



WHITE PAPER

Common web application security problems and how to identify them

Part 10

Insecure configuration management

By

Lee J Lawson
Lead Penetration Tester, **dns**.

Introduction

This is final part of a series of papers designed to raise the level of security knowledge regarding common security flaws in web applications. All papers revolve around the top 10 list of web application security issues as defined by the Open Web Application Security Project (OWASP). The last issue described the vulnerability known as **Application denial of service**. This paper concentrates on **Insecure configuration management**, the identification and resolution of this flaw.

Here is a list of the most common security flaws at the time of writing taken directly from the OWASP site.

1. **Unvalidated input**
2. **Broken access control**
3. **Broken authentication and session management**
4. **Cross site scripting**
5. **Buffer overflows**
6. **Injection flaws**
7. **Improper error handling**
8. **Insecure storage**
9. **Application denial of service**
10. **Insecure configuration management**

It should be noted that this paper is not intended to be used as a replacement to professional penetration testing. Rather it is aimed toward raising the level of security knowledge in the community. Professional penetration testing requires years of experience to be able to apply worthwhile interpretation to results from testing tools and gain the skills required to perform manual assurance testing.

Insecure configuration management

Having a strong server configuration standard is critical to a secure web application. These servers have many configuration options that affect security and are not secure out of the box.

There is a significant gap between out of the box security and production environment security. Vendors produce software that is easy to use and quick to set up which does not necessarily lend itself to a secure baseline. Sometimes, significant work is required to reduce that gap.

The Problem

Web servers have many configuration options that add functionality or reduce it. Remember that the more complex a system is, the more likely that security flaws will be found.

the more complex a system is, the more likely that security flaws will be found

Purely installing web servers such as IIS or Apache is not enough to ensure that they have a secure configuration. Options such as directory listing, home directory access and anonymous web method execution all require changing prior to 'going live'. These weaknesses of course refer to the web server, but what about the application that resides on the server?

Applications, especially COTS (Commercial Off The Shelf) applications tend to have a number of configuration elements that expose them to hacker attacks. These problems include administrative interfaces accessible to the Internet, high levels of debugging left turned on and one of the biggest problems being default accounts still active – username=admin password=admin etc.

The main culprits for insecure configuration management vulnerabilities are:

- ⇒ **Administrative interfaces exposed to the Internet.**
- ⇒ **Redundant functions such as debugging enabled.**
- ⇒ **Redundant or default credentials.**
- ⇒ **Flawed hardening procedures.**

Identification

Identifying configuration problems with web servers and applications requires a systematic approach to ensure that all weaknesses have been discovered. The detailed description of a methodology is outside the scope of this document; however we will discuss some of the more common issues.

Google can be used quite effectively to search for all Internet accessible pages

Administrative interfaces exposed to the Internet

Google can be used quite effectively to search for all Internet accessible pages for any given site as the advanced search parameters can be filtered in many ways.

To search for all entries that are within any web site, use the site: search.

site:domainname.com

Other types of administrative interfaces such as Telnet on perimeter routers are also ill advised as all data to Telnet is unencrypted, potentially exposing highly sensitive data to interception.

Excessive web methods allowed

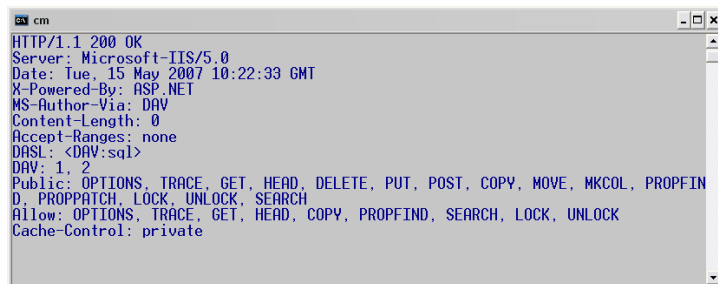
At a minimum, most web servers require the following methods or extensions to be enabled: HEAD, GET and POST. However, web servers tend to come out of the box with many more methods enabled, such as PUT and DELETE.

To identify if your web server has excessive web methods enabled, simply telnet into it on the HTTP port 80, this can be defined by adding a space and the desired port number after the IP address or DNS name.

telnet 10.1.1.112 80

Once connected, submit an OPTIONS command followed by two enter key presses, as follows:

OPTIONS / HTTP/1.0



```

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Tue, 15 May 2007 10:22:33 GMT
X-Powered-By: ASP.NET
MS-Author-Via: DAV
Content-Length: 0
Accept-Ranges: none
DAV: <DAV:sql>
DAV: 1, 2
Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPPATCH, LOCK, UNLOCK, SEARCH
Allow: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK, UNLOCK
Cache-Control: private

```

The screenshot shows an almost default installation of IIS 5.0, demonstrating the web methods that are enabled including the most dangerous such as DELETE and PUT. These options may allow, depending on further ACLs, an attacker to upload content to the web server or delete pages etc.

Local account enumeration

Certain versions of Apache web server allow an Internet user to identify local accounts on that server. This is due to the web server having access to home directories of users on the Linux/Unix file system. In actual fact, the server is denied access to the home directory, but then presents a 403 Forbidden message for a directory that exists and a 404 Not Found message for one that does not.

Due to the error messages, an attacker could identify an existing home directory and therefore local account.

Enter a tilde (~) and then the name of an account that you want to determine exists.

an attacker could identify an existing home directory and therefore local account

<http://www.victimwebsite.com/~root>

<http://www.victimwebsite.com/~lee>



If you receive a 403 Forbidden message, then the account exists.

These are just some examples of common configuration issues but do not demonstrate the scope of the problem.

Countermeasures

A methodical guide for hardening different types of servers should be written and updated to reflect new security issues.

the procedure should be written in consultation with the server support team and specialists in information security

The procedure should be written in consultation with the server support team and specialists in information security to ensure that all bases are covered. It should be clear, understandable, possibly with the use of diagrams and screenshots and repeatable by all members of staff responsible for this role.

The hardening procedure should be bespoke for each type of operating system (Windows 2003, Solaris 10 etc), each type of application (Microsoft IIS 6.0, Apache 2.3.29 etc) and each role the server plays (public website, private authenticated website etc), with the hardening guides working to support each other.

Organisations should consider a scheduled review of all hardening documentation to ensure that they stay relevant for the servers in use, the roles of the servers and the current security posture of the company.

Series summary

This is the final instalment of a 10 paper series detailing the most common security issues facing web applications.

If you have missed any of the papers, or want to talk to a consultant about this or any other security related subject, please email **dns Ltd** at info@dns.co.uk or call on **0870 085 8555**.

About the author

Lee is the lead penetration tester at **dns Ltd** with broad range of experience in IT security analysis, systems engineering & network security. He heads up a penetration testing team which has full testing capability.

Lee has taught penetration testing professionals throughout the world and has developed a deep understanding of the tools and techniques in use today which is constantly kept current. Lee has been involved with secure computer systems for over a decade, as a user and later as an engineer and penetration tester of Military networks.

He also brings to the table experiences gained in less obvious roles; that of a Military Intelligence operator on communication intercept duties (SigInt) and a Bomb Disposal Electronic Counter Measures (ECM) operator.

Lee has taught the EC-Council's '**Certified Ethical Hacker**' and Mile2's '**Certified Penetration Testing Specialist**' and '**Certified Penetration Testing Expert**' courses which are in worldwide release, he also authored the latter two courses.