Forced double OGNL evaluation, when evaluated on raw user input in tag attributes, may lead to remote code execution.

Who should read this: All Struts 2 developers and users

Impact of vulnerability: Possible Remote Code Execution vulnerability

Maximum security rating: Important

Recommendation:
- Always validate incoming parameters’ values when re-assigning them to certain Struts’ tags attributes.
- Don’t use %{...} or ${...} syntax referencing user modifiable input in tag attributes other than value unless you have a valid use-case.
- Consider activating Proactive OGNL Expression Injection Protection if applicable.
- Alternatively upgrade to Struts 2.5.22 or greater

Affected Software: Struts 2.0.0 - Struts 2.5.20

Reporters: Matthias Kaiser, Apple Information Security

CVE Identifier: CVE-2019-0230

Problem:
The Apache Struts frameworks, when forced, performs double evaluation of attributes’ values assigned to certain tags attributes such as id so it is possible to pass in a value that will be evaluated again when a tag’s attributes will be rendered. With a carefully crafted request, this can lead to Remote Code Execution (RCE).

The problem only applies when forcing OGNL evaluation inside a Struts tag attribute, when the expression to evaluate references raw, unvalidated input that an attacker is able to directly modify by crafting a corresponding request.

Example:
```xml
<s:url var="url" namespace="/employee" action="list"/>
<s:a id="%{skillName}" href="%{url}">List available Employees</s:a>
```

If an attacker is able to modify the skillName attribute in a request such that a raw OGNL expression gets passed to the skillName property without further validation, the provided OGNL expression contained in the skillName attribute gets evaluated when the tag is rendered as a result of the request.

The opportunity for using double evaluation is by design in Struts since 2.0.0 and a useful tool when done right, which most notably means only referencing validated values in the given expression. However, when referencing unvalidated user input in the expression, malicious code can get injected. In an ongoing effort, the Struts framework includes mitigations for limiting the impact of injected expressions, but Struts before 2.5.22 left an attack vector open which is addressed by this report.

This issue is similar to:
- S2-029
- S2-036

Solution:
The ultimate fix is adding a proper validation of each value that’s coming in and it’s used in tag’s attributes. Don’t use forced evaluation of an attribute other than value using %{...} or ${...} syntax unless really needed for a valid use-case.

By upgrading to Struts 2.5.22, possible malicious effects of forced double evaluation are further limited and close the reported attack vector, especially when combined with Proactive OGNL Expression Injection Protection.

Backward compatibility:
No issues expected when upgrading to Struts 2.5.22

References:
- LGTM / Semmle: An analysis of the double evaluation behaviour and which attributes support it
• Use Struts tags instead of raw EL expressions
• Proactively protect from OGNL Expression Injections attacks if easily applicable