New issue Jump to bottom

The RSA-PSS implementation does not detect signature modification (prepending "0" bytes) to the signature #438

Closed **adelapie** opened this issue on Jun 6, 2020 · 3 comments

Labels bug

adelapie commented on Jun 6, 2020

The jsrsasign 8.0.16 RSASSA-PSS (RSA-PSS) implementation does not detect prepending 0s to the signature and accepts modifies signatures with prepended 0's as valid.

You can verify this using the following test vectors from Google Wycheproof:


```
{
  "algorithm" : "RSASSA-PSS",
  "generatorVersion" : "0.8r12",
  "numberOfTests" : 103,
  "header" : [
    "Test vectors of class RsassaPssVerify are intended for checking the",
    "verification of RSASSA-PSS signatures."
  ],
  "notes" : {
  },
  "schema" : "rsassa_pss_verify_schema.json",
  "testGroups" : [
    {
      "e" : "010001",
      "keyAsn" :
"3082010a0282010100a2b451a07d0aa5f96e4556715135505l4a8a5b462ebef717094fa1fee82224e637f9746d3f7cafd31878d80325b6ef5a1700f65903b469429e89d6eac8845097b5ab393189db92512ed8a7711a1253facd20
      "keyDer" :
"30820122300d06092a864886f70d01010105000382010f003082010a0282010100a2b451a07d0aa5f96e4556715135505l4a8a5b462ebef717094fa1fee82224e637f9746d3f7cafd31878d80325b6ef5a1700f65903b469429e89
      "keyPem" : "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAorRRoH0KpfluRVZxUTVQ\nUUqKW0YuvvcXCU+h/ugiJOY3+XRtP3yv0xh42AMltu9aFwD2WQO0aUKeidbqyIRQ\nl7WrOTGJ25JRLtincRoSU/rNIPecFegkfz0+QuRuSMmOJUov6
----END PUBLIC KEY-----",
      "keysize" : 2048,
      "mgf" : "MGF1",
      "mgfSha" : "SHA-256",
      "n" :
"00a2b451a07d0aa5f96e455671513550514a8a5b462ebef717094fa1fee82224e637f9746d3f7cafd31878d80325b6ef5a1700f65903b469429e89d6eac8845097b5ab393189db92512ed8a7711a1253facd20f79c15e8247f3d3e
      "sLen" : 32,
      "sha" : "SHA-256",
      "type" : "RsassaPssVerify",
      "tests" : [
        {
          "tcId" : 99,
          "comment" : "prepending 0's to signature",
          "msg" : "313233343030",
          "sig" :
"000068caf07e71ee654ffabf07d342fc4059deb4f7e5970746c423b1e8f668d5332275cc35eb61270aebd27855b1e80d59def47fe8882867fd33c2308c91976baa0b1df952caa78db4828ab81e79949bf145cbdfd1c4987ed036f8
          "result" : "invalid",
          "flags" : []
        },
        {
          "tcId" : 100,
          "comment" : "correct signature",
          "msg" : "313233343030",
          "sig" :
"68caf07e71ee654ffabf07d342fc4059deb4f7e5970746c423b1e8f668d5332275cc35eb61270aebd27855b1e80d59def47fe8882867fd33c2308c91976baa0b1df952caa78db4828ab81e79949bf145cbdfd1c4987ed036f81e84
          "result" : "valid",
          "flags" : []
        },
        {
          "tcId" : 101,
          "comment" : "appending 0's to signature",
          "msg" : "313233343030",
          "sig" :
"68caf07e71ee654ffabf07d342fc4059deb4f7e5970746c423b1e8f668d5332275cc35eb61270aebd27855b1e80d59def47fe8882867fd33c2308c91976baa0b1df952caa78db4828ab81e79949bf145cbdfd1c4987ed036f81e84
          "result" : "invalid",
          "flags" : []
        }
      ]
    }
  ]
}
```


in the following proof of concept:

```
var rs = require('jsrsasign');
var obj = require("./rsa_pss.json");

for (let testGroup of obj.testGroups) {

    var keyPem = testGroup.keyPem;

    for(let test of testGroup.tests) {
     console.log("[*] Test " + test.tcId + " result: " + test.result)

     try {
      var sig = new rs.Signature({alg: 'SHA256withRSAandMGF1'});
      sig.init(keyPem);
```

```
        sig.updateHex(test.msg);
        var result = sig.verify(test.sig);

       if (result == true) {
        if (test.result == "valid" || test.result == "acceptable")
         console.log("Result: PASS");
        else
         console.log("Result: FAIL")
       }

       if (result == false) {
        if (test.result == "valid" || test.result == "acceptable")
         console.log("Result: FAIL");
        else
         console.log("Result: PASS")
       }

       } catch (e) {
        console.log("ERROR - VERIFY: " + e)

        if (test.result == "valid" || test.result == "acceptable")
         console.log("Result: FAIL");
        else
         console.log("Result: PASS")

       }

      }
  }
```

with result:

```
[*] Test 99 result: invalid
Result: FAIL
[*] Test 100 result: valid
Result: PASS
[*] Test 101 result: invalid
Result: PASS
```

Best regards,
Antonio

kjur commented on Jun 19, 2020 — Owner

Thank you for your report. The issue was fixed in 8.0.17 release today.

kjur closed this as completed on Jun 19, 2020

---

adelapie commented on Jun 22, 2020 — Author

CVE-2020-14968 is assigned to this issue with the following description: An issue was discovered in the jsrsasign package before 8.0.17 for Node.js. Its RSASSA-PSS (RSA-PSS) implementation does not detect signature manipulation/modification by prepending '\0' bytes to a signature (it accepts these modified signatures as valid). An attacker can abuse this behavior in an application by creating multiple valid signatures where only one signature should exist. Also, an attacker might prepend these bytes with the goal of triggering memory corruption issues.

kjur commented on Jun 23, 2020 — Owner

jsrsasign security advisory (2020-Jun-24):
CVE-2020-14968
RSA-PSS signature validation vulnerability by prepending zeros
GHSA-q3gh-5r98-j4h3

kjur added the bug label on Aug 18, 2020

This was referenced on Mar 13, 2021

Bump jsrsasign from 8.0.12 to 8.0.19 m0rphtail/Teleport#6
Closed

Bump jsrsasign from 8.0.12 to 8.0.19 Cyper77/CyberChef#1
Closed

fhirfly mentioned this issue on Apr 22, 2021

NPM still reports Vulnerabilities in Utils even though issues closed #481
Closed

Assignees

No one assigned

Labels

bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants