Chapter 10 – Hacking Web Servers

Notes

Overview

This chapter describes Web applications and their components. You will learn about Web applications and their vulnerabilities and explore several tools used to attack Web servers.

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

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

Tips

Understanding Web Applications

1. This section will cover several aspects of Web applications such as its components, the use of scripting languages, and connectivity to databases.

Web Application Components

1. Understand the difference between a static Web page and a dynamic Web page. Static Web pages display the same information regardless of the time or the user. Dynamic Web pages can vary the information that's displayed.

Web Forms

1. Web forms allow users to send information that can be processed by Web applications at the Web server.

2. Using Web forms can produce security vulnerabilities.

Tip	Check out http://www.microsoft.com/technet/security/bulletin/MS00-100.mspx for an example of a Web form vulnerability.

Common Gateway Interface

1. The Common Gateway Interface (CGI) is another standard that handles moving data from a Web server to a Web browser and enables Web designers to create dynamic Web pages.

2. Understand how to write CGIs and see the example from the book.

Third Party Frameworks and Libraries

1. Frameworks are typically called on for a specific purpose and are designed to make programming easier.

2. The use of libraries saves developer time and means less documentation is required for complex routines of custom code.

Active Server Pages

1. Understand the differences between HTML Web pages and ASP Web pages. ASP Web pages use scripting languages such as JScript or VBScript to create Web pages on the fly (after a user has requested a page).

2. Not all Web servers support ASP. Understand how to install and configure Internet Information Services (IIS) to create Web servers that support ASP Web pages.

3. View what an ASP Web page looks like with the example from the book.

Tip	Read https://www.sans.org/reading-room/whitepapers/securecode/security-checklist-web-application-design-1389 an article about a security checklist for web application design.

Apache Web Server

1. Apache Web Server is another Web server program option. Apache works on almost any *nix and Windows platform and it is free.

Using Scripting Languages

1. Web pages can be developed with several scripting languages, such as VBScript and JavaScript. This section explains several scripting languages that can be used to create dynamic Web pages.

PHP Hypertext Processor

1. PHP is an open-source server-side scripting language that allows Web developers to create dynamic Web pages. PHP is similar to ASP. It is used primarily on UNIX systems, but it is also supported on Macintosh and Windows platforms.

ColdFusion

1. ColdFusion is a server-side scripting language that allows Web developers to create dynamic Web pages. ColdFusion uses its own proprietary tags written in ColdFusion Markup Language (CFML).

VBScript

1. VBScript is another scripting language developed by Microsoft. VBScript is used to convert static Web pages into dynamic Web pages.

Tip	See http://www.tutorialspoint.com/vbscript/ for a VBScript tutorial.

JavaScript

1. JavaScript is another scripting language used to create dynamic Web pages. JavaScript introduces the power of a full programming language into a Web page.

Tip	See http://www.w3schools.com/js/default.asp for a JavaScript tutorial.

Connecting to Databases

1. Most Web pages displaying company information to users are stored on a database server. This section explains three technologies used to connect databases to Web pages: Open Database Connectivity (ODBC), Object Linking and Embedding Database (OLE), and ActiveX Data Objects (ADO).

Open Database Connectivity

1. ODBC is a standard database access method developed by the SQL Access Group.

2. Understand the main features of the ODBC interface. This interface allows programs to connect and work with a database regardless of the DBMS that is being used.

Object Linking and Embedding Database

1. OLE DB is a set of interfaces that also enables applications to access data stored in DBMS.

2. OLE DB was developed by Microsoft and was designed to be faster, more efficient, and more stable than its predecessor, ODBC.

3. Understand how OLE DB uses connection strings and providers to connect to different DBMSs.

ActiveX Data Objects

1. ActiveX is a set of technologies that allow Web applications to interact with a database.

2. The steps used to access a database from a Web page.
    a. Create an ADO connection
    b. Open the database connection you just created
    c. Create an ADO recordset
    d. Open the recordset
    e. Select the data you need
    f. Close the recordset and the database connection

Understanding Web Applications Vulnerabilities

1. Many security professionals pay attention only to network security details and forget about application security.

2. The problems a company may face once an attacker gains control over a Web server.
    a. Deface the Web site
    b. Destroy or steal company's data or sell its contents
    c. Gain control of user accounts
    d. Perform secondary attacks from the Web site
    e. Gain root access to other applications or servers

Application Vulnerabilities Countermeasures

1. The Open Web Application Security Project (OWASP) is a not-for-profit foundation dedicated to finding and fighting vulnerabilities in Web applications. OWASP publishes the Ten Most Critical Web Application Vulnerabilities list.

2. View the top ten Web applications vulnerabilities listed by the OWASP.

3. Understand the WebGoat project some of tests that can be accomplished using WebGoat.

Web Application Test Execution

1. The two techniques by which an application can be tested:
    a. Static Application Security Testing (SAST)
    b. Dynamic Application Security Testing (DAST)

Information Gathering and Architecture Mapping

1. Some questions that should be considered during this phase:
    a. Does the application have a database?
    b. Does the application require authentication?
    c. Does the application have static or dynamic pages?
    d. What languages and platforms does the application use?
    e. Are there devices in-between your Web browser and the application designed to stop attacks from occurring?
    f. How does data flow in the application?

Platform Security and Configuration

1. Knowing the platform and technology a Web application was developed with makes the job of attackers easier.

Authentication and Session Testing

1. Many Web applications require a server other than the Web server to authenticate users. In this case, you should examine how authentication information is passed between servers and whether logon and password information is stored in a secure location

Authorization Testing

1. Authorization is the act of checking a user's privileges to understand if they should or should not have access to a page, field, resource, or action in an application.

2. Authorization testing can reveal major areas of concern and is an important part of any application test.

Input Validation

1. SQL injection is a type of attack where the attacker can pass SQL commands when asked to fill a Web application form field.

2. Review detailed examples of SQL injection attacks in the book.

3. Some testing recommendations. Basic testing should look for:
    a. Whether you can enter text with punctuation marks
    b. Whether you can enter a single quotation mark followed by any SQL keywords
    c. Whether you get any sort of database error when attempting to inject SQL

Tip	Read a SQL Injection Walkthrough at http://www.securiteam.com/securityreviews/5DP0N1P76E.html.

Error Handling

1. Developers should minimize the amount of information shared with attackers when an application encounters an error. Only a generic message should be displayed to users in error cases.

Cryptography Testing

1. Some of the problems that occur in cryptography:
    a. using bad random number generators
    b. using a known weak method of encryption
    c. an application that doesn't enforce the sue of secure channels
    d. using a self-signed certificate instead of a purchased certificate

Business Logic Testing

1. Business logic refers to the flow a user is expected to follow in an application to accomplish a goal. Business logic testing involves utilizing creative ways to bypass logic checks.

Client-Side Testing

1. Understand client-side issues that arise from code executing on the user's machine.

2. Client-side controls are insufficient on their own and should be paired with server-side controls that cannot be bypassed.

Tools of Web Attackers and Security Testers

1. This section describes several Web tools used to attack or test Web applications and servers.

Web Tools

1. The Kali DVD is packed with free tools for hacking Web applications. They are located in the Kali, Web Application Analysis menu.

2. Firefox and Chrome each come with a similar set of developer tools that are useful for an application security tester.

3. Burp Suite, which is included in Kali Linux, allows you to intercept traffic between the Web browser and the server to inspect and manipulate requests before sending to the server.

4. Understand the similarities that Burp Suite has with Zed Attack Proxy.

5. Wapiti is a Web application vulnerability scanner that uses a black box approach, meaning it doesn't inspect code.

Tip	Read more about Wapiti at http://wapiti.sourceforge.net/.

6. Wfetch is a GUI testing tool that you can download for free from Microsoft. This tool queries the status of a Web server and attempts authentication using any of the following methods:
    a. Multiple HTTP methods
    b. Configuration of host name and TCP port
    c. HTTP 1.0 and HTTP 1.1 support
    d. Anonymous, Basic, NTLM, Kerberos, Digest, and Negotiation authentication types
    e. Multiple connection types
    f. Proxy support
    g. Client-certificate support
    h. Capability to enter requests manually or have them read from a file
    i. On-screen and file-based logging

Additional Resources

1. IIS 7.0 and later Security:
   https://www.iis.net/configreference/system.webserver/security?showTreeNavigation=true

2. IIS Lockdown and Urlscan:
   http://www.securityfocus.com/infocus/1755

3. ODBC tutorial:
   http://www.easysoft.com/developer/languages/perl/dbd_odbc_tutorial_part_1.html

4. SQL Injection Attacks – What is it and how to prevent it?:
   http://www.zdnet.com/article/sql-injection-attack-what-is-it-and-how-to-prevent-it/

5. PHF Prober Perl Script:
   http://www.eng.auburn.edu/users/rayh/software/phf.html

Key Terms

Active Server Pages (ASP)
ActiveX Data Objects (ADO)
Asynchronous JavaScript and XML (AJAX)
ColdFusion
Common Gateway Interface (CGI)
dynamic Web pages
Dynamic Application Security Testing (DAST)
Object Linking and Embedding Database (OLE DB)
Open Database Connectivity (ODBC)
Open Web Application Security Project (OWASP)
PHP Hypertext Processor (PHP)
SQL injection (SQLi)
static Web pages
Static Application Security Testing (SAST)
virtual directory
WebGoat