



WHITE PAPER

Common web application security problems and how to identify them

Part 7 Improper error handling

By

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

Introduction

This is **part 7** 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 **Injection flaws**. This paper concentrates on **Improper error handling**, 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.

Improper error handling

Error conditions that occur during normal operation are not handled properly. If an attacker can cause errors to occur that the web application does not handle, they can gain detailed system information, deny service, cause security mechanisms to fail, or crash the server.

java application error messages are sometimes great for the hacker

The Problem

Error messages are a method of informing the user that something has gone wrong. That message is critical in the normal operation of any web application. During the development stages, error messages may be required to be more informative and therefore debugging levels may be raised. This is also critical to the developers. Problems arise if the debugging levels and error messages are too informative to end users.

Java application error messages are sometimes great for the hacker and penetration tester alike especially if the debugging mode is high. They can list source code and potentially expose other sensitive information such as SQL statements.

```

General Error

Could not obtain post/user information.

DEBUG MODE

SQL Error : 1016 Can't open file: 'nuke_bbposts_text.MYD'. (errno: 145)

SELECT u.username, u.user_id, u.user_posts, u.user_from, u.user_website, u.user_email, u.user_icq, u.user_aim, u.user_yim,
u.user_regdate, u.user_msnm, u.user_viewemail, u.user_rank, u.user_sig, u.user_sig_bbcodes, u.user_avatar,
u.user_avatar_type, u.user_allowavatar, u.user_allowsmile, p.*, pt.post_text, pt.post_subject, pt.bbcode_uid FROM
nuke_bbposts p, nuke_users u, nuke_bbposts_text pt WHERE p.topic_id = '1547' AND pt.post_id = p.post_id AND u.user_id =
p.poster_id ORDER BY p.post_time ASC LIMIT 0, 15

Line : 435
File : /usr/home/geeks/www/vonage/modules/Forums/viewtopic.php

```

Other error messages may give away much less information but may be just as useful to an attacker. The two example error messages shown below were generated by simply entering a single quote into a database web application. The resultant error messages differ because of the database server, Microsoft Access and SQL Server respectively. What is important with them however is the error number. This number indicates that the SQL statement has broken which may lead an attacker onto a SQL injection attack.

Microsoft OLE DB Provider for ODBC Drivers
error '80040e14'

([Microsoft][ODBC Microsoft Access Driver] Extra)
In query expression 'User=' AND Pass ='

/_tblemployees/logon.asp, line 36

error '80040e14'
/sqlall.asp, line 15

this could have severe financial consequences with loss of customer confidence and transactions

Apart from information disclosure, other problems may also occur. A specially crafted string of characters may cause errors to be generated on an application server whether that is with the application code or the actual server. The handling of that error could be critical to the ongoing availability of the service.

Should an unhandled error be discovered by an external attacker that causes the application to enter into an eternal loop or crash completely, then an ongoing series of crashes could be generated leading to a complete loss of service availability. This could have severe financial consequences with loss of customer confidence and transactions.

Other issues also include increased logging of error messages with the intent of filling up the allocated space for logs. Depending on how the server/application is configured, this may result in a shutdown of the server.

A common method for mapping out a web application is the use of HTTP error messages. To find web pages on a site, an attacker would browse the site, but what if authorisation was required? They could still find web pages by submitting GET requests to common names such as index, admin, default etc. The resulting error message would either read 'Not Found' or 'Access Denied'. Based on the error message, the attacker could determine whether the requested page exists or not.

Identification

Identifying security failings with error handling requires an error to be generated. There are numerous methods to achieve this depending on the application being targeted. Attempting to insert invalid characters into any application may result in error messages.

To identify if any of your applications are vulnerable to improper error handling, attempt to input the list of characters below and record the result.

```
|/\<>&& (; '+
```

If the application returns some form of error, or shows in some way that the process failed, then you may be vulnerable and further analysis is required.

Countermeasures

As usual, all user input should be validated prior to being applied to the application. This validation should be carried out at the server for security and on the client for speed.

For example, if an error is found on a user input field for a date of birth field, then validation should be carried out that the input data follows the following format:

```
DD/MM/YYYY
```

There are a number of methods for achieving this and they depend on the type of application and the input data.

A general rule of thumb is that all system generated error messages should be suppressed and replaced with bespoke error messages. These error messages can be more user friendly, therefore increasing the usability of the application, and they should not contain any sensitive information.

The next part of this series concentrates on **Insecure storage** and how it is used in exploitation.

all system generated error messages should be suppressed and replaced with bespoke error messages