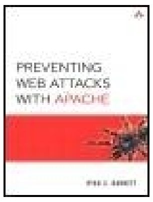


Preventing Web Attacks with Apache
By Ryan C. Barnett
....
Publisher: **Addison Wesley Professional**
Pub Date: **January 27, 2006**
Print ISBN-10: **0-321-32128-6**
Print ISBN-13: **978-0-321-32128-2**
Pages: **624**

Table of Contents | Index

Overview

"Ryan Barnett has raised the bar in terms of running Apache securely. If you run Apache, stop right now and leaf through this book; you need this information."

Stephen Northcutt, The SANS Institute

The only end-to-end guide to securing Apache Web servers and Web applications

Apache can be hacked. As companies have improved perimeter security, hackers have increasingly focused on attacking Apache Web servers and Web applications. Firewalls and SSL won't protect you: you must systematically harden your Web application environment. Preventing Web Attacks with Apache brings together all the information you'll need to do that: step-by-step guidance, hands-on examples, and tested configuration files.

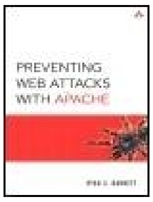
Building on his groundbreaking SANS presentations on Apache security, Ryan C. Barnett reveals why your Web servers represent such a compelling target, how significant exploits are performed, and how they can be defended against. Exploits discussed include: buffer overflows, denial of service, attacks on vulnerable scripts and programs, credential sniffing and spoofing, client parameter manipulation, brute force attacks, web defacements, and more.

Barnett introduces the Center for Internet Security Apache Benchmarks, a set of best-practice Apache security configuration actions and settings he helped to create. He addresses issues related to IT processes and your underlying OS; Apache downloading, installation, and configuration; application hardening; monitoring, and more. He also presents a chapter-length case study using actual Web attack logs and data captured "in the wild."

For every sysadmin, Web professional, and security specialist responsible for Apache or Web application security.

With this book, you will learn to

- Address the OS-related flaws most likely to compromise Web server security
- Perform security-related tasks needed to safely download, configure, and install Apache
- Lock down your Apache httpd.conf file and install essential Apache security modules
- Test security with the CIS Apache Benchmark Scoring Tool
- Use the WASC Web Security Threat Classification to identify and mitigate application threats
- Test Apache mitigation settings against the Buggy Bank Web application
- Analyze an Open Web Proxy Honeypot to gather crucial intelligence about attackers
- Master advanced techniques for detecting and preventing intrusions



Preventing Web Attacks with Apache
By Ryan C. Barnett
....
Publisher: **Addison Wesley Professional**
Pub Date: **January 27, 2006**
Print ISBN-10: **0-321-32128-6**
Print ISBN-13: **978-0-321-32128-2**
Pages: **624**

Table of Contents | Index

Copyright

The article "How we defaced apache.org" in Chapter 2 is reprinted courtesy of Peter van Dijk and Frank van Vliet.

The excerpt of the Open Proxy in Chapter 10 was reprinted courtesy of LURHQ's Threat Intelligence Group (http://www.lurhq.com).

The section on DNS Cache Poisoning in Chapter 11 was reprinted courtesy of the Internet Storm Center.

Appendix A is reprinted courtesy of the Web Application Security Group, and covered under the Creative Commons License, which can be found at http://creativecommons.org/licenses/by/2.5/.

At the time of publication, Ryan Barnett serves as the volunteer leader of the Center for Internet Security (CIS) consensus team that is responsible for development and maintenance of the CIS Benchmark for Apache Web Server. While some of the security configuration recommendations described in this book are based, in part, on those defined in the CIS Benchmark for Apache Web Server, version 1.0, this book and its contents represent the opinion and recommendations of the author and not the Center for Internet Security. The CIS Benchmark for Apache Web Server document and the CIS Scoring Tool for Apache are available free of charge from the CIS web site at www.cisecurity.org.

Visit us on the web: www.awprofessional.com

Library of Congress Cataloging-in-Publication Data

Barnett, Ryan C.
Preventing web attacks with Apache / Ryan C. Barnett.
p. cm.
ISBN 0-321-32128-6 (pbk. : alk. paper) 1. Apache (Computer file : Apache Group) 2. Web sitesSecurity measuresC
programs. 3. Web serversComputer programs. I. Title.
TK5105.8885.A63B37 2005
005.8dc22
2005030960

Text printed in the United States on recycled paper at R.R. Donnelley in Crawfordsville, Indiana.

First printing, February 2006

Dedication

THIS BOOK IS DEDICATED TO MY GRANDFATHER, DR. VERLE GLENN CRAGO, WHO HAS ALWAYS BE
TREMENDOUS SOURCE OF ENCOURAGEMENT THROUGH THE YEARS. WITHOUT HIM, NOT ONLY WO
THIS BOOK NOT HAVE BEEN POSSIBLE, I WOULDN'T BE WHERE I AM TODAY. THANK YOU, GRANDP

About the Author

Ryan C. Barnett is a chief security officer for EDS. He currently leads both Operations Security and Incident Response Teams for a government bureau in Washington, DC. In addition to his nine-to-five job, Ryan is also a faculty member for the SANS Institute, where his duties include instructor/courseware developer for Apache Security, Top 20 Vulnerabilities team member, and local mentor for the SANS Track 4, "Hacker Techniques, Exploits, and Incident Handling," course. He holds six SANS Global Information Assurance Certifications (GIAC): Intrusion Analyst (GCIA), Systems and Network Auditor (GSNA), Forensic Analyst (GCFA), Incident Handler (GCIH), Unix Security Administrator (GCUX), and Security Essentials (GSEC). In addition to the SANS Institute, he is also the team lead for the Center for Internet Security Apache Benchmark Project and a member of the Web Application Security Consortium.

Foreword

Ryan Barnett recently asked if I'd write the foreword to his book. I was delighted to even be considered because Ryan is an exceptional security professional and the honor could have easily gone to anyone in the industry. Ryan has a background as someone who actively defends government web sites. He's the person who led the effort to create the Apache Benchmark standard for the Center for Internet Security (CIS). He's a co-author of the Web Security Threat Classification for the Web Application Security Consortium (WASC), and has more certifications than I knew existed. Ryan is also a SANS Instructor for Apache Security. There's quite a bit more, but suffice it to say Ryan has to be one of the most-qualified experts to write Preventing Web Attacks with Apache.

A foreword is an opportunity to express why a particular topic is important and describe what role the information plays in a broader context. Even though I've been part of the web application security field for a really long time (back before there was a term to describe what we do), more research was in order. I fired up Firefox and headed on over to Google for some investigation. Netcraft, the WASC, the CIS, the Open Source Vulnerability Database (OSVDB), SecurityFocus, and Wikipedia are incredible resources for collecting security information. While I was taking notes and saving bookmarks, it suddenly occurred to me that during my research, I must have crossed paths with hundreds of Apache web servers without realizing it. What a perfect way to describe the importance of Apache security!

According to Netcraft's Web Server Survey (September 2005), Apache accounts for roughly 70 percent of the Internet's web servers. Through our tiny browser window, it's difficult to imagine the global hum of 72 million web servers, the keyboard chatter of over 800 million international netizens, wading through a sea of 8 billion web pages. Apache is a fundamental part of our daily online livesso much so, it's become a transparent artifact in the architecture of the web. When we shop for books, reserve plane tickets, read the news, check our bank account, bid in an auction, or do anything else with a web browser, the odds are there's an Apache web server involved. How's that for important?

The web has become bigger and more powerful than we ever imagined. 24x7x365, web sites carry out mission-critical business processes, exchanging even the most sensitive forms of information including names, addresses, phone numbers, social security numbers, financial records, medical history, birth dates, business contacts, and more. Web sites may also supply access to source code, intellectual property, customer lists, payroll data, HR data, routers, and servers. If a particular computer system or business process isn't web-enabled today, bet that it will be tomorrow. Anything a cyber-criminal would ever want is available somewhere on a web site. With all the great things we can do on the web, one must temper the benefits with the risk that any information available on or behind a web site is also a target for identify theft, industrial espionage, extortion, and fraud. It should come as no surprise that the attack trends we're witnessing are migrating from the network layer up to the web application layer.

Here's where things get interesting and scary at the same time. Firewalls, anti-virus scanners, and Secure Sockets Layer (SSL) do not help secure a web site. Let me say that again. Firewalls, anti-virus scanners, and SSL do not help secure a web site. When you visit any web site, we don't see any of these things because they functionally don't exist at the web layer. On the web, there's nothing standing between a hacker, your web server, your web applications, and your database. With something as pervasive as Apache, the knowledge of how to prevent web attacks is vital.

A martial arts black belt is a suitable analogy. It might take someone years to acquire the knowledge required to proficiently react to a given security scenario. Both Ryan and myself have experience defending extremely large and public web-enabled systems. We've witnessed the sophisticated and voluminous attacks that inundate our web servers. From Brute-Force or Cross-Site Scripting to Denial of Service or SQL Injection and Worms, the attacks are varied and pervasive. A single day of monitoring web server log files is enough to appreciate how much skill is required to thwart the ever-growing security threats.

Ryan has done a remarkable job combining his years of personal experience with the collective knowledge of a community of experts. Readers will be well served by this material for as long as there are web servers. I'll finish up with a famous quote I feel captures the essence of Preventing Web Attacks with Apache.

> "The significant problems we face cannot be solved at the same level of thinking we were at when we created them."
>
> Albert Einstein (1879-1955)

We must be diligent, we must keep learning, we will prevail.

Jeremiah Grossman

Founder and CTO of WhiteHat Security

Cofounder of the Web Application Security Consortium (WASC)

September, 2005

Acknowledgments

First and foremost, I want to thank my wonderful wife Linda. There were many nights that I spent sequestered away in my office writing this book, and she was always understanding and supportive. She is my best friend and consigliere. I love you.

Next, I would like to thank my brother Scott. He was the one who convinced me to move into the computer field. Not only did he offer me moral support when I was making this career change, but he and his family actually took me in and let me live with them for a short time when I moved to the DC area. Thanks go to both Scott and his wife Jen. I don't know if I will ever be able to repay you for your kindness in my time of need.

I would also like to thank the guys at WASC. You all are incredibly intelligent men and your passion for web security is inspiring. Special thanks go to Jeremiah Grossman and Robert Auger for founding WASC and for just being two really cool guys to work with.

Thanks also go to Ivan Ristic for creating Mod_Security and for putting up with all of my crazy "What if..." questions. You are truly pushing the envelope with your web security expertise.

A very special thank-you go to the folks at the SANS Institute. To Stephen Northcutt, Ed Skoudis, and Zoe Dias, thank you all for your support. You are truly helping to raise the bar for information security and getting the message out to the masses.

To John Banghart and the folks at the Center for Internet Security (CIS), you are providing an outstanding service, and I hope that you keep up the good work.

Introduction

Why This Book?

There were a number of motivating factors that moved me to write this book. First and foremost was the fact that the number of web server and web application attacks are rising quickly. A report from security firm Zone-H (27/26 April 2005) finds that web server attacks and web site defacements increased by 36 percent in the last year, from 251,000 in 2003 to 392,545 in 2004. According to the report, 2,500 web servers are successfully attacked every day.

> "Defacement is just one option for an attacker," said Roberto Preatoni, the founder of Zone-H. "In most circumstances the techniques used by defacers are the same used by serious criminals to cause damage. The data on cybercrime provides information on the evolution of trends and [this] allows system administrators to close the security holes that are used."

These types of news reports seem endless. The purpose of presenting this example article data in this book is not to spread F.U.D. (Fear, Uncertainty, and Doubt) with regards to web security, but rather to adequately depict the fact that attackers are targeting web applications more and more. In response to organizations beefing up their network perimeters defenses, attackers have moved the targets of their attacks to the application layer. There are a number of resources, beyond news reports, where this increased targeting of web attacks may be identified.

Symantec's Internet Security Threat Report

Another example report providing evidence of this concept is presented in Symantec's Internet Security Threat Report. This report is released quarterly, and the March 2005 release provided the following information:

> Increase in Attacks Against Web Applications: Web applications are popular targets because they enjoy widespread deployment and can allow attackers to circumvent traditional perimeter security measures such as firewalls. They are a serious security concern because they may allow attackers access to confidential information without having to compromise individual servers. Nearly 48 percent of all vulnerabilities documented between July 1 and Dec. 31, 2004 were web application vulnerabilities, a significant increase from the 39 percent documented in the previous six-month period.

SANS @RISK: The Consensus Security Vulnerability Report

This is a weekly report sent out to subscribers of the SANS Institute's newsletter. The letter outlines all of the new vulnerabilities and exploits that were released in the previous week. An online archive of past newsletters is available from the SANS web site (www.sans.org/newsletters/risk). The vulnerabilities are broken out into various categories such as Windows, third-party apps, Unix, and web applications. I have been monitoring these reports for quite awhile and I found that the Web Application section consistently has the most number of vulnerabilities. Here is an example listing from one of the reports:

```
05.36.27 - CVE: Not Available
Platform: Web Application
Title: myBloggie login.php SQL Injection
Description: myBloggie is a weblog application. It is vulnerable to an SQL injection
issue due to a failure in the application to properly sanitize user-supplied input to
the "login.php" script before using it in an SQL query. An attacker could exploit this
issue to pass malicious input to database queries, resulting in compromise of the
application. MyBloggie versions 2.1.1 to 2.1.3 are vulnerable.
Ref: http://mywebland.com/forums/showtopic.php?t=399
```

SANS Institute's Internet Storm Center

The Internet Storm Center (http://isc.sans.org) releases periodic threat updates. According to Dr. Johannes Ullrich, CTO of the SANS Institute's Internet Storm Center, "Once you move past the worms targeting buffer overflows or weak passwords and examine the manual attacks, classically called hacking, web application attacks account for a significant portion of the activity."

SANS Class: Web Intrusion Detection and Prevention with Apache

I am the courseware developer and instructor for a two-day class on Apache security and web intrusion detection. I created this course due in large part to the flood of emails sent by the public to SANS with questions about Apache and web application security. There is a large population of Apache users who need in-depth details on how to address security issues that arise when trying to deploy Apache as a front-end web server for a web application. These users were having difficulty finding a consolidated resource for this information. They needed practical, hands-on examples of how to fix these web security issues. The sheer number of requests for this class and the overwhelmingly positive responses to the class also played a key role in prompting me to write this book.

Apache's Market Share

According to the Netcraft Web Server Survey (http://news.netcraft.com/archives/web_server_survey.html), roughly 69 percent of all web servers on the Internet run Apache, which translates to over 49 million servers. These numbers present a huge pool of potential targets for web attackers. Not all of these servers are being utilized for eCommerce functions; however, Apache does play a rather significant role in many multi-tiered web application deployments.

Apache's Role in Common Web Application Architectures

A common configuration for today's complex web applications is to distribute the architecture into separate tiers (or servers) with specific roles in the overall flow of the applications. There are two distinct phases in which Apache provides a key role.

Presentation TierApache Running as a Reverse Proxy for Back-End Web Applications

In the Presentation Tier setting, Apache is often utilized as a reverse proxy. This architecture provides increased security by allowing Apache to be deployed within a DMZ and allowing it to interact with web clients. Apache will then apply security checks, translate the web request, and forward it on to the destination application on the internal network. We will discuss the benefits and challenges of using a reverse proxy in a later chapter.

Application TierApache Is Integrated into Many Commercial Web Application Front-Ends

Apache is an extremely popular choice for commercial vendors who need to add a web front-end to their custom applications. Because Apache is open source and free, it is relatively easy for vendors to take the Apache code, customize it to work with their application, and then ship it out to customers. Another technique used by vendors is to create a web application that will easily plug directly into Apache, such as through a module. Some popular examples of this process are the following application servers:

- Oracle 9iAS
- IBM WebSphere
- BEA WebLogic
- Allaire ColdFusion
- Apple WebObjects

Due to the fact that Apache is so heavily utilized in common web architectures, it becomes evident that in order to secure your web environment, you have to secure Apache first and then move on to specific application issues. Can Apache be hacked? Absolutely. A "default" installation of any application is a bad idea, and Apache is no exception.

What This Book Will Cover

At a high level, we will be covering the following topics in the book:

- Chapter 1, "Web Insecurity Contributing Factors."

  This chapter will set the scene with all of the different factors that impact the security of most web environments. Many of these issues are not directly technical in nature but rather are by-products of most organization's processes.
- Chapter 2, "CIS Apache Benchmark."

  This chapter outlines a handful of OS-related issues that should be addressed as they directly impact the overall security of the web server.
- Chapter 3, "Downloading and Installing Apache."

  There are a surprising number of security-related tasks associated with properly downloading, configuring, and installing the Apache software onto a server.
- Chapter 4, "Configuring the httpd.conf File."

  This chapter discusses the various changes to the httpd.conf file that are outlined in the CIS Apache Benchmark document.
- Chapter 5, "Essential Security Modules for Apache."

There are various Apache security modules that should be installed for maximum benefit. This chapter highlights the modules and discusses some high-level configurations.

- Chapter 6, "Using the Center for Internet Security Apache Benchmark Scoring Tool."

  After applying the CIS Apache Benchmark settings, the scoring tool should be run to verify the settings. This chapter shows you how to run the tool and to interpret the results.
- Chapter 7, "Mitigating the WASC Web Security Threat Classification with Apache."

  This chapter shows you how to protect your web applications from the WASC Threat Classification items by using Apache.
- Chapter 8, "Protecting a Flawed Web Application: Buggy Bank."

  Building upon the previous chapter, this chapter uses an example web application called Buggy Bank to demonstrate how to use Apache to mitigate common web vulnerabilities.
- Chapter 9, "Prevention and Countermeasures."

  This chapter outlines the many different web intrusion detection and prevention concepts. We discuss issues such as IDS evasion, identifying probes, and creating custom web security filters.
- Chapter 10, "Open Web Proxy Honeypot."

  The chapter presents data gathered from the Honeynet Project's Scan of the Month Challenge that the author sponsored. During the project, a specially configured Apache open proxy server honeypot was deployed on the Internet for one week and the resulting logs are analyzed.
- Chapter 11, "Putting It All Together."

  This final chapter summarizes what was covered in the book and also highlights some other web security issues of which the reader should be aware.
- Appendix A, "WASC Glossary."

  This appendix lists common web security terminology and their definitions. We would like to thank the Consortium for granting us permission to reprint this document.
- Appendix B, "Apache Module Listing."

  This appendix lists the Apache modules that are commonly used and provides security-relevant information and recommendations for their use.
- Appendix C, "Example httpd.conf File."

  An example httpd.conf file is presented as a template for review when discussing the various security settings throughout the book.

What This Book Will Not Cover

I would be remiss if I didn't outline some of the topics that this book will not cover due to scope restraints. Although we will not be covering these specific topics in the book, it does not mean that they are not important to web security. On the contrary! These issues are vital to the successful protection of all web deployments; however, they require such great detail that entire books have been created to cover them. Here is a listing of some topics that are beyond the scope of this book.

Operating System (OS) Security

Chapter 2 does highlight some specific OS security-related information that directly impacts the overall security of the web server. Beyond those specific settings, there are vast amounts of information concerning proper OS lockdown procedures that are beyond the scope of this book.

Secure Coding Strategies

To be honest, the vast amount of web application vulnerabilities could be addressed if the source code was properly written. Writing secure code is a complex task as it is different for each language being used.

Web Scripting Languages (Such as PERL, PHP, and Java) Security Issues

As mentioned in the previous section, each individual scripting language has its own security functions. These features help to validate and sanitize user input and help to guard against the application using unsafe processes. Once you know what scripting language you will be using, you should research that language's specific security mechanisms.

Conventions Used Throughout This Book

I am a strong proponent of the philosophy that in order to be effective at protecting web applications, you must understand the methodology used by web attackers. If you don't know how to attack, then how can you defend? It is the age-old warfare axiom of "Know Your Enemy." It is for this reason that the vast majority of web attacks are discussed from both the attacker and defender points of view.

Security Documents Used Throughout this Book

Before we discuss the particulars of advanced web application security and intrusion detection with Apache, we first need to address the steps to secure Apache itself. How can we expect to protect a robust web application with an Apache front end, if we don't first secure Apache? In the next few chapters, we will be discussing the steps needed to properly secure an Apache installation. During these chapters, we will be utilizing some security documents and tools from two security organizations. The two organizations and documents are listed next.

The Center for Internet Security's Apache Benchmark Document and Scoring Tool

I am the CIS Apache Benchmark project leader and created the initial draft document. While we will be covering the Apache Benchmark in this book, I will also be making some of my own comments and expressing some of my own beliefs with regards to Apache security. Due to this separation between the official CIS Apache Benchmark document and the information I present within this book, I am compelled to provide the following CIS disclaimer:

> "At the time of publication, Ryan Barnett serves as the volunteer leader of the Center for Internet Security (CIS) consensus team that is responsible for development and maintenance of the CIS Benchmark for Apache Web Server. While some of the security configuration

> recommendations described in this book are based, in part, on those defined in the CIS Benchmark for Apache Web Server, version 1.0, this book and its contents represents the opinion and recommendations of the author and not the Center for Internet Security. The CIS Benchmark for Apache Web Server document and the CIS Scoring Tool for Apache are available free of charge from the CIS web site at www.cisecurity.org."

This book discusses most of the benchmark settings, and in greater detail than in the Benchmark. I wanted to include more examples and background information in the Benchmark document itself; however, it was agreed upon that this was not the right format to cover this information. The benchmarks are supposed to be concise, quickly applied security settings. It was therefore decided that this book would be a better platform to discuss these security settings in more detail.

The Web Application Security Consortium's (WASC) Web Security Threat Classification and Glossary Documents

I am a contributing member of WASC and assisted with the creation of the Threat Classification document. We will be discussing WASC's mission statement, purpose, and the threat classification details later in Chapter 7.

The Center for Internet Security's Apache Benchmark

You are getting ready to install and configure your Apache web server; however, you have the following questions:

- What do I need to do to make my systems sufficiently reliable and secure, based on my organization's assessment of the costs of security measures versus the value of operating reliable systems for my customers?
- How much is enough? What method can I use to determine the minimum level of due care based on best-practice benchmarks needed to reduce my enterprise risk to an acceptable level?
- Whom can I trust to tell me what I need to do and to help me protect my systems and networks?

Answering these questions is not an easy task. You will most likely have to scour the Internet searching for various security documents outlining steps to take to secure your installation. You check out your vendor's web site, then at CERT, and finally start searching various news groups. Needless to say, this can become quite frustrating with the amount of time required to obtain this information, coupled with conflicting recommended settings often identified. Where can someone find this vital security information? It was out of this type of environment that the Center for Internet Security (CIS) was born (www.cisecurity.org).

What Is the CIS?

The Center for Internet Security's mission is to help organizations around the world effectively manage the risks related to information security. CIS provides methods and tools to improve, measure, monitor, and compare the security status of Internet-connected systems and appliances, plus those of business partners.

The Center strives to reduce the frequency of failures and attacks, and the losses that arise from them. The mission of the Center is to help organizations around the world effectively manage the organizational risks related to information security by providing them with methods and tools to improve, measure, monitor, and compare the security status of their own Internet-connected systems and appliances plus those of their business partners.

The Center is not tied to any proprietary product or service. It manages a consensus process whereby members will articulate security threats that concern them, followed by prioritization and development of benchmarks and accreditation methodologies to reduce the threats of concern to members. The consensus process is already in use and has proved viable in creating widely adopted Internet security practices.

What Are the CIS Benchmarks?

CIS Benchmarks enumerate security configuration settings and actions that "harden" your systems. They are unique, not because the settings and actions are unknown to any security specialist, but because consensus among hundreds of security professionals worldwide has defined these particular configurations.

CIS Level-I Benchmarksthe Prudent Level of Minimum Due Care

Level-I Benchmark settings/actions meet the following criteria:

- System administrators with any level of security knowledge and experience can understand and perform the specified actions.
- The action is unlikely to cause an interruption of service to the operating system or the applications that run on it.
- The actions can be automatically monitored, and the configuration verified, by Scoring Tools that are available from the Center or by CIS-certified Scoring Tools.

Many organizations running the CIS Scoring Tools report that compliance with a CIS "Level-I" benchmark produces substantial improvement in security for their systems connected to the Internet.

CIS Level-II BenchmarksPrudent Security Beyond the Minimum Level

Level II security configurations vary depending on network architecture and server function. These are of greatest value to system administrators who have sufficient security knowledge to apply them with consideration to the operating systems and applications running in their particular environments. Generally speaking, the Level II settings address higher risk security issues and are more likely to cause disruption to service if not implemented correctly.

What Are the Scoring Tools?

The CIS Scoring Tools provide a quick and easy way to evaluate systems and networks, comparing their security configurations against the CIS Benchmarks. They automatically create reports that guide users and system administrators to secure both new installations and production systems. The tool is also effective for monitoring systems to assure that security settings continuously conform to CIS Benchmark configurations. We will be discussing the CIS Apache Benchmark Scoring Tool and its use in Chapter 6.

Summary

As you can see from the data presented in this chapter, there is a real need for this type of book. Web attacks are at an all-time high and Apache is being used in a vast amount of these web environments. This information should have adequately brought attention to the fact that web applications are ripe targets for attackers. In order to re-enforce these statistics, many real "in the wild" web attacks will be presented throughout the book. Now that we have covered the basis for how the book will work, let's jump right in with some information that will set the scene for current process issues that are affecting most web deployments.

1. Web Insecurity Contributing Factors

A Typical Morning

"Morning, Stan."

"Morning, Ryan. Hey, it looks like it was a pretty quiet night, as I didn't receive any pager alerts. After looking at some of the IDS logs, it is all normal script kiddie probes and such."

"Good, let's hope it stays that way," I said as I laid down my laptop bag in my cubicle. I slid into my chair, locked the keyboard tray into position, and gazed up at the computer monitor as it began to glow to life. It was now time for my morning ritual, so I turned on some Radiohead, took a sip of my coffee, and began to read my email.

As my eyes quickly glanced down the multitude of emails, one subject line in particular caught my eyeMultiple Oracle vulnerabilities discovered. This was an email from one of the many vulnerability alerts mail lists to which I have subscribed. Only having the information provided in the subject line was not enough information to confirm or dismiss this matter, so I hesitantly double-clicked on it to read the entire message.

I don't know about you, but every time I receive a vulnerability email alert, I get a little twinge in my stomach that reminds me of elementary school when the teacher would call out the name of the student who had to stay after school for a parent-teacher conference. That is a meeting where you did not want to hear your name called for the invitee list. These vulnerability alerts are similar in that you are crossing your fingers that the "Systems Affected" list will not include your software. I, therefore, anxiously opened the email to review some of the details.

```
Researchers at NGS Software have discovered multiple critical vulnerabilities in
Oracle Database Server and Oracle Application Server. Versions affected include:

Oracle Database 10g Release 1, version 10.1.0.2
Oracle9i Database Server Release 2, versions 9.2.0.4 and 9.2.0.5
Oracle9i Database Server Release 1, versions 9.0.1.4, 9.0.1.5, and 9.0.4
Oracle8i Database Server Release 3, version 8.1.7.4
Oracle Application Server 10g (9.0.4), versions 9.0.4.0 and 9.0.4.1
Oracle9i Application Server Release 2, versions 9.0.2.3 and 9.0.3.1
Oracle9i Application Server Release 1, version 1.0.2.2
```

```
The vulnerabilities range from buffer overflow issues, PL/SQL Injection, trigger
abuse, character set conversion bugs, and denial of service.
```

Shoot. This affects a number of our financial systems because they are utilizing Oracle 9iAS as the web server front end for their applications. These web applications are Internet accessible and are mission critical. Man, there goes the quiet morning.

"Hey Stan," I called out over the cubicle wall that separates our desks.

"Yo," he responded in typical fashion.

"Have you seen that alert email about Oracle 9iAS?" I asked almost rhetorically, already knowing the answer.

"Reading it now," he said with a sigh.

I continued reading the vulnerability alert email, hoping for some sign of good news, but to no avail. The more I read, the more I realized that this was going to get ugly. There were multiple Oracle vulnerabilities identified, and they were wide ranging from buffer overflows and denial of service to command execution exposures.

Just then my phone rang. "OPS Security, this is Ryan," I answered with my standard greeting. It was my boss, and apparently she had heard of these Oracle vulnerabilities as well.

"Ryan, do any of these Oracle vulnerabilities affect us?" she asked, fearing the answer.

"Yep, all of them," I replied matter of factly.

She continued, "What do you see as our mitigation strategy?"

I paused for a moment as I reviewed the vulnerabilities in my head and then responded with, "We will have to reconfigure some Apache Oracle 9iAS settings and figure out if we can create some security filters on our DMZ proxy server to mitigate some of these vulnerabilities. There are also some new patches that Oracle has released." That was met with a long pause.

"Hmm, you know that we cannot patch all of those servers, right?" she finally stated. You see, we were running many custom web applications that were developed by the previous contractors, and historically, any patches that we had applied had broken some form of functionality within the applications. Management finally decided that for the time being, no patches would be applied to these applications. This, of course, made my job much more challenging.

"Yeah, I know," I said as I rolled my eyes. Just then I heard Stan chuckling in the next cube. He always had an uncanny knack for comprehending an entire phone conversation by hearing only my comments. He seemed quite amused by my quandary.

"I need you to come up with a plan for mitigating these Oracle 9iAS issues. We have a meeting with the ISSO in a half hour and we need you to present your recommendations," she ordered as she hung up the phone.

"Great, a whole 30 minutes," I thought. "That isn't even enough time to finish my coffee."

Why Web Security Is Important

The scenario outlined here is not unique. I have spoken with countless web security colleagues and they have all related similar stories; simply substitute the names of the people and the vulnerability numbers, and the stories are pretty much interchangeable. There is a never-ending flood of vulnerabilities released, and keeping up with them is quite literally a full-time job. To make matters worse, the bad guys are targeting web applications more often due to their potential value (business transactions) coupled with their lack of proper protections. In order to have a fighting chance against these attacks, web security professionals must have the skills and techniques to efficiently analyze the current threats and implement appropriate security mitigations.

This book addresses the following questions: How do you secure your web deployments? Do you know what methodology web attackers utilize? Do you know how to identify web attacks either actively targeting your system or by reviewing historical system logs? Do you know the tricks that web attackers may employ in order to evade detection? Can you prevent these attacks? What are the limitations of your web defenses? Do you have adequate logging to assist with incident response? By the end of this book, you will be able to answer these questions.

Before we jump into the nitty-gritty details of web intrusion detection and prevention and the tools that we will use, we need to discuss some cultural, political, and technical contributing factors that affect the security of many web deployments. Providing solutions to all of these factors is beyond the scope of the book; however, I feel that it is prudent to discuss them at a high level to help facilitate a broader understanding of the landscape. Sometimes it is helpful for security practitioners to take off our blinders and look at our environment from different perspectives.

Web Insecurity Contributing Factors

There exists an all-too-common trend in today's eCommerce world. Companies are growing increasingly reliant upon the functionality provided to them by web-based applications; however, these same applications are not becoming more secure. Web applications are being targeted more and more by attackers, and yet most businesses are not adequately developing and deploying secure web applications.

Why is this? A multitude of different factors attribute to the current state of web security. The issues discussed next all play a role at some level. Odds are that someone working in the security field has faced some, if not all, of the following problems.

Managerial/Procedural Issues

Management and the Bottom Line

The most influential player is the Almighty Dollar. Businesses are forced to speed up the development life cycle of web applications in order to have a competitive market advantage. Unfortunately, the overriding mindset of most businesses is to be the "First to market with an application that works." The repercussions of this doctrine are that web applications are built to address a business requirement, and therefore issues such as secure coding, security validation of web logic, and secure network infrastructure deployment are often reduced, if not ignored all together.

As the comedian Dennis Miller often quipped on his HBO television show, "I don't want to get off on a rant here, but...," I strongly believe that the current state of web application security is directly impacted by Management not being held accountable for deploying insecure applications. Management is always responsible when an organization successfully attains their earning projections, and they enjoy the praise and bonuses that accompany these achievements. On the other hand, they are seldom held accountable when an attacker steals customer credit card data from a compromised web site.

Things are starting to change with the emergence of many different government regulations such as the Federal Information Security Management Act (FISMA), the Health Information Privacy and Protection Act (HIPPA), and the Gramm-Leach-Bliley Act (GLB). Each of these regulations focuses on different aspects of information security and confidentiality; however, they all have the common thread of holding Management accountable for the security of all of their information technologies. Until the time when Management places adequate time, money, and resources into forcing security into the development life cycle, security practitioners will continue to fight an uphill battle in securing web applications.

Selling Loaded Guns

Coming in at a close second place on the web insecurity contributor list are the commercial software vendors. Vendors are battling the same issues as Managementthey need their software to work, and it must be the first to market. This directly leads to the following outcomes: There are a higher number of flaws in the code due to decreased timelines, and most security features are turned off by default for ease of use and functionality. This is a lethal combination.

Flaws in software code are something that we will most likely never get rid of, due to the fact that humans create it. After all, to err is human. While this is a reality, the sheer numbers of vulnerabilities found suggest a lack of due diligence. It also seems that Independent Verification and Validation (IV&V) processes are severely lacking; otherwise, more of these flaws would be found internally.

As the title of this section suggests, selling software with security features turned off by default is like selling guns that are fully loaded at the time of purchase. Although this may be more convenient for the buyer, it is obviously quite dangerous. In order to have the gun work appropriately, the buyer must load the gun with ammunition. This demonstrates both intent to use the gun and understanding of how the ammunition and gun work together to function properly. In the same vein, if software vendors sold software with security features enabled by default, then buyers would have to read the manuals to gain the information necessary to disable the security features. By disabling these features, responsibility for security is then transferred from the seller to the buyer. Unfortunately, software vendors have not embraced this concept.

The issue of selling insecure software has heated up recently with organizations such as the Center for Internet Security (CIS) leading the fight to force vendors to meet minimum-security standards. CIS has promoted the concept that the collective buying power of customers should be leveraged to demand that companies offer securely configured products by default. A promising sign for this movement was when PC giant Dell started utilizing the CIS Windows Security Benchmark document to secure their desktop and server products. We will discuss the CIS Benchmark projects in greater detail later in this book.

The Two-Minute Drill

Much in the same way that a quarterback is called upon to lead a game-winning drive in the final minutes of a football game, security teams are often limited to a very short amount of time to review applications just before they are deployed into production. The downside to this approach is two-fold.

First, by segmenting the involvement of security from the development process, its effectiveness is decreased. Correcting security issues is most effective during the development phase when there is adequate time and resources to fix the problem.

Second, in contrast to testing security during development, security testing performed immediately prior to