1 **WEB LANGUAGES**

The Babylon of the 21st Century

2

Bab·y·lon – n.

1. A city or place of great luxury, sensuality and often vice and corruption.

2. A place of captivity or exile.

3. A place devoted to materialism and sensual pleasure.

3 **Introduction**

- Babylonians were an ancient people noted for many accomplishments
  - Having a single language aided in their accomplishments
  - Having a single language caused their demise: the Tower of Babel
- This is also the case for the Internet and the Web!
  - In its infancy, there was no "fun" in hacking Web sites
  - Now, there is eCommerce, eBusiness and a plethora of Web languages

4 **Languages of the Web**

- To communicate 2 peers must speak the same language
- On the Web, a number of languages exist each with strengths and weaknesses
- Web Languages – you need to:
  - Know a little of each
  - Understand the security implications of each
-

5 **HTML**

- File extensions: .html, .htm, .html4
- Hypertext Markup Language (HTML) is the underlying framework of the Web
- HTML standard is managed by W3C
- Made up of a series of *elements*
- Elements made up of *tags*, *attributes* and element *contents*:
  <tag attr=val>Ele. Cont.</tag>
- Simple elements can be used to gain unauthorized access on Web servers

6 **HTML: Form Element**

- <form>
  - The form is used as a container for user input controls
  - The form is the culprit for most Web site attacks: submission of characters (via a form) to an unsuspecting program
- <form action>
  - action attribute names the program that processes the input
  - Knowing the name of the program, attacker can get valuable info about the Web server

7 **HTML: Form Element**

- <form method>
  - The method attribute specifies the HTTP method used to transfer data from the client to the server
  - 2 methods exist that do this: GET & POST
  - By knowing the method, attacker can:
    - "listen" to a submission gaining valuable info (i.e., credit card numbers, passwords, etc.)
    - Possibly modify the submission producing abnormal results

8 **HTML: Script Element**

- <script language=<variable>>
  - The script element along with the language attribute allows an attacker to modify any client-side scripting being sent to the browser
  - Attacker can thus remove filtering and sanitization from scripts
  - Client-side scripting include:
    - JavaScript VBScript
    - Jscript XML

9 **HTML: Input Element**

- <input>
  - The input element is used to get user input
  - Used in conjunction with the form element
  - Specific attributes can be altered producing undesirable results on the server
- <input type=text>
  - This is the textbox
  - Text is sent to the server where it is processed
  - Text must be filtered and sanitized

10 **HTML: Input Element**

- <input type=hidden>
  - The hidden attribute renders form elements invisible
  - Used to pass data to the server that the user did not directly enter into the form, i.e., the item price in a shopping cart
  - Attacker can modify the value to something undesirable, i.e., change the price to something lower
  - If server does not check the value in the hidden field… surprise!

11 **HTML: Input Element**

- <input maxlength=<variable>>
  - maxlength attribute describes the maximum size of the input (not the size of the element)
  - Can be altered causing large strings to be submitted to unsuspecting server-side processors
- <input size=<variable>>
  - size attribute describes the size of the input element
  - Same problem as maxlength

12 **HTML: Applet Element**

- <applet>
  - Used to execute a Java applet program
  - Java compiles into a known byte-code
  - The byte-code can be "seen" by a packet sniffer
  - More on Java later…

13 **HTML: Object Element**

- <object>
  - Supplies the browser with information about data types not natively supported by the browser:
    - Applets
    - Plug-in
    - Some other helper
  - Attacker can send an email with embedded HTML containing <object> and execute an ActiveX control on the users system
  - Used mainly for spreading email viruses

14 **XHTML**
- File extensions: .html, .htm, .html4, .xhtml
- HTML + XML = XHTML
- HTML that complies with the XML standard
- HTML 4 is dead; always use XHTML
- Similar security issues as HTML though work is still being done…

15 **XML**
- Extensible Markup Language (XML)
- Used to describe data not markup
- Can define your own tags
- Document Type Definition (DTD) defines the tags and attributes
- Use the tags and attributes in an XML document
- Example in the text
- XML is a relatively "new" technology so attackers not sure yet

16 **Common Gateway Interface (CGI)**
- Older, mature technology
- Not a language but a set of guidelines
- Almost any language can be used with CGI
- Makes an extensive use of environment variables
  - Passes data to a script through Env. Vars.
  - Many opportunities for attacks

17 **Perl**
- Language that has been around since 1987
- Popularity due to portability and price
- Mostly used as a scripting language but can standalone
- Security never a fundamental component of the language
- On the Web, used with Common Gateway Interface (CGI) to do form processing

18 **Perl**
- The Perl/CGI sequence of events:
  - HTML displays form for user to fill in
  - Data submitted via a HTTP method to the processing program
  - Program does its processing and produces output in HTML
  - Browser displays the resulting HTML
- CGI is not used as much anymore… replaced with PHP, ASP, JSP, etc.

19 **PHP**
- File extentions: .php, .php3
- Uses the embedded model like ASP or JSP
- Mostly used on Linux systems running Apache
- Has a close tie to MySQL
- PHP is much like embedded perl
- Has many of the same features and problems of perl

20 **PHP**
- Web pages contain embedded PHP
  - <? phpinfo() ?>
- Security wise:
  - Use the same general processes as described in perl section
    - Input sanitization is critical
    - Limit your applications use of shell-outs
    - Check input sizes

- Know you php.ini, i.e., register_globals

21 **ColdFusion**
- Allair/Macromedia/Adobe Web development system
- Has 3 components:
  1. Application Server
  2. Markup Language
  3. Studio
- Application Server
  - "Brains" behind the system
  - Processes ColdFusion page requests

22 **ColdFusion**
- ColdFusion Markup Language (CFML)
  - Server-side language powers CF
  - Follows HTML conventions
  - Used with App. Server to create Web apps like shopping carts, online bank accounts, etc.
  - Pages stored as plain text
  - Tags provide functionality (XML)
    - DB connectivity
    - Post Office Protocol (POP)
    - Simple Mail Transfer Protocol (SMTP)
    - Component Object Model (COM)

23 **ColfFusion**
- Studio
  - Integrated Development Environment (IDE)
- Security Issues:
  - Sample files
    - People would use the sample scripts "as is"
    - Everyone can see them since they are public sample files
  - Unsanitized input (a theme to remember throughout this entire course!)

24 **Active Server Pages (ASP)**
- File extension: .asp
- Microsoft's version of a server-side scripting environment
- Designed to be used with Internet Information Server (IIS)
- Creates dynamic content like JSP, PHP & CGI/Perl
- Default language: VBScript (a stripped down version of MS Visual Basic)

25 **Active Server Pages (ASP)**
- Like PHP, embedded in HTML server-side:
  - <% =date %>
- Can also be client-side
  - <script language="VBScript">
- Problem with ASP (and ASP.NET) are well documented…
  - We could spend an entire course on these!
  - More later…

26 **Active Server Pages (ASP)**
- ActiveX
  - Microsoft's version of a browser plug-in
  - Programs built in C++, VB or Java and stored in cabinet files (.cab)
  - Linked to the Web page through an <object> tag with classid and codebase

attributes
- Security problems:
  - Attackers can create ActiveX controls that do file access or shell-out
  - Location of .cab file is revealed

27 **Java**
- Object oriented programming language
- Compiler produces byte-code
- Byte-code executed in a Virtual Machine
- VM's ported to various platforms
- Theory: compile once run anywhere
- 2 main types of Java code: client-based and server-based

28 **Java**
- Client-based Java
  - There are 2 formats of client-based Java: applets and scripting languages
  - Applets
    - Uses the <applet> tag
    - Downloaded and run by the client
    - Can be downloaded separately and decompiled giving attacker access to source

29 **Java**
- Client-based Java (cont)
  - Scripting Languages
    - JavaScript
      - Object-based not object-oriented
      - Useful (but limited) for input field checking
      - Example p. 43-44
      - Can remove the validation because JS is client-side
    - JScript
      - Microsoft's attempt to clone JavaScript
      - Allows access to ActiveX where JS does not

30 **Java**
- Server-based Java
  - Java Server Pages (JSP)
    - File extension: .jsp
    - Allows you to embed Java in your HTML
    - Needs a Java Application Server to run
      - App Server creates a Java Servlet
      - BEA WebLogic, Sun Glassfish, Adobe JRun, IBM WebSphere, Oracle Jdeveloper, Apache Tomcat
    - Provides db connectivity through many different "drivers"
    - Many different types of attacks
    - More on them later…

31 **Java**
- Serve-based Java (cont)
  - JHTML
    - File extension: .jhtml
    - Sun's JavaSoft standard for including Java in HTML
    - Uses the <java> tag
    - Basically, same issues as JSP
    - More on these later…

32 **Summary**

- All Web languages can be attacked in some way
- You need to understand the languages so you are prepared for the attack
- From this chapter you should understand the *nature* of the attack
- Obviously, there are many more types of attacks!