

# S2-015

## Summary

A vulnerability introduced by wildcard matching mechanism or double evaluation of OGNL Expression allows remote command execution.

<b>Who should read this</b>	All Struts 2 developers and users
<b>Impact of vulnerability</b>	Remote command execution, remote server context manipulation, injection of malicious client side code
<b>Maximum security rating</b>	Critical
<b>Recommendation</b>	Developers should immediately upgrade to <a href="#">Struts 2.3.14.3</a>
<b>Affected Software</b>	Struts 2.0.0 - Struts 2.3.14.2
<b>Reporter</b>	Jon Passki from Coverity Security Research Laboratory reported directly to security@struts.a.o and via <a href="#">blog post</a>
<b>CVE Identifier</b>	<a href="#">CVE-2013-2135</a> , <a href="#">CVE-2013-2134</a>

## Problem

Struts 2 allows define action mapping base on wildcards, like in example below:

```
<action name="*" class="example.ExampleSupport">
  <result>/example/{1}.jsp</result>
</action>
```

If a request doesn't match any other defined action, it will be matched by \* and requested action name will be used to load JSP file base on the name of action. And as value of {1} is threaten as an OGNL expression, thus allow to execute arbitrary Java code on server side. This vulnerability is combination of two problems:

- requested action name isn't escaped or checked againsts whitelist
- double evaluation of an OGNL expression in `TextParseUtil.translateVariables` when combination of \$ and % open chars is used.

## Proof of concept

### Wildcard matching

1. Run struts2-blank app
2. Open the following url, resulting in dynamic action name resolution based on passed value of #foo

```
http://localhost:8080/example/%24%7B%23foo%3D%27Menu%27%2C%23foo%7D
```

```
http://localhost:8080/example/${#foo='Menu',#foo}
```

As you can notice, action name is resolved based on user input and you can put any arbitrary code to perform attack.

### Double evaluation of an expression

1. Open example.xml present in the Struts Blank App and change result of HelloWorld action to one below:

```
<result type="httpheader">
  <param name="headers.fooBar">${message}</param>
</result>
```

2. Open HelloWorld.java and change `execute()` method as below:

```
public String execute() throws Exception {  
    return SUCCESS;  
}
```

3. Run struts2-blank app
4. Open the following url (you must have a tool to check response headers)

```
http://localhost:8080/example/HelloWorld.action?message=%24{%25{1%2B2}}
```

```
http://localhost:8080/example/HelloWorld.action?message=${%{1+2}}
```

5. Check value of `foobar` header, it should be 3

As you can notice, passed value of `message` parameter was used to set value of `foobar` header and the value was double evaluated - first time when `${message}` was evaluated, secondly when parsed value (`${%{1+2}}`) was evaluated again.

## Solution

With the new version actions' names whitelisting was introduced and by default is set to accept actions that match the following regex:

```
[a-z]*[A-Z]*[0-9]*[.\-_!/*]
```

user can change the definition by setting up a new constant in `struts.xml` as below:

```
<constant name="struts.allowed.action.names" value="[a-zA-Z]*" />
```

Double evaluation of passed expression was removed from `OgnlTextParser` which is used by `TextParseUtil.translateVariables`.



### Backward Compatibility

There should be no problems with migration from previous version.



It is strongly recommended to upgrade to **Struts 2.3.14.3**.