

Objectives

- Describe Web applications
- Explain Web application vulnerabilities
- Describe the tools used to attack Web servers

Web Servers

The two main Web servers are Apache (Open source) and IIS (Microsoft)

Understanding Web Applications

It is nearly impossible to write a program without bugs

- Some bugs create security vulnerabilities

Web applications also have bugs

- Web applications have a larger user base than standalone applications
- Bugs are a bigger problem for Web applications

Web Application Components

Static Web pages

- Created using HTML

Dynamic Web pages

- Need special components
 - <form> tags
 - Common Gateway Interface (CGI) scripts
 - Active Server Pages (ASP)
 - PHP
 - ColdFusion
 - Scripting languages like JavaScript
 - ODBC (Open Database connector)

Web Forms

Use the <form> element or tag in an HTML document

- Allows customer to submit information to the Web server

Web servers process information from a Web form by using a Web application

Easy way for attackers to intercept data that users submit to a Web server

Web form example

```
<html><body>
<form>
Enter your username:
<input type="text" name="username">
<br>
Enter your password:
<input type="text" name="password">
</form></body></html>
```

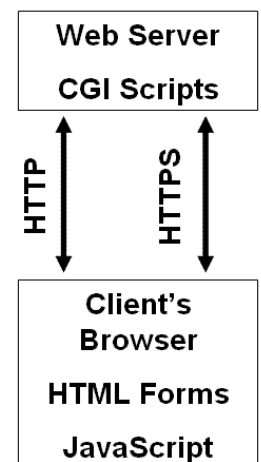
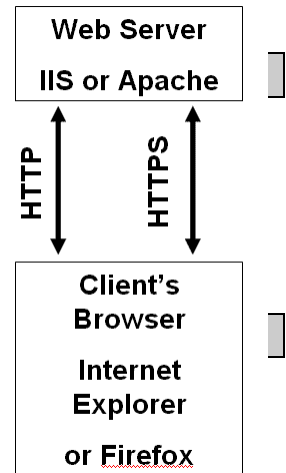
Common Gateway Interface (CGI)

Handles moving data from a Web server to a Web browser

The majority of dynamic Web pages are created with CGI and scripting languages

Describes how a Web server passes data to a Web browser

- Relies on Perl or another scripting language to create dynamic Web pages



Chapter 10: Hacking Web Servers

CGI Languages

CGI programs can be written in different programming and scripting languages

- C or C++
- Perl
- Unix shell scripting
- Visual Basic
- FORTRAN

CGI example

- Written in Perl
- Hello.pl
- Should be placed in the *cgi-bin* directory on the Web server

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "Hello Security Testers!";
```

Another CGI Example

Link Ch 10a: Sam's Feedback Form

Link Ch 10b: CGI Script in Perl that processes the data from the form

Active Server Pages (ASP)

Microsoft's server-side script engine

- HTML pages are static—always the same
- ASP creates HTML pages as needed. They are not static

ASP uses scripting languages such as JScript or VBScript

Not all Web servers support ASP

- IIS supports ASP
- Apache doesn't support ASP as well

Active Server Pages (ASP)

You can't see the source of an ASP page from a browser

This makes it harder to hack into, although not impossible

ASP examples at links

Ch 10d, e, f

Apache Web Server

Apache is the most popular Web Server program

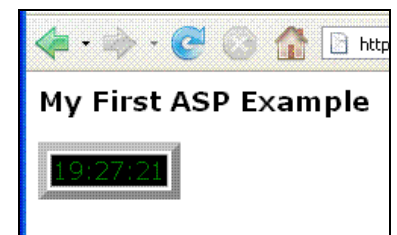
Advantages

- Stable and reliable
- Works on just about any *NIX and Windows platform
- It is free and open source
See links Ch 10g, 10h

Using Scripting Languages

Dynamic Web pages can be developed using scripting languages

- VBScript
- JavaScript
- PHP



```
<% @language = vbscript %>
<% option explicit %>
<html><head><title>ASP Example</head>
<body><table border=6><tr><td bgcolor=black>
<font face=verdana color=green size=3>
<% = time() %>
</font></td></tr></table></body>
</html>
```

Chapter 10: Hacking Web Servers

PHP: Hypertext Processor (PHP)

Enables Web developers to create dynamic Web pages

- Similar to ASP

Open-source server-side scripting language

- Can be embedded in an HTML Web page using PHP tags `<?php and ?>`

Users cannot see PHP code in their Web browser

Used primarily on UNIX systems

- Also supported on Macintosh and Microsoft platforms

PHP Example

```
<html><head><title>Example</title></head>
<body>
<?php
echo 'Hello, World!';
?>
</body></html>
```

- See links Ch 10k, 10l

PHP has known vulnerabilities

- See links Ch 10m, 10n

PHP is often used with MySQL Databases

ColdFusion

Server-side scripting language used to develop dynamic Web pages

Created by the Allaire Corporation

- Purchased by Macromedia, now owned by Adobe -- Expensive

Uses its own proprietary tags written in ColdFusion Markup Language (CFML)

CFML Web applications can contain other technologies, such as HTML or JavaScript

ColdFusion Example

```
<html><head><title>Ex</title></head>
<body>
<CFLOCATION URL="www.isecom.org/cf/index.htm" ADDTOKEN="NO">
</body>
</html>
```

- See links Ch 10o

ColdFusion Vulnerabilities

See links Ch 10p, 10q

Macromedia ColdFusion Vulnerabilities :

- 14.02.2007 : Adobe ColdFusion MX Default Error Page Client-Side Cross Site Scripting Vulnerability
- 11.12.2006 : Adobe Macromedia ColdFusion Information Disclosure and Cross Site Scripting Issues
- 11.10.2006 : Adobe Macromedia ColdFusion Verity Library Privilege Escalation Vulnerabilities
- 12.09.2006 : Adobe Macromedia ColdFusion Error Page Cross Site Scripting Vulnerability
- 12.09.2006 : Adobe Macromedia ColdFusion Denial of Service and Security Bypass Vulnerabilities
- 09.08.2006 : Adobe Macromedia ColdFusion MX AdminAPI Local Authentication Bypass Vulnerability
- 16.12.2005 : Macromedia ColdFusion Multiple Security Bypass Vulnerabilities
- 15.07.2005 : Macromedia JRun Internal Authentication Token Vulnerability
- 10.05.2005 : Macromedia ColdFusion MX Error Page Cross Site Scripting Issue
- 08.04.2005 : Macromedia ColdFusion MX Updater File Disclosure Vulnerability

VBScript

Visual Basic Script is a scripting language developed by Microsoft

You can insert VBScript commands into a static HTML page to make it dynamic

- Provides the power of a full programming language
- Executed by the client's browser

VBScript Example

```
<html><body>
<script type="text/vbscript">
document.write("<h1>Hello!</h1>")
document.write("Date Activated: " & date())
</script>
</body></html>
```

See link Ch 10r – works in IE, but not in Firefox

Firefox does not support VBScript (link Ch 10s)

VBScript vulnerabilities

- See links Ch 10t, 10u

JavaScript

Popular scripting language

JavaScript also has the power of a programming language

- Branching
- Looping
- Testing

JavaScript Example

```
<html><head>
<script type="text/javascript">
function chastise_user(){
alert("So, you like breaking rules?")
document.getElementById("cmdButton").focus()}
</script></head>
<body><h3>Don't click the button!</h3>
<form>
<input type="button" value="Don't Click!" name="cmdButton" onClick="chastise_user()" />
</form></body></html>
```

- See link Ch 10v – works in IE and Firefox

JavaScript Vulnerabilities

See link Ch 10w

Microsoft Security Bulletin MS02-009

Incorrect VBScript Handling in IE can Allow Web Pages to Read Local Files

Originally posted: February 21, 2002

Updated: May 09, 2003

JavaScript vulnerabilities surface in multiple browsers

by [John McCormick](#) | [More from John McCormick](#) | 6/12/06

Tags: [Web browsers](#) | [Security threats](#) | [Internet Explorer \(IE\)](#) | [Patches](#)

Chapter 10: Hacking Web Servers

Connecting to Databases

Web pages can display information stored on databases

There are several technologies used to connect databases with Web applications

- Technology depends on the OS used
 - ODBC
 - OLE DB
 - ADO
- Theory is the same

Open Database Connectivity (ODBC)

Standard database access method developed by the SQL Access Group

ODBC interface allows an application to access

- Data stored in a database management system (DBMS)
- Can use Oracle, SQL, or any DBMS that understands and can issue ODBC commands

Interoperability among back-end DBMS is a key feature of the ODBC interface

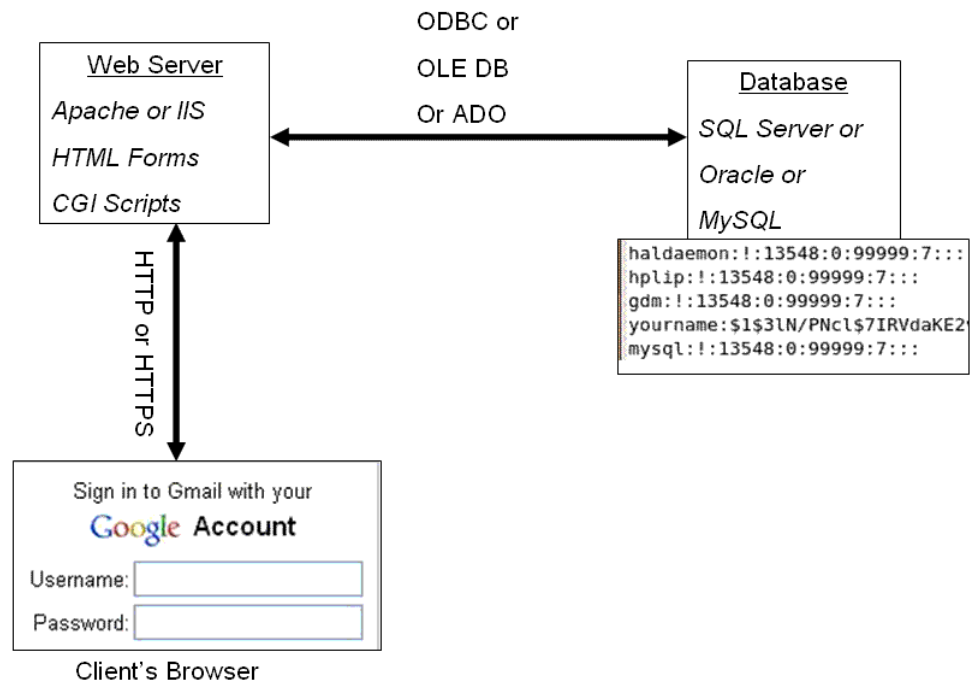
ODBC defines

- Standardized representation of data types
- A library of ODBC functions
- Standard methods of connecting to and logging on to a DBMS

OLE DB and ADO

Object Linking and Embedding Database (OLE DB) and ActiveX Data Objects (ADO)

- These two more modern, complex technologies replace ODBC and make up "Microsoft's Universal Data Access"
- See link Ch 10x



Understanding Web Application Vulnerabilities

Many platforms and programming languages can be used to design a Web site
Application security is as important as network security

Attackers controlling a Web server can

- Deface the Web site
- Destroy or steal company's data
- Gain control of user accounts
- Perform secondary attacks from the Web site
- Gain root access to other applications or servers

Chapter 10: Hacking Web Servers

Open Web Application Security Project (OWASP)

- Open, not-for-profit organization dedicated to finding and fighting vulnerabilities in Web applications
- Publishes the Ten Most Critical Web Application Security Vulnerabilities

Top-10 Web application vulnerabilities

Unvalidated parameters

- HTTP requests from browsers that are not validated by the Web server
- Inserted form fields, cookies, headers, etc. (See link Ch 10y)

Broken access control

- Developers implement access controls but fail to test them properly
For example, letting an authenticated user read another user's files

Broken account and session management

- Enables attackers to compromise passwords or session cookies to gain access to accounts

Cross-site scripting (XSS) flaws

- Attackers inject code into a web page, such as a forum or guestbook
- When others user view the page, confidential information is stolen
- See link Ch 10za

Buffer overflows

- It is possible for an attacker to use C or C++ code that includes a buffer overflow

Command injection flaws

- An attacker can embed malicious code and run a program on the database server
- Example: SQL Injection

Error-handling problems

- Error messages may reveal information that an attacker can use

Insecure use of cryptography

- Storing keys, certificates, and passwords on a Web server can be dangerous

Remote administration flaws

- Attacker can gain access to the Web server through the remote administration interface

Web and application server misconfiguration

- Any Web server software out of the box is usually vulnerable to attack
 - Default accounts and passwords
 - Overly informative error messages

WebGoat project

- Helps security testers learn how to perform vulnerabilities testing on Web applications
- Developed by OWASP

It's like HackThisSite without the helpful forum

- Tutorials for WebGoat are being made, but they aren't yet ready

Assessing Web Applications

Issues to consider

- Dynamic Web pages
- Connection to a backend database server
- User authentication
- What platform was used?

Does the Web Application Use Dynamic Web Pages?

Static Web pages do not create a secure environment

IIS attack example: Directory Traversal

- Adding `..\` to a URL refers to a directory above the Web page directory
- Early versions of IIS filtered out `\`, but not `%c1%9c`, which is a Unicode version of the same character
- See link Ch 10 zh

Connection to a Backend Database Server

Security testers should check for the possibility of SQL injection being used to attack the system

SQL injection involves the attacker supplying SQL commands on a Web application field

SQL Injection Example

HTML form collects *name* and *pw*

SQL then uses those fields:

- `SELECT * FROM customer WHERE username = 'name' AND password = 'pw'`

If a hacker enters a name of

`' OR 1=1 --`

The SQL becomes:

- `SELECT * FROM customer WHERE username = '' OR 1=1 --' AND password = 'pw'`

Which is always true, and returns all the records

HackThisSite



Basic testing should look for

- Whether you can enter text with punctuation marks
- Whether you can enter a single quotation mark followed by any SQL keywords
- Whether you can get any sort of database error when attempting to inject SQL

User Authentication

Many Web applications require another server to authenticate users

Examine how information is passed between the two servers

- Encrypted channels

Verify that logon and password information is stored on secure places

Authentication servers introduce a second target

What Platform Was Used?

Popular platforms include:

- IIS with ASP and SQL Server (Microsoft)
- Linux, Apache, MySQL, and PHP (LAMP)

Footprinting is used to find out the platform

- The more you know about a system the easier it is to gather information about its vulnerabilities

Tools of Web Attackers and Security Testers

Choose the right tools for the job

Attackers look for tools that enable them to attack the system

- They choose their tools based on the vulnerabilities found on a target system or application

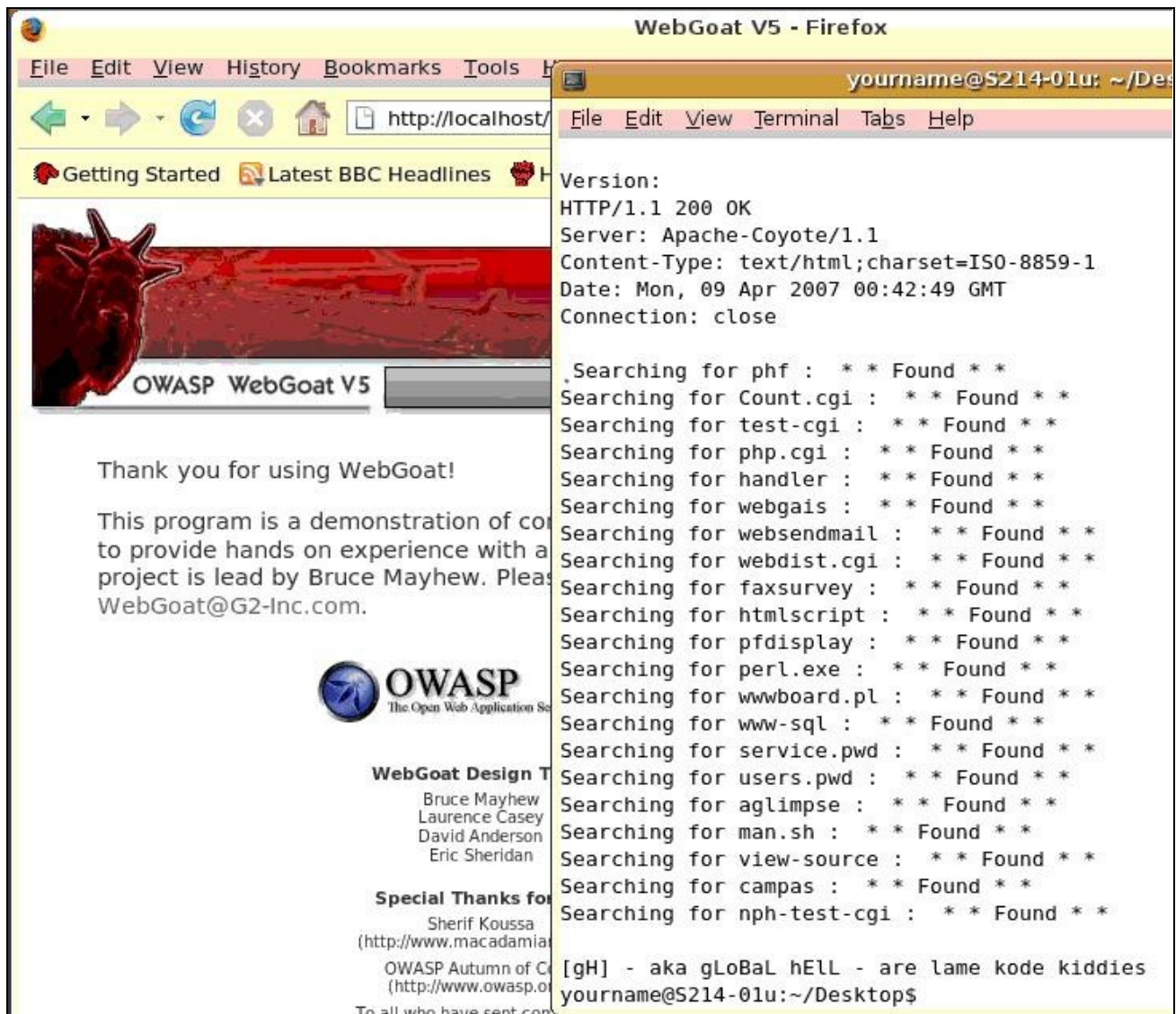
Web Tools

Cgiscan.c: CGI scanning tool

- Written in C in 1999 by Bronc Buster
- Tool for searching Web sites for CGI scripts that can be exploited
- One of the best tools for scanning the Web for systems with CGI vulnerabilities

See link Ch 10zi

cgiscan and WebGoat



Phfscan.c

- Written to scan Web sites looking for hosts that could be exploited by the PHF bug
- The PHF bug enables an attacker to download the victim's `/etc/passwd` file
- It also allows attackers to run programs on the victim's Web server by using a particular URL

See links Ch 10zj, 10 zk

Chapter 10: Hacking Web Servers

Wfetch: GUI tool from Microsoft

- Displays information that is not normally shown in a browser, such as HTTP headers
 - It also attempts authentication using
 - Multiple HTTP methods
 - Configuration of host name and TCP port
 - HTTP 1.0 and HTTP 1.1 support
 - Anonymous, Basic, NTLM, Kerberos, Digest, and Negotiation authentication types
 - Multiple connection types
 - Proxy support
 - Client-certificate support
- See link Ch 10zl

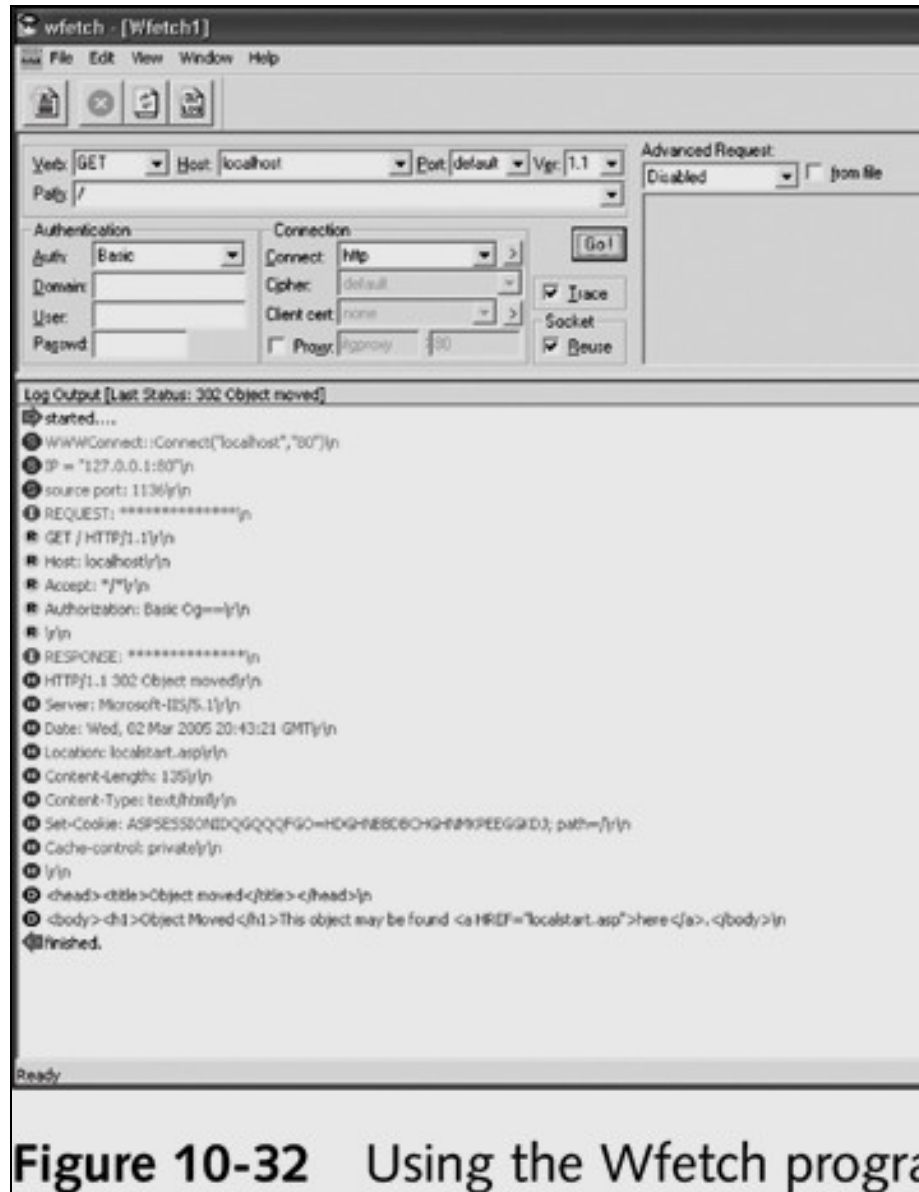


Figure 10-32 Using the Wfetch program

Last modified 4-8-07 6 pm