| **Name** | **Regular Expression** |
| --- | --- |
| SQL Injection | [\s]\*((delete)|(exec)|(drop\s\*table)|(insert)|(shutdown)|(update)|(\bor\b)) |
| Server-Side Include Injection | <!--#(include|exec|echo|config|printenv)\s+.\*  XML encoded:  &lt;!--#(include|exec|echo|config|printenv)\s+.\* |
| XPath Abbreviated Syntax Injection | (/(@?[\w\_?\w:\\*]+(\+\])\*)?)+ |
| XPath Expanded Syntax Injection | /?(ancestor(-or-self)?|descendant(-or-self)?|following(-sibling)) |
| JavaScript Injection | <\s\*script\b[^>]\*>[^<]+<\s\*/\s\*script\s\*>  XML encoded: &lt;\s\*script\b[^&gt;]\*&gt;[^&lt;]+&lt;\s\*/\s\*script\s\*&gt;  See also [this community post](https://community.apigee.com/questions/27198/xss-threat-protection-patterns.html#answer-27465) on XSS threat protection patterns. |
| Java Exception Injection | .\*Exception in thread.\* |

<Pattern>[\s]\*(?i)((delete)|(exec)|(drop\s\*table)|(insert)|(shutdown)|(update)|(\bor\b))</Pattern>

<RegularExpressionProtection name="JsonPathRegExProtection">

<DisplayName>Regular Expression Protection 1</DisplayName>

<Source>request</Source>

<JSONPayload escapeSlashCharacter="true">

<JSONPath>

<Expression>$.</Expression>

<Pattern>&lt;\s\*script\b[^&gt;]\*&gt;[^&lt;]+&lt;\s\*\/\s\*script\s\*&gt;

</Pattern>

<Pattern>n\s\*\\\\\s\*slash</Pattern>

<Pattern>n\s\*\/\s\*slash</Pattern>

<Pattern>n\s\*\\"\s\*quotes</Pattern>

<Pattern>n\s\*\\b\s\*space</Pattern>

<Pattern>n\s\*\\f\s\*forwardfeed</Pattern>

<Pattern>n\s\*\\n\s\*newline</Pattern>

<Pattern>n\s\*\\r\s\*carria</Pattern>

<Pattern>n\s\*\\t\s\*tab</Pattern>

<Pattern>n\s\*\\uFFFF\s\*hex</Pattern>

</JSONPath>

</JSONPayload>

</RegularExpressionProtection>