys
- Use escrow or shared key systems

Legal Issues

- Backup tapes may be subpoenaed
- Retention policies should apply to backups
- Segregate sensitive data
- Back up carefully

Preventing Theft

- Theft often occurs due to opportunity
- Prevention reduces risk significantly

Locks

- Laptops include security slots
- **Figure 11-2**: Laptop lock slot
- Cable locks prevent grab-and-run theft
- Vendors: Kensington, Kryptonite

Figure 11-2. Most laptops today are sold with a security slot (reprinted with permission of Kensington)

Tagging

- Equipment tags deter resale
- **Figure 11-3**: STOP theft tag
- Tags:
  - Are serial-numbered
  - Leave permanent marks if removed
- Used by governments and universities

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

Laptop Recovery Software and Services

- Tracing software reports location
- Example: **Computrace**
- Cost-effective
- Works unless disk is reformatted

Awareness

- Simple habits reduce theft:
  - Never leave laptops unattended
  - Secure laptops in hotels
  - Carry laptops personally
  - Avoid window-side placement

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.
- **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.

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.

Five Key Questions:

1. Who has physical access?
2. What if access is abused violently?
3. What if competitors enter unnoticed?
4. What if fire destroys systems?
5. 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

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.
- **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.

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.

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).

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).

Explosion

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

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.

Bugs (biological)

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

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.

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.

Vibration

- **Effects**:
  - Loosened boards.
  - Disk misalignment.
- **Guidelines**:
  - Rubber/foam mats.
  - Avoid placing printers on computers.
  - Laptops are more vibration-resistant.

Humidity

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

Water

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

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.

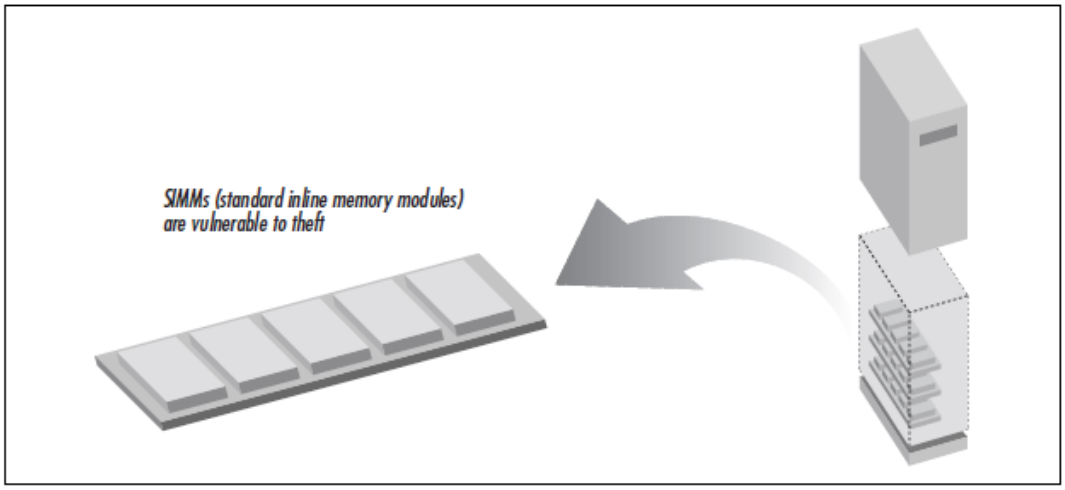

Figure 14-1. There are many recent cases of theft of all or part of computer RAM. RAM is easy to resell and all but untraceable.

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

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

Protecting Your Data

There is a close relationship between **physical security** of computer systems and the **privacy, integrity, and availability of data**. If a computer system is stolen or physically compromised, the data it contains is also at risk. Many attacks on data bypass physical safeguards entirely. This section discusses various **data attacks** and **countermeasures**, reinforcing earlier security concepts from a data-centric perspective.

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.

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.

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

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**)

Although not a concern for most users, radio eavesdropping is easier than expected and should not be ignored.

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
- Physical destruction may still be required

Sanitizing Printed Media

Printed materials often contain sensitive information:

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

Improper disposal enables:

- Social engineering
- Corporate espionage

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**

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**

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**
- Additional references provided in **Appendix E**.

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

(Typically illustrated using attack flow diagrams or classification figures)

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)**

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

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

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

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)

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)

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.

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

(Typically illustrated using post-compromise usage diagrams)

Compromised systems are used for:

- Launching further attacks
- Distributed denial-of-service (DDOS)
- Running covert servers (e.g., IRC rendezvous points)
- 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.

Back Orifice and Netbus

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

bots

- Distributed attack agents.
- Used for:
  - DDOS
  - IRC control

- 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

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

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.

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

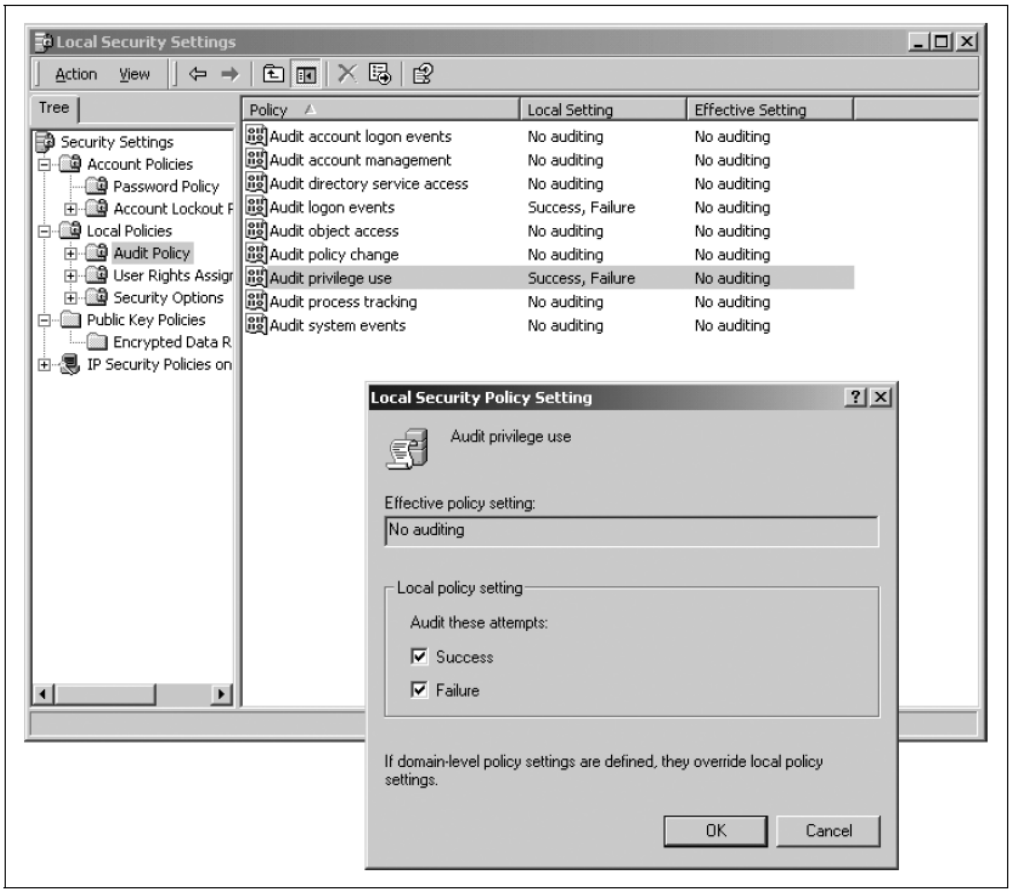

Figure 15-1. Enable auditing from the Local Secure Policy Setting application.

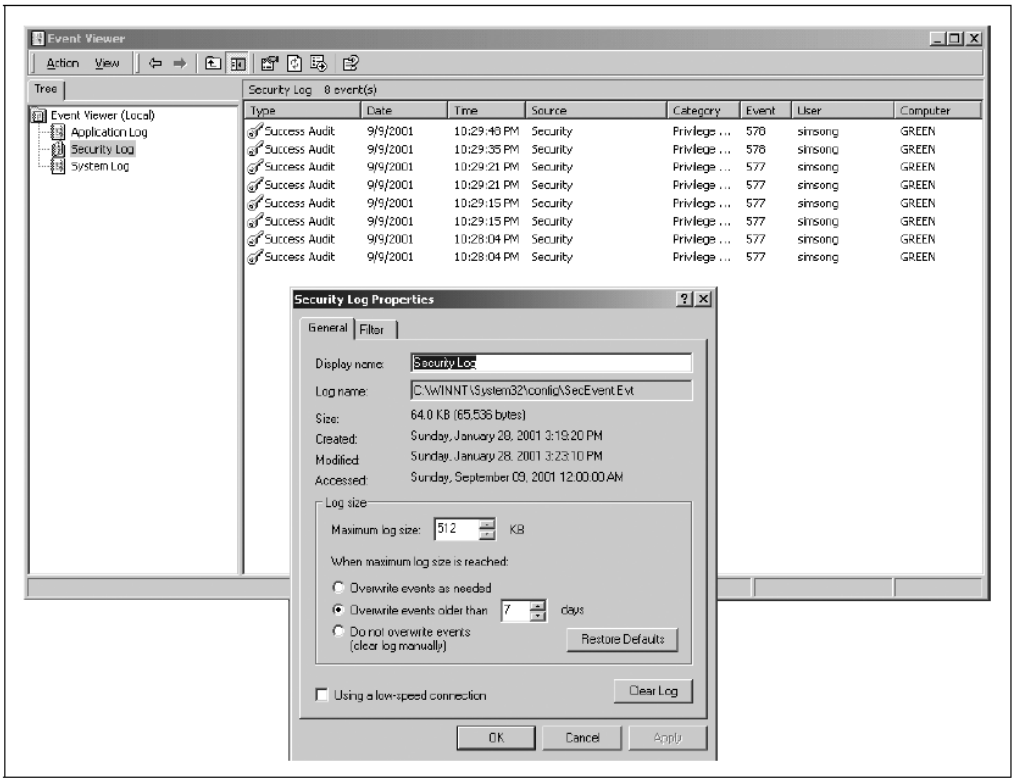

Figure 15-2. Run the Event Viewer application to view the contents of the log.

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:**
  1. Snapshot tools
  2. Change-detecting tools
  3. Network scanners
  4. Intrusion detection systems
  5. 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

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.

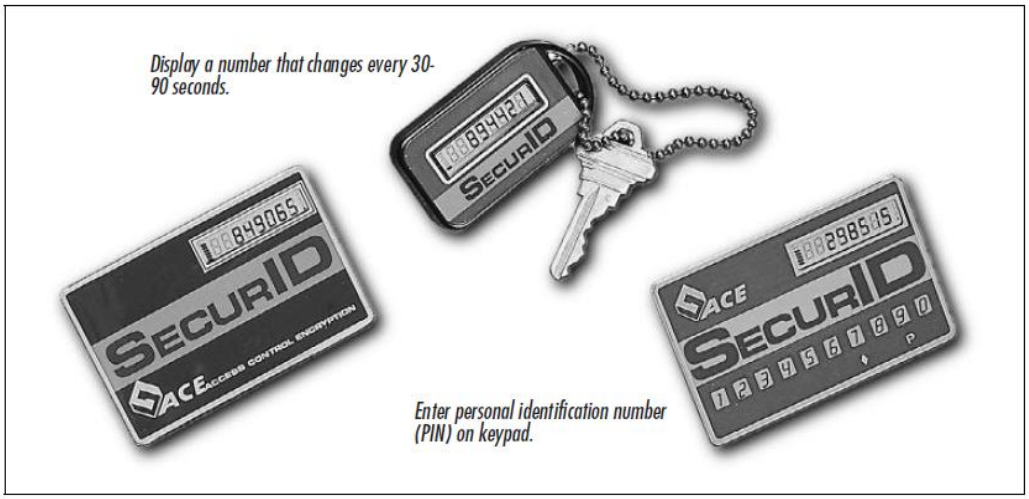

Figure 15-3. Security Dynamics' SecurID card (reprinted with permission)

```
~~xy3221~~      seeme4
~~screen42!~~   421mefoo
~~apss42~~      Jo4!me
~~foobedobe~~
sometix4
Abrac!42
ohmy000
asyucan
```

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.

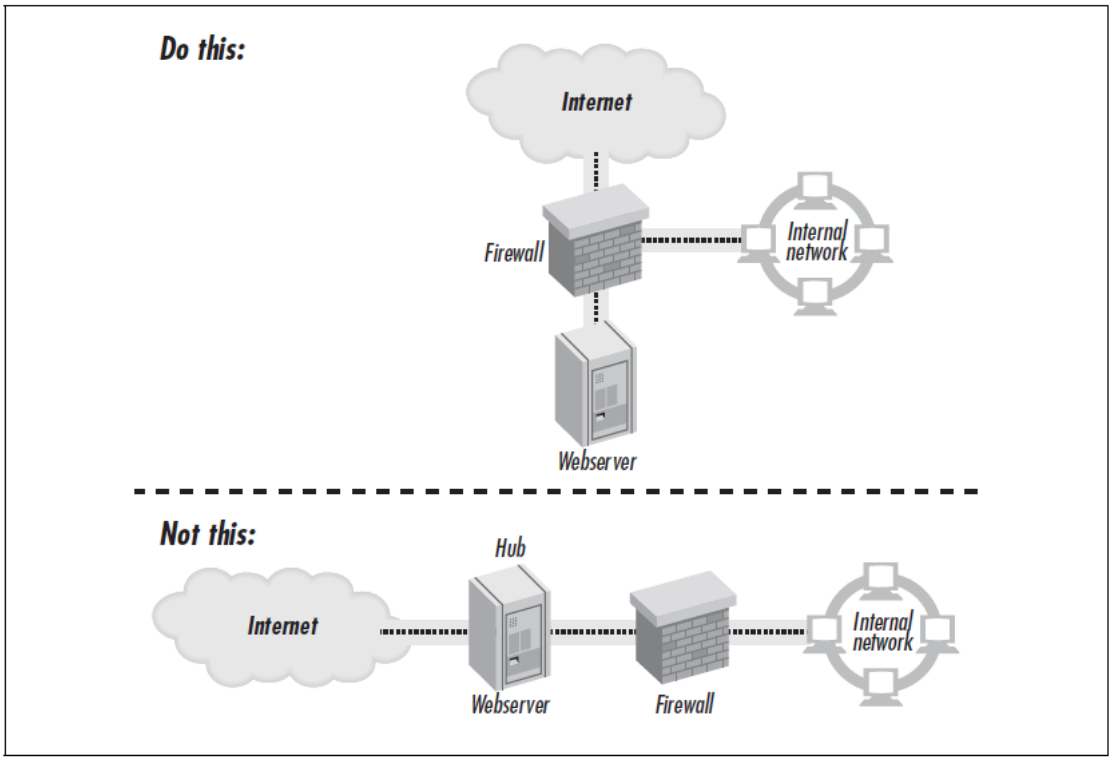

Figure 15-5. For high security, use a firewall to protect your web server from attackers on the Internet. Position the firewall so that it also protects your own organization from the web server.

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.
- **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**.

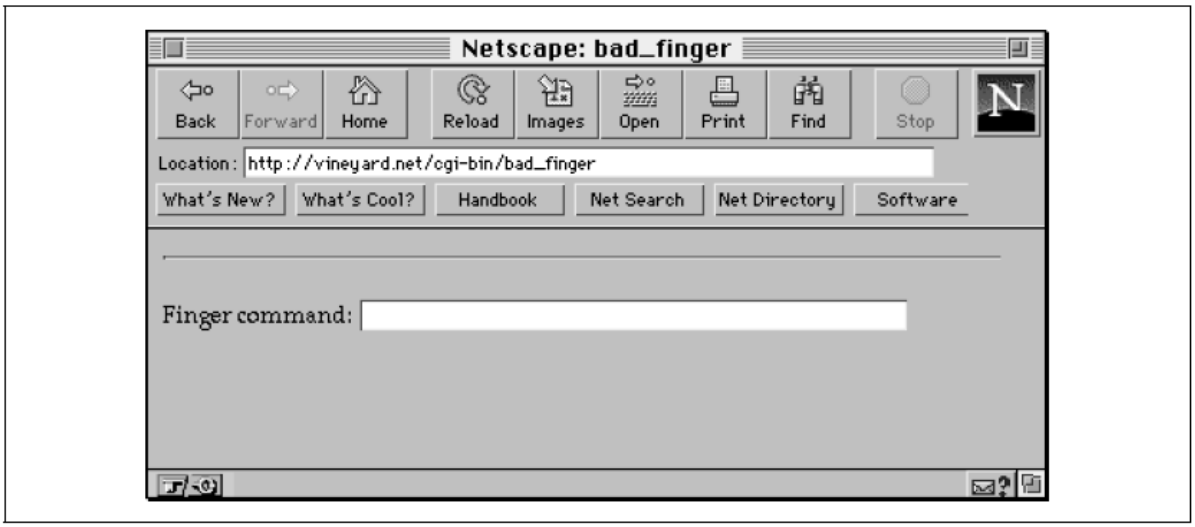

Figure 16-1. The finger gateway

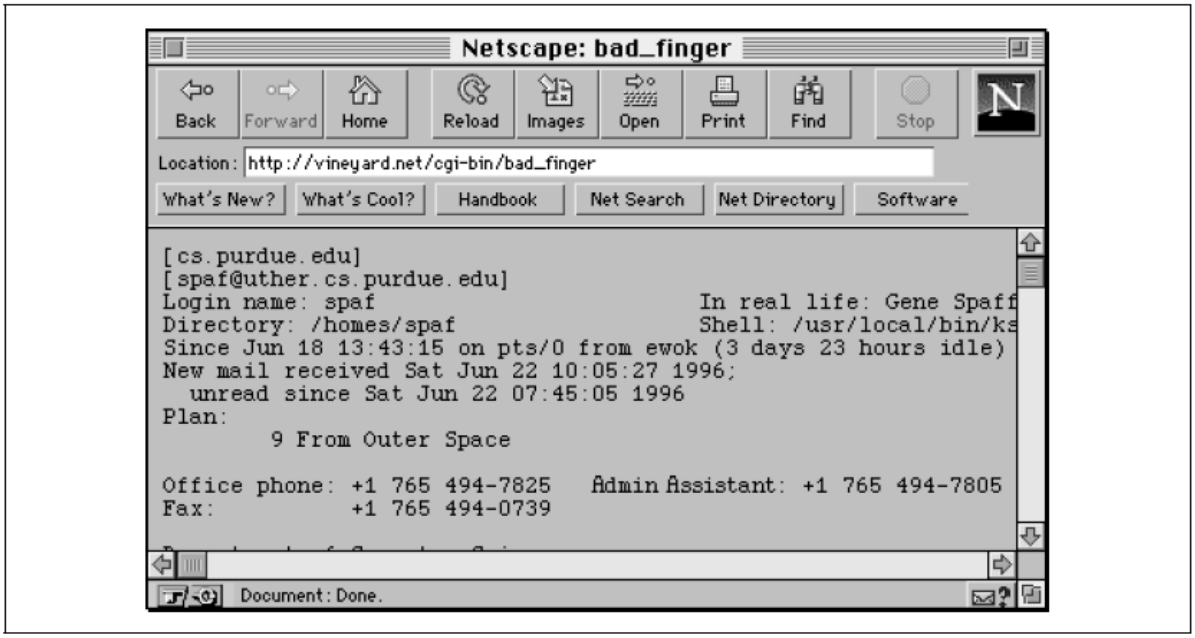

Figure 16-2. The form displayed by the finger script

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.
- **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**.

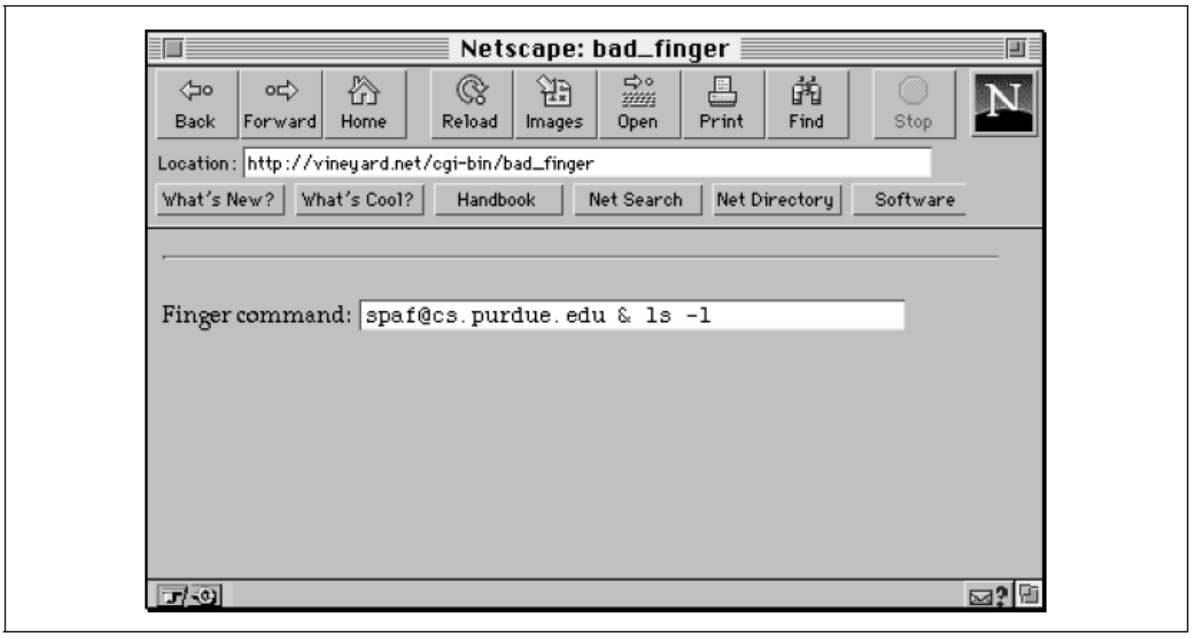

Figure 16-3. Attacking the bad_finger script

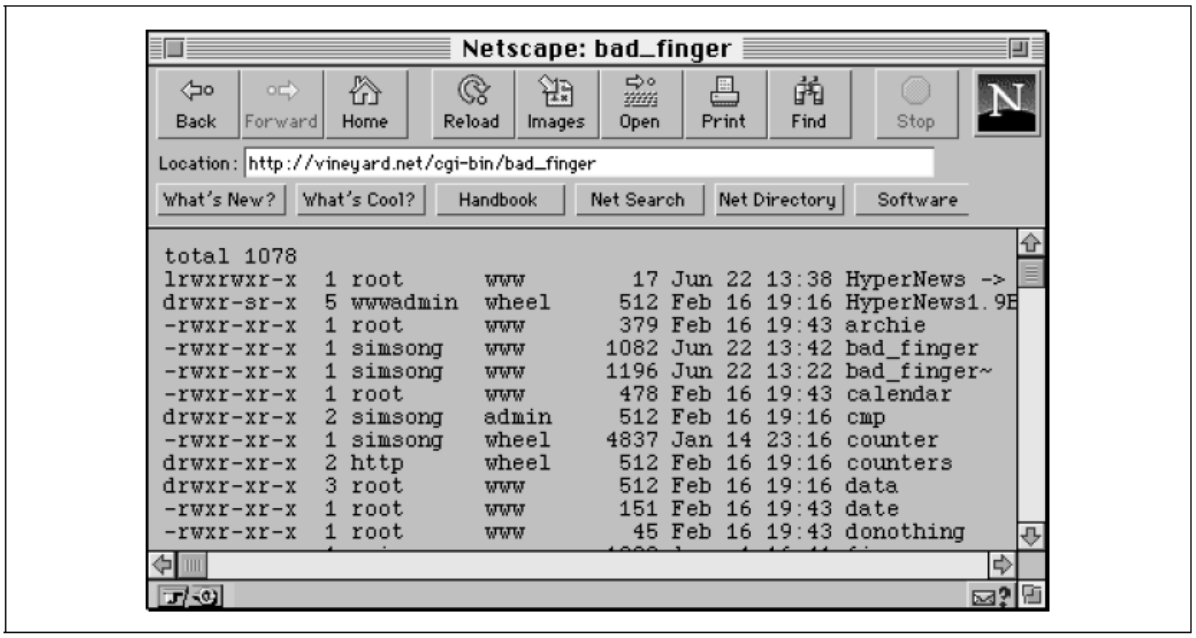

Figure 16-4. Output from the bad_finger script under attack

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.
- 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

This section outlines rules for writing more secure programs in different 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().
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**
    - 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()

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.

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.
- **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.
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.
- **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).
- **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.

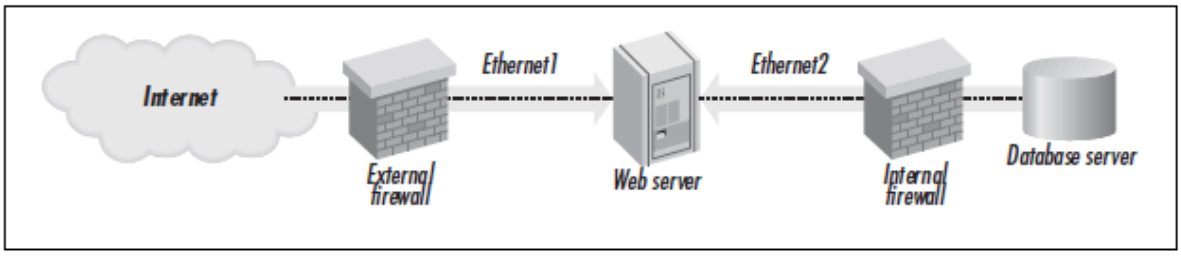

Figure 16-5. Connecting a database server and a web server to the Internet and your internal network with multiple firewalls.