

FREE E-BOOK DOWNLOAD

Web Application Vulnerabilities

DETECT, EXPLOIT, PREVENT

Learn to Attack and Defend Mission-Critical Web Applications!

- Defend Web-Based Applications Developed with AJAX, SOAP, XMLPRC, and More
- See why Cross-Site Scripting Attacks Are So Devastating
- Download Working Code from the Companion Web Site.

Michael Cross
Steven Kapinos
Haroon Meer
Igor Muttik PhD
Steve Palmer
Petko "pdp" D. Petkov

Web Application Vulnerabilities Detect, Exploit, Prevent

Michael Cross
Steven Kapinos
Haroon Meer
Igor Muttik PhD

Steve Palmer
Petko "pdp" D. Petkov
Roger Shields
Roelof Temmingh

Elsevier, Inc., the author(s), and any person or firm involved in the writing, editing, or production (collectively "Makers") of this book ("the Work") do not guarantee or warrant the results to be obtained from the Work.

There is no guarantee of any kind, expressed or implied, regarding the Work or its contents. The Work is sold AS IS and WITHOUT WARRANTY. You may have other legal rights, which vary from state to state.

In no event will Makers be liable to you for damages, including any loss of profits, lost savings, or other incidental or consequential damages arising out from the Work or its contents. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

You should always use reasonable care, including backup and other appropriate precautions, when working with computers, networks, data, and files.

Syngress Media[®], Syngress[®], "Career Advancement Through Skill Enhancement[®]", "Ask the Author UPDATE[®]", and "Hack Proofing[®]", are registered trademarks of Elsevier, Inc. "Syngress: The Definition of a Serious Security Library"[™], "Mission Critical[™]", and "The Only Way to Stop a Hacker is to Think Like One[™]" are trademarks of Elsevier, Inc. Brands and product names mentioned in this book are trademarks or service marks of their respective companies.

KEY	SERIAL NUMBER
001	HJIRTCV764
002	PO9873D5FG
003	829KM8NJH2
004	BAL923457U
005	CVPLQ6WQ23
006	VBP965T5T5
007	HJJJ863WD3E
008	2987GVTWMK
009	629MP5SDJT
010	IMWQ295T6T

PUBLISHED BY
Syngress Publishing, Inc.
Elsevier, Inc.
30 Corporate Drive
Burlington, MA 01803

Web Application Vulnerabilities Detect, Exploit, Prevent

Copyright © 2007 by Elsevier, Inc. All rights reserved. Printed in the United States of America.

Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

Printed in the United States of America

1 2 3 4 5 6 7 8 9 0

ISBN 13: 978-1-59749-209-6

Publisher: Andrew Williams
Page Layout and Art: SPi
Copy Editor: Audrey Doyle and Judy Eby

For information on rights, translations, and bulk sales, contact Matt Pedersen, Commercial Sales Director and Rights, at Syngress Publishing; email m.pedersen@elsevier.com.

Visit us at

www.syngress.com

Syngress is committed to publishing high-quality books for IT Professionals and delivering those books in media and formats that fit the demands of our customers. We are also committed to extending the utility of the book you purchase via additional materials available from our Web site.

SOLUTIONS WEB SITE

To register your book, visit www.syngress.com/solutions. Once registered, you can access our solutions@syngress.com Web pages. There you may find an assortment of value-added features such as free e-books related to the topic of this book, URLs of related Web sites, FAQs from the book, corrections, and any updates from the author(s).

ULTIMATE CDs

Our Ultimate CD product line offers our readers budget-conscious compilations of some of our best-selling backlist titles in Adobe PDF form. These CDs are the perfect way to extend your reference library on key topics pertaining to your area of expertise, including Cisco Engineering, Microsoft Windows System Administration, CyberCrime Investigation, Open Source Security, and Firewall Configuration, to name a few.

DOWNLOADABLE E-BOOKS

For readers who can't wait for hard copy, we offer most of our titles in downloadable Adobe PDF form. These e-books are often available weeks before hard copies, and are priced affordably.

SYNGRESS OUTLET

Our outlet store at syngress.com features overstocked, out-of-print, or slightly hurt books at significant savings.

SITE LICENSING

Syngress has a well-established program for site licensing our e-books onto servers in corporations, educational institutions, and large organizations. Contact us at sales@syngress.com for more information.

CUSTOM PUBLISHING

Many organizations welcome the ability to combine parts of multiple Syngress books, as well as their own content, into a single volume for their own internal use. Contact us at sales@syngress.com for more information.

Contributing Authors

Michael Cross (MCSE, MCP+I, CNA, Network+) is an Internet Specialist/Computer Forensic Analyst with the Niagara Regional Police Service (NRPS). He performs computer forensic examinations on computers involved in criminal investigation. He also has consulted and assisted in cases dealing with computer-related/Internet crimes. In addition to designing and maintaining the NRPS Web site at www.nrps.com and the NRPS intranet, he has provided support in the areas of programming, hardware, and network administration. As part of an information technology team that provides support to a user base of more than 800 civilian and uniform users, he has a theory that when the users carry guns, you tend to be more motivated in solving their problems.

Michael also owns KnightWare (www.knightware.ca), which provides computer-related services such as Web page design, and Bookworms (www.bookworms.ca), where you can purchase collectibles and other interesting items online. He has been a freelance writer for several years, and he has been published more than three dozen times in numerous books and anthologies. He currently resides in St. Catharines, Ontario, Canada, with his lovely wife, Jennifer, his darling daughter, Sara, and charming son, Jason.

Igor Muttik PhD is a senior architect with McAfee Avert™. He started researching computer malware in 1980s when anti-virus industry was in its infancy. He is based in the UK and worked as a virus researcher for Dr. Solomon's Software where he later headed the anti-virus research team. Since 1998 he has run Avert Research in EMEA and switched to his architectural role in 2002. Igor is a key contributor to the core security technology at McAfee. He takes particular interest in new emerging malware techniques, and in the design of security software and hardware appliances. Igor holds a PhD degree in physics and mathematics from Moscow University. He is a regular speaker at major international security conferences and a member of the Computer Antivirus Research Organization.

Haroon Meer is the Technical Director of SensePost. He joined SensePost in 2001 and has not slept since his early childhood. He has played in most aspects of IT Security from development to deployment and currently gets most of his kicks from reverse engineering, application assessments, and similar forms of pain. Haroon has spoken and trained at Black Hat, Defcon, Microsoft Tech-Ed, and other conferences. He loves “Deels,” building new things, breaking new things, reading, deep find-outering, and making up new words. He dislikes sleep, pointless red-tape, dishonest people, and watching cricket.

Steve Palmer has 14 years of experience in the information technology industry. Steve has worked for several very successful security boutiques as an ethical hacking consultant. Steve has found hundreds of previously undiscovered critical vulnerabilities in a wide variety of products and applications for a wide variety of clients. Steve has performed security assessments and penetration tests for clients in many diverse commercial industries and government agencies. He has performed security assessments for companies in many different verticals such as the entertainment, oil, energy, pharmaceutical, engineering, automotive, aerospace, insurance, computer & network security, medical, and financial & banking industries. Steve has also performed security assessments for government agencies such as the Department of Interior, Department of Treasury, Department of Justice, Department of Interior, as well as the Intelligence Community. In 2001, Steve’s findings contributed to the entire Department of Interior being disconnected from the Internet during the Cobel vs Norton lawsuit. Prior to being a security consultant Steve worked as a System Administrator, administering firewalls, UNIX systems, and databases for the Department of Defense, Department of Treasury, and the Department of Justice. Prior to that, Steve served 6 years in the United States Navy as an Electronics Technician. Steve has also written several security tools which have yet to be released publicly. Steve is also a member of the Department of Justice’s Infragard organization.

Petko “pdp” D. Petkov is a senior IT security consultant based in London, United Kingdom. His day-to-day work involves identifying vulnerabilities, building attack strategies and creating attack tools and penetration testing

infrastructures. Petko is known in the underground circles as pdp or architect but his name is well known in the IT security industry for his strong technical background and creative thinking. He has been working for some of the world's top companies, providing consultancy on the latest security vulnerabilities and attack technologies.

His latest project, GNUCITIZEN (gnucitizen.org), is one of the leading web application security resources on-line where part of his work is disclosed for the benefit of the public. Petko defines himself as a cool hunter in the security circles.

He lives with his lovely girlfriend Ivana, without whom his contribution to this book would not have been possible.

Roelof Temmingh Born in South Africa, Roelof studied at the University of Pretoria and completed his Electronic Engineering degree in 1995. His passion for computer security had by then caught up with him and manifested itself in various forms. He worked as developer, and later as a system architect at an information security engineering firm from 1995 to 2000. In early 2000 he founded the security assessment and consulting firm SensePost along with some of the leading thinkers in the field. During his time at SensePost he was the Technical Director in charge of the assessment team and later headed the Innovation Centre for the company. Roelof has spoken at various international conferences such as Blackhat, Defcon, Cansecwest, RSA, Ruxcon, and FIRST. He has contributed to books such as *Stealing the Network: How to Own a Continent*, *Penetration Tester's Open Source Toolkit*, and was one of the lead trainers in the "Hacking by Numbers" training course. Roelof has authored several well known security testing applications like Wikto, Crowbar, BiDiBLAH and Suru. At the start of 2007 he founded Paterva in order to pursue R&D in his own capacity. At Paterva Roelof developed an application called Evolution (now called Maltego) that has shown tremendous promise in the field of information collection and correlation.

Contents

Chapter 1 Introduction to Web Application Hacking	1
Introduction	2
Web Application Architecture Components	3
The Web Server	3
The Application Content	3
The Data Store	4
Complex Web Application Software Components	4
Login	4
Session Tracking Mechanism	6
User Permissions Enforcement	9
Role Level Enforcement	10
Data Access	10
Application Logic	10
Logout	11
Putting it all Together	11
The Web Application Hacking Methodology	12
Define the Scope of the Engagement	13
Before Beginning the Actual Assessment	14
Open Source Intelligence Scanning	15
Default Material Scanning	16
Base Line the Application	17
Fuzzing	18
Exploiting/Validating Vulnerabilities	19
Reporting	20
The History of Web Application Hacking and the Evolution of Tools	21
Example 1: Manipulating the URL Directly (GET Method Form Submittal)	26
Example 2: The POST Method	31
Example 3: Man in the Middle Sockets	37
The Graphical User Interface Man in the Middle Proxy	45
Common (or Known) Vulnerability Scanners	49
Spiders and other Crawlers	49
Automated Fuzzers	49
All in One and Multi Function Tools	49
OWASP's WebScarab Demonstration	50

Starting WebScarab.	52
Next: Create a new session.	53
Next: Ensure the Proxy Service is Listening	56
Next, Configure Your Web Browser	57
Next, Configure WebScarab to Intercept Requests.	59
Next, Bring up the Summary Tab.	60
Web Application Hacking Tool List	68
Security E-Mail Lists	69
Summary.	73

Chapter 2 Information Gathering Techniques.75

Introduction	76
The Principles of Automating Searches	76
The Original Search Term	80
Expanding Search Terms	80
E-mail Addresses	81
Telephone Numbers.	83
People.	85
Getting Lots of Results	85
More Combinations.	88
Using “Special” Operators	88
Getting the Data From the Source	89
Scraping it Yourself – Requesting and Receiving Responses.	89
Scraping it Yourself – The Butcher Shop	95
Dapper	100
Aura/EvilAPI	101
Using Other Search Engines.	102
Parsing the Data	102
Parsing E-mail Addresses	102
Domains and Sub-domains.	106
Telephone Numbers.	107
Post Processing.	109
Sorting Results by Relevance.	109
Beyond Snippets	111
Presenting Results	111
Applications of Data Mining.	112
Mildly Amusing	112
Most Interesting.	115
Taking It One Step Further	127
Collecting Search Terms	130
On the Web	130

Spying on Your Own	132
Search Terms	132
Gmail	135
Honey Words	137
Referrals	139
Summary	141

Chapter 3 Introduction to Server Side

Input Validation Issues	143
Introduction	144
Cross Site Scripting (XSS)	146
Presenting False Information	147
How this Example Works	148
Presenting a False Form	149
Exploiting Browser Based Vulnerabilities	152
Exploit Client/Server Trust Relationships	152

Chapter 4 Client-Side Exploit Frameworks 155

Introduction	156
AttackAPI	156
Enumerating the Client	161
Attacking Networks	172
Hijacking the Browser	180
Controlling Zombies	184
BeEF	188
Installing and Configuring BeEF	189
Controlling Zombies	190
BeEF Modules	191
Standard Browser Exploits	194
Port Scanning with BeEF	195
Inter-protocol Exploitation and Communication with BeEF	196
CAL9000	198
XSS Attacks, Cheat Sheets, and Checklists	199
Encoder, Decoders, and Miscellaneous Tools	202
HTTP Requests/Responses and Automatic Testing	204
Overview of XSS-Proxy	207
XSS-Proxy Hijacking Explained	210
Browser Hijacking Details	212
Initialization	212
Command Mode	213
Attacker Control Interface	215

Using XSS-Proxy: Examples	216
Setting Up XSS-Proxy	216
Injection and Initialization Vectors For XSS-Proxy	219
HTML Injection	219
JavaScript Injection	220
Handoff and CSRF With Hijacks	222
CSRF	222
Handoff Hijack to Other Sites	222
Sage and File:// Hijack With Malicious RSS Feed	223
Summary	243
Solutions Fast Track	243
Frequently Asked Questions	245
Chapter 5 Web-Based Malware	247
Introduction	248
Attacks on the Web	248
Hacking into Web Sites	250
Index Hijacking	252
DNS Poisoning (Pharming)	257
Malware and the Web: What, Where, and How to Scan	262
What to Scan	262
Where to Scan	265
How to Scan	266
Parsing and Emulating HTML	268
Browser Vulnerabilities	271
Testing HTTP-scanning Solutions	273
Tangled Legal Web	274
Summary	276
Solutions Fast Track	276
Frequently Asked Questions	281
Chapter 6 Web Server and Web Application Testing with BackTrack	283
Objectives	284
Introduction	284
Web Server Vulnerabilities: A Short History	284
Web Applications: The New Challenge	285
Chapter Scope	285
Approach	286
Web Server Testing	286

CGI and Default Pages Testing	288
Web Application Testing	289
Core Technologies	289
Web Server Exploit Basics.	289
What Are We Talking About?	289
Stack-Based Overflows	290
Heap-based Overflows.	293
CGI and Default Page Exploitation.	293
Web Application Assessment	296
Information Gathering Attacks	296
File System and Directory Traversal Attacks	296
Command Execution Attacks	297
Database Query Injection Attacks	297
Cross-site Scripting Attacks.	298
Impersonation Attacks	298
Parameter Passing Attacks	298
Open Source Tools.	298
Intelligence Gathering Tools	299
Scanning Tools	307
Assessment Tools.	319
Authentication.	323
Proxy	335
Exploitation Tools.	337
Metasploit	337
SQL Injection Tools	341
DNS Channel.	344
Timing Channel	345
Requirements	345
Supported Databases	345
Example Usage	346
Case Studies: The Tools in Action	348
Web Server Assessments	348
CGI and Default Page Exploitation.	355
Web Application Assessment	363
Chapter 7 Securing Web Based Services	381
Introduction	382
Web Security.	382
Web Server Lockdown	382
Managing Access Control.	383

Handling Directory and Data Structures	384
Directory Properties	384
Eliminating Scripting Vulnerabilities	386
Logging Activity	387
Performing Backups	387
Maintaining Integrity	388
Finding Rogue Web Servers	388
Stopping Browser Exploits	389
Exploitable Browser Characteristics	390
Cookies	390
Web Spoofing	392
Web Server Exploits	395
SSL and HTTP/S	396
SSL and TLS	397
HTTP/S	398
TLS	399
S-HTTP	400
Instant Messaging	400
Packet Sniffers and Instant Messaging	401
Text Messaging and Short Message Service (SMS)	402
Web-based Vulnerabilities	403
Understanding Java-, JavaScript-, and ActiveX-based Problems	404
Java	404
ActiveX	406
Dangers Associated with Using ActiveX	409
Avoiding Common ActiveX Vulnerabilities	411
Lessening the Impact of ActiveX Vulnerabilities	412
Protection at the Network Level	412
Protection at the Client Level	413
JavaScript	414
Preventing Problems with Java, JavaScript, and ActiveX	415
Programming Secure Scripts	418
Code Signing: Solution or More Problems?	419
Understanding Code Signing	420
The Benefits of Code Signing	420
Problems with the Code Signing Process	421
Buffer Overflows	422
Making Browsers and E-mail Clients More Secure	424
Restricting Programming Languages	424

Keep Security Patches Current	425
Securing Web Browser Software	426
Securing Microsoft IE	426
CGI	431
What is a CGI Script and What Does It Do?	431
Typical Uses of CGI Scripts	433
Break-ins Resulting from Weak CGI Scripts	434
CGI Wrappers	436
Nikto	436
FTP Security	437
Active and Passive FTP	437
S/FTP	438
Secure Copy	439
Blind FTP/Anonymous	439
FTP Sharing and Vulnerabilities	440
Packet Sniffing FTP Transmissions	441
Directory Services and LDAP Security	441
LDAP	442
LDAP Directories	443
Organizational Units	443
Objects, Attributes and the Schema	444
Securing LDAP	445
Summary	448
Solutions Fast Track	448
Frequently Asked Questions	451
Index	453

Introduction to Web Application Hacking

Solutions in this chapter:

- What is a Web Application?
- How Does the Application Work?
- The History of Web Application Hacking and Evolution of Tools
- Modern Web Application Hacking Methodology and Tools
- Automated Tools: What they are good at and what they aren't
- A Brief Tutorial on how to use WebScarab

☒ Summary

Introduction

What is hacking? To me, the act of hacking is the tinkering, studying, analyzing, learning, exploring and experimenting. Not just computers, but anything. One of the great outcomes of this activity is discovering ways to make the object of your attention bend to your will for your benefit, under your control. An accountant who discovers a new tax loophole can be considered a hacker. Through out time tinkerers, thinkers, scholars and scientists who created things like the wheel, lever and fulcrum, capacitor, inductor, polio vaccine, the light bulb, batteries, phone, radio, air plane, and of course the computer, in a sense, were all hackers. All of the individuals behind most every great invention had a relentless pursuit to bend the will of whatever force they could leverage to a desired outcome. Very few innovations were created by accident, and even if the result of an accident was the inspiration, a great degree of tinkering, studying, analyzing, learning, exploring and experimenting was most certainly necessary to obtain or perfect the desired goal. Most great innovations came from an almost unnatural amount of tinkering, studying, analyzing, learning, exploring and tinkering ... or hacking. The act of hacking when applied to computer security typically results in making the object of your desire (in this case, usually a computer) bend to your will. The act of hacking when applied to computers, just like anything else, requires tenacity, intense focus, attention to detail, keen observation, and the ability to cross reference a great deal of information, oh and thinking “outside of the box” definitely helps.

In this book, we aim to describe how to make a computer bend to your will by finding and exploiting vulnerabilities specifically in Web Applications. We will describe common security issues in web applications, tell you how to find them, describe how to exploit them, and then tell you how to fix them. We will also cover, how and why some hackers (the bad guys) will try to exploit these vulnerabilities to achieve their own end. We will also try to explain how to detect if hackers are actively trying to exploit vulnerabilities in your own web applications.

In this book the examples will be teaching how to find vulnerabilities using “Black Box” methods (where the user does not have the source code, documentation or web server logs for the application). Once the black box methods have been described, source code and audit trail methods of discovering vulnerabilities will also be mentioned.

It should also be noted that it is not possible to document every possible scenario you will run into and fit all of that information into one moderately sized book, but we will try to be as broad and encompassing as possible. Also this book more aims to teach the reader how to fish by defining a methodology of web application hacking and then describes how to find common vulnerabilities using those methodologies.

To begin our lessons in web application hacking it is important that you (the reader) are familiar with what a web application is and how one works. In this chapter, the next few sections describe how a web application works and the later sections in this chapter describe web hacking methodologies.

Web Application Architecture Components

Basically a web application is broken up into several components. These components are a web server, the application content that resides on the web server, and typically there a backend data store that the application accesses and interfaces with. This is a description of a very basic application. Most of the examples in this book will be based on this model. No matter how complex a Web application architecture is, i.e. if there is a high availability reverse proxy architecture with replicated databases on the backend, application firewalls, etc., the basic components are the same.

The following components makeup the web application architecture:

- The Web Server
- The Application Content
- The Datastore

The Web Server

The Web Server is a service that runs on the computer the serves up web content. This service typically listens on port 80 (http) or port 443 (https), although often times web servers will run on non standard ports. Microsoft's Internet Information Server and Apache are examples of web servers. It should be noted that sometimes there will be a "middleware" server, or web applications that will access other web or network applications, and we will discuss middleware servers in future chapters.

Most web servers communicate using the Hyper Text Transfer Protocol (HTTP) context and requests are prefixed with "http://". For more information about HTTP please refer to RFC 2616 (HTTP 1.1 Specification) and RFC 1945 (HTTP 1.0 Specification).

Ideally web applications will run on Secure Socket Layer (SSL) web servers. These will be accessed using the Hyper Text Transfer Protocol Secure (HTTPS) context and requests will be prefixed with "https://". For more information about HTTP please refer to RFC 2818 (HTTP Over TLS Specification). (We'll cover hardening a Web server in Chapter 7.)

The Application Content

The Application Content is an interactive program that takes web requests and uses parameters sent by the web browser, to perform certain functions. The Application Content resides on the web server. Application Content is not static content but rather programming logic content, or content that will perform different actions based on parameters sent from the client. The way the programs are executed or interpreted vary greatly. For example with PHP an interpreter is embedded in the web server binary, and interactive PHP scripts are then interpreted by the web server itself. With a Common Gateway Interface (CGI) a program resides in a special directory of the web server and