Hacking Salesforce Lightning: A Guide for Bug Hunters

BY BLACKBIRD-EU · JULY 24, 2024 · LAST UPDATED ON MARCH 6, 2025

Salesforce Experience (or Community) Cloud is a CRM platform that helps software companies and organizations manage their customer relationships. Software companies and organizations often use it to manage their customer relationships, share information, and work with employees and customers (documentation, knowledge bases, help articles, etc.), provide support using support ticketing, and much more.

In this article, we'll go over some of the most common security misconfigurations present in Salesforce Experience Cloud that can lead to a wide variety of security vulnerabilities, ranging from sensitive data exposure (often including personally identifiable information) to allowing unauthorized users to perform unwanted actions. Please do note that other vulnerabilities like SOQL injections won't be part of this article.

What's Salesforce Experience Cloud?

Salesforce Experience (or Community) Cloud, is a CRM platform used by software teams to help manage their customer relationships. Companies can make their website exactly how they want it, add company logos, and add new features or functions.

Salesforce Communities is built upon the Salesforce Lightning framework. Lightning framework lets Salesforce customers quickly create new apps and websites using Aura Components and Apex, Salesforce's programming language.

Aura parts are used many times. They have interface and style elements, objects (a place to store data), functions (written in the Apex programming language), and other things. The Lightning framework comes with base parts that Salesforce admins can use. However, sometimes companies use their own custom Aura components to support more web pages, functions, and features.

These Aura parts made by you can be dangerous if they aren't set up correctly. From sensitive data leaks to privilege escalation vulnerabilities often caused by improper access controls. This article will show you one of the most common security mistakes made by Salesforce admins. These mistakes can often lead to very serious problems.

Common security misconfigurations

Salesforce admins can create Custom Aura components to provide additional functionality and features. Salesforce has many settings that can be changed. This can make it hard to use and could cause new security problems.

The most commonly occurring security vulnerability is the lack of access control.

You can easily find any Aura components that are enabled or used in javascript files that are in most web pages. Once you have enumerated all Object names, you can craft a POST HTTP request and interact with these different components through the Aura API endpoint, usually located at one of the following endpoints:

```
/aura
/sfsites/aura
/s/sfsites/aura
```

```
POST /aura HTTP/2
Host: {TARGET}.lightning.force.com
Content-Type: application/x-www-form-urlencoded

message=...
```

TIP: Misconfig Mapper is an automated tool that can help you detect Aura API endpoints at scale.

Using the Aura API endpoint, we can craft specific requests to interact with the available Objects and Components. Let's reconstruct some basic requests.

To further simplify the examples below, we've URL-decoded the contents of the "message" body parameter in the request.

```
{
  "actions": [
    {
      "id": "1234;a",
      "descriptor":
"serviceComponent://ui.force.components.controllers.hostConfig.HostConfigController/ACTION$getConfigData",
      "callingDescriptor": "UNKNOWN",
      "params": { }
    }
  ]
}
```

In the JSON message above, we are requesting more information about the available Salesforce Objects and Components (including custom ones). We can further deconstruct the message into the following properties:

- **ID:** A unique identifier to help separate multiple methods (if multiple methods are sent in a single API call)
- **Descriptor:** The method descriptor that instructs Salesforce to return/interact with the correct Object or Component. The Descriptor takes the following formatting: `{classType}://{controllerNamespace}/ACTION${methodName}`
- **CallingDescriptor:** This property defines the component responsible for calling the Component or Object in the request. It is almost always set to "UNKNOWN."
- **Params:** This property specifies any required Component parameters in a key-value format.

The request above will give you information about all the Objects and Components you can find. It'll also show you their controller namespaces and any required parameters. We can ask to work with all the Components and Objects and test them for security problems using this information.

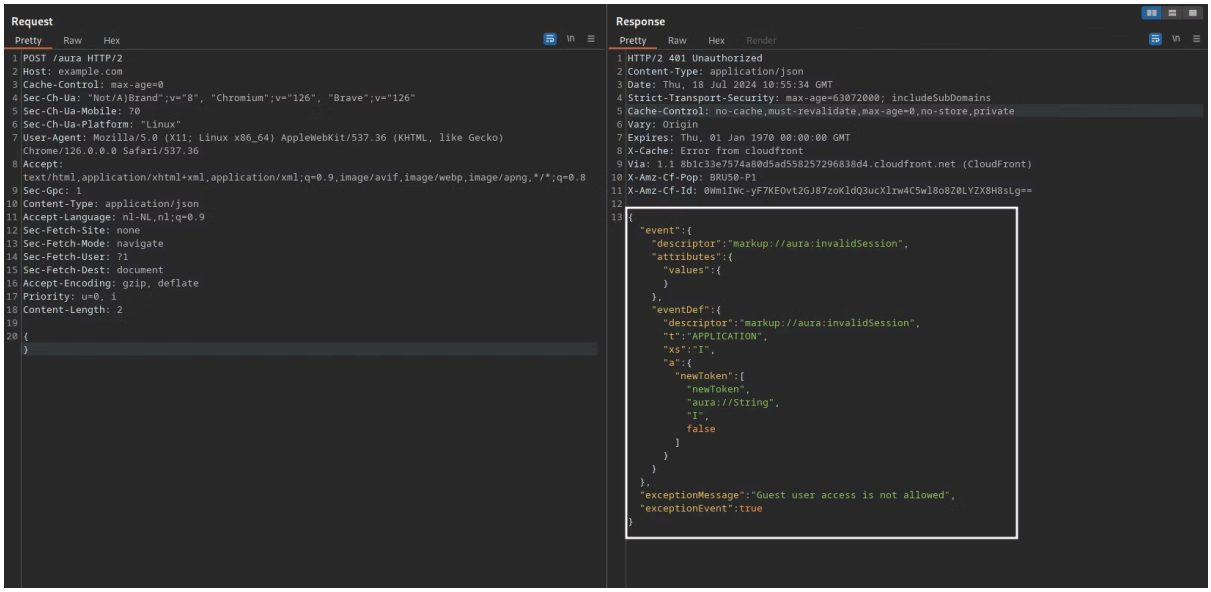
1) Improper access controls

Salesforce Communities is often used as a public-facing support portal to help customers and employees get quick access to information (help articles, KBs, documentation, ...) and also to manage support tickets. This means that you'll have both authenticated as well as unauthenticated visitors.

Salesforce lets you control who can access certain parts of the website. For example, you can use different Profiles to do this. Salesforce has three different levels of access control: Object level (an Object is like a database), Field level (a Field is like a database column), and Record level (like a record data entry).

If you miss a simple check, people who aren't allowed can read more data than allowed or do illegal things that can cause unwanted changes. Therefore, it's essential to cross-check all enabled options, especially for custom Profiles and Components.

Most of the time, visitors who aren't signed in can see the full names of other users. Sometimes, they can also see other important data, like email and phone numbers. To avoid this, first look at all the options for the Guest profile and use the principle of least privilege where possible:



Disabled access for unauthorized users

Important: Make sure that in case you limit the privileges of unauthenticated visitors, the option for self-signup is disabled. If this is not the case, bad actors can still circumvent all your access control measures by just registering for an account.

2) File uploads

As we said before, some places let customers (or employees) create support tickets. They can also upload any files to help with the support case. Another common security problem in Salesforce Communities, especially older versions, is that visitors who don't have an account can create support tickets without

being logged in. This means that the Salesforce part that gets these files from a support ticket can't check who owns them. Leaving files that often contain sensitive data at risk of being viewed by anyone.

Fortunately, Salesforce started pushing security patches each season. These patches are aimed to help mitigate these types of security misconfigurations. Salesforce admins do receive notifications of these but aren't implemented by default as some may introduce breaking changes.

One of the past security patches made sure that uploaded files were automatically assigned to the Admin Profile. This made it impossible for others to see them, which was a working solution.

Another common security best practice that's often not followed is restricting allowed file types or a limit on file sizes. Setting a white-list for file uploads makes sure that the team who gets support tickets won't have to deal with any unwanted files. So make sure to set a strict allow list if possible.

Automated tooling

Salesforce offers official tools such as Salesforce Shield and Salesforce Code Analyzer to help you secure your Salesforce Communities instance and make sure you follow best practices where possible. That way, you can ensure that you mitigate most security vulnerabilities that are accidentally introduced in custom Components that could for example lead to data leaks.

As a Salesforce admin, you can also make use of Salesforce Health Check to get a security score on your current instance configurations. Salesforce Health Check will also report back any potential vulnerabilities to further help you secure your instance.

Conclusion

Third-party services can save teams a lot of time, and help them solve technical problems. They can also make them more efficient, but they often have new tasks. Always follow the best ways to do things and any security guides you can find to reduce any possible security problems.