New issue

Jump to bottom

Lack of encoding checks allows a certain degree of signature malleability in ECDSA signatures #226

Closed **adelapie** opened this issue on Jun 1, 2020 · 19 comments

adelapie commented on Jun 1, 2020

Hello,

Using elliptic 6.5.2 I've found that the ECDSA verification functionality validates signatures as 'true' when the encoding is incorrect i.e. it has been modified / altered against the standard, allowing a certain degree of malleability in the signatures. Based on the Google Wycheproof test vectors, the following changes on an ECDSA signature are not detected:

- "long form encoding of length of sequence",
- "length of sequence contains leading 0",
- "length of integer contains leading 0",
- "uint32 overflow in length of integer",
- "uint64 overflow in length of integer",
- "prepending 0's to integer",
- "long form encoding of length of integer"

See the proof of concept and test vectors below:

```
// ECDSA test

var crypto = require('crypto')
var EC = require('elliptic').ec;
var ec = new EC('secp256k1');

var obj = require("./poc_ecdsa_secp256k1_sha256_test.json");

for (let testGroup of obj.testGroups) {

    var key = ec.keyFromPublic(testGroup.key.uncompressed, 'hex');

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

     msgHash = crypto.createHash('sha256').update(Buffer.from(test.msg, 'hex')).digest();

     try {
      result = key.verify(msgHash, Buffer.from(test.sig, 'hex'));

     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")



     }


    }

}
```

Test vectors:

```
{
  "algorithm" : "ECDSA",
  "generatorVersion" : "0.8r12",
  "numberOfTests" : 380,
  "header" : [
    "Test vectors of type EcdsaVerify are meant for the verification",
    "of ASN encoded ECDSA signatures."
  ],
  "notes" : {
    "BER" : "This is a signature with correct values for (r, s) but using some alternative BER encoding instead of DER encoding. Implementations should not accept such signatures
to limit signature malleability.",
    "EdgeCase" : "Edge case values such as r=1 and s=0 can lead to forgeries if the ECDSA implementation does not check boundaries and computes s^(-1)==0.",
    "MissingZero" : "Some implementations of ECDSA and DSA incorrectly encode r and s by not including leading zeros in the ASN encoding of integers when necessary. Hence, some
implementations (e.g. jdk) allow signatures with incorrect ASN encodings assuming that the signature is otherwise valid.",
    "PointDuplication" : "Some implementations of ECDSA do not handle duplication and points at infinity correctly. This is a test vector that has been specially crafted to check
for such an omission."
  },
```

```
  "schema" : "ecdsa_verify_schema.json",
  "testGroups" : [
    {
      "key" : {
        "curve" : "secp256k1",
        "keySize" : 256,
        "type" : "EcPublicKey",
        "uncompressed" : "04b838ff44e5bc177bf21189d0766082fc9d843226887fc9760371100b7ee20a6ff0c9d75bfba7b31a6bca1974496eeb56de357071955d83c4b1badaa0b21832e9",
        "wx" : "00b838ff44e5bc177bf21189d0766082fc9d843226887fc9760371100b7ee20a6f",
        "wy" : "00f0c9d75bfba7b31a6bca1974496eeb56de357071955d83c4b1badaa0b21832e9"
      },
      "keyDer" :
"3056301006072a8648ce3d020106052b8104000a03420004b838ff44e5bc177bf21189d0766082fc9d843226887fc9760371100b7ee20a6ff0c9d75bfba7b31a6bca1974496eeb56de357071955d83c4b1badaa0b21832e9",
      "keyPem" : "-----BEGIN PUBLIC KEY-----\nMFYwEAYHKoZIzj0CAQYFK4EEAAoDQgAEuDj/ROW8F3vyEYnQdmCC/J2EMiaIf812\nA3EQC37iCm/wyddb+6ezGmvKGXRJbutW3jVwcZVdg8Sxutqgshgy6Q==\n-----END
PUBLIC KEY-----",
      "sha" : "SHA-256",
      "type" : "EcdsaVerify",
      "tests" : [
        {
          "tcId" : 4,
          "comment" : "long form encoding of length of sequence",
          "msg" : "313233343030",
          "sig" : "3081450221008813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc983236502206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : [
            "BER"
          ]
        },
        {
          "tcId" : 5,
          "comment" : "length of sequence contains leading 0",
          "msg" : "313233343030",
          "sig" : "308200450221008813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc983236502206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : [
            "BER"
          ]
        },
        {
          "tcId" : 8,
          "comment" : "uint32 overflow in length of sequence",
          "msg" : "313233343030",
          "sig" : "3085010000004502210081 3ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc983236502206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : []
        },
        {
          "tcId" : 9,
          "comment" : "uint64 overflow in length of sequence",
          "msg" : "313233343030",
          "sig" :
"3089010000000000000004502210081 3ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc983236502206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : []
        },
        {
          "tcId" : 68,
          "comment" : "long form encoding of length of integer",
          "msg" : "313233343030",
          "sig" : "3046028121008813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc983236502206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : [
            "BER"
          ]
        },
        {
          "tcId" : 69,
          "comment" : "long form encoding of length of integer",
          "msg" : "313233343030",
          "sig" : "3046022100813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc983236502 81206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : [
            "BER"
          ]
        },
        {
          "tcId" : 70,
          "comment" : "length of integer contains leading 0",
          "msg" : "313233343030",
          "sig" : "3047028200210081 3ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc983236502206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : [
            "BER"
          ]
        },
        {
          "tcId" : 71,
          "comment" : "length of integer contains leading 0",
          "msg" : "313233343030",
          "sig" : "3047022100813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc9832365028200206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : [
            "BER"
          ]
        },
        {
          "tcId" : 76,
          "comment" : "uint32 overflow in length of integer",
          "msg" : "313233343030",
          "sig" : "304a028501000000 2100813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc983236502206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : []
        },
        {
          "tcId" : 77,
          "comment" : "uint32 overflow in length of integer",
          "msg" : "313233343030",
          "sig" : "304a022100813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc9832365028501000000206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : []
        },
        {
          "tcId" : 78,
```

```
          "comment" : "uint64 overflow in length of integer",
          "msg" : "313233343030",
          "sig" :
"304e0289010000000000000002100813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc983236502206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : []
        },
        {
          "tcId" : 79,
          "comment" : "uint64 overflow in length of integer",
          "msg" : "313233343030",
          "sig" :
"304e022100813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc98323650289010000000000000206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : []
        },
        {
          "tcId" : 95,
          "comment" : "prepending 0's to integer",
          "msg" : "313233343030",
          "sig" : "3047022300000813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc983236502206ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : [
            "BER"
          ]
        },
        {
          "tcId" : 96,
          "comment" : "prepending 0's to integer",
          "msg" : "313233343030",
          "sig" : "3047022100813ef79ccefa9a56f7ba805f0e478584fe5f0dd5f567bc09b5123ccbc9832365022200006ff18a52dcc0336f7af62400a6dd9b810732baf1ff758000d6f613a556eb31ba",
          "result" : "invalid",
          "flags" : [
            "BER"
          ]
        }
      ]
    }
  ]
}
```


adelapie commented on Jun 4, 2020 — Author

I've obtained a CVE with identifier 2020-13822 (in case developers want to refer to this problem) and the following description: The Elliptic package 6.5.2 for Node.js allows ECDSA signature malleability via variations in encoding, leading '\0' bytes, or integer overflows. This could conceivably have a security-relevant impact if an application relied on a single canonical signature.

adelapie commented on Jun 4, 2020 — Author

The following links elaborate on the implications of signature malleability in the case of ECDSA:

- https://yondon.blog/2019/01/01/how-not-to-use-ecdsa/
- https://medium.com/@herman_10687/malleability-attack-why-it-matters-7b5f59fb99a4

drifterz28 commented on Jun 16, 2020

Is there any movement on this? any help needed?

👍 6

Rahul1408 commented on Jun 17, 2020

We are getting high severity vulnerability with "elliptic" package.
Package Manager: npm
Vulnerable module: elliptic

Link: https://snyk.io/vuln/SNYK-JS-ELLIPTIC-571484

Remediation:
There is no fixed version for elliptic.

Any idea by when this can be fixed?

fanatid mentioned this issue on Jun 18, 2020

Security vulnerability #227

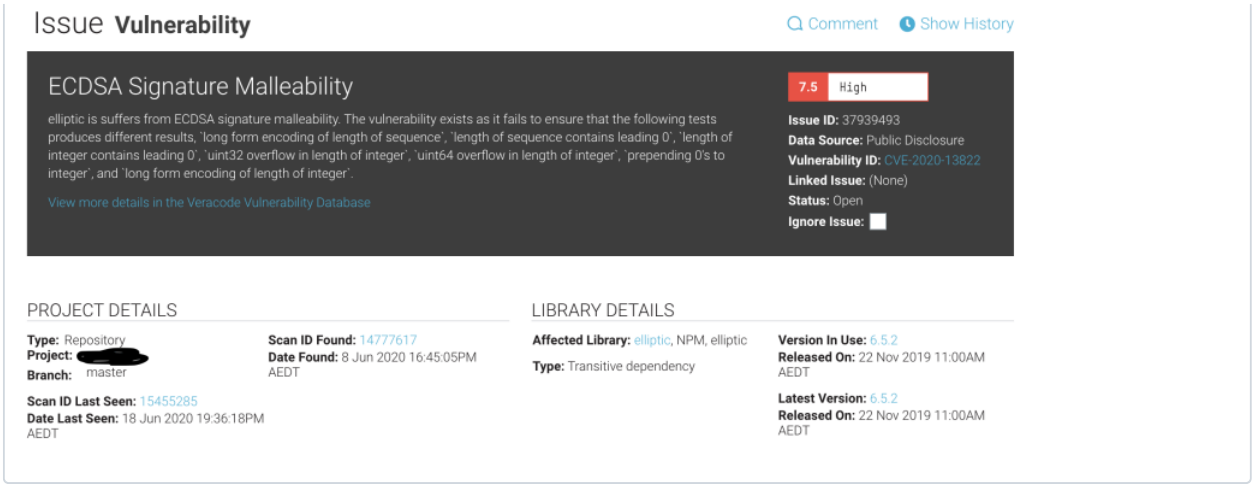
Closed

yenanandu commented on Jun 18, 2020

We also facing the vulnerability issue with latest version of "elliptic"

kiranio commented on Jun 18, 2020

Guys, any pointers for this, we are getting the high severity vulnerability from sourceclear



indutny commented on Jun 18, 2020 — Owner

Published 6.5.3 . It appears that it passes all tests.

@adelapie would you feel like contributing the test suite to the library? It is unclear whether the code that you have submitted is under compatible license.

👍 3 🎉 3 🚀 2

mleneveut mentioned this issue on Jun 18, 2020

CVE-2020-13822 in elliptic : upgrade to 6.5.3 Brightspace/node-jwk-to-pem#38

Closed

yenanandu commented on Jun 19, 2020

@indutny - Still snyk throws high severity issue on 6.5.3 version. Please refer the below link for same.
https://snyk.io/test/npm/elliptic/6.5.3

👍 1

indutny commented on Jun 19, 2020 — Owner

For the lack of better channels - I have sent them a DM on twitter 😂 Not sure what's the best way to handle this.

adelapie commented on Jun 19, 2020 — Author

> Published 6.5.3 . It appears that it passes all tests.
>
> **@adelapie** would you feel like contributing the test suite to the library? It is unclear whether the code that you have submitted is under compatible license.

Sure, please consider the code I submitted as public domain

benjifin commented on Jun 19, 2020

@indutny we review all new releases within 24 hours of when they are published - but we do need to manually verify things, so it's a slight delay between when you publish and when we update the DB (we just did that verification process now - and saw this update in this issue :)) but you can always ping the security team at snyk direct using our email: report@snyk.io if you want to nudge us along or give us a heads up that you're readying a release!

> For the lack of better channels - I have sent them a DM on twitter 😂 Not sure what's the best way to handle this.

This was referenced on Jun 29, 2020

web/ui/react-app: bump elliptic package version prometheus/prometheus#7477

Merged

Upgrade elliptic package to 6.5.3 (or later) grafana/grafana#25888

Closed

munepom commented on Jun 30, 2020

sourceclear determined elliptic v6.5.3 to be safe version!!!
Thank you!!!

indutny commented on Jun 30, 2020 — Owner

Hooray. Thanks everyone!

🎉 1

nalgritz commented on Jul 23, 2020

hello all, I went to sourceclear and it still shows the latest version has vulnerable. Which source should I believe?
https://www.sourceclear.com/vulnerability-database/libraries/elliptic/javascript/npm/lid-6755/summary

mvayngrib mentioned this issue on Jul 29, 2020

signature: prevent malleability and overflows ExodusMovement/elliptic#3

Merged

cronicc mentioned this issue on Aug 1, 2020

Release 1.2.6 HorizenOfficial/zencashjs#39

Merged

williams-brian commented on Aug 3, 2020

@indutny Looks like this issue can be closed since it was fixed w/ 6.5.3? Open status gives the impression that it's not fixed yet.

indutny closed this as completed on Aug 3, 2020

---

indutny commented on Aug 3, 2020 — Owner

Thanks for a reminder **@williams-brian** !

marlonbrgomes mentioned this issue on Aug 17, 2020

High vulnerability issue found when installing tronweb tronprotocol/tronweb#99

Closed

bsomeshwer commented on Oct 1, 2020 • edited

Hi

I'm using Elliptic 6.5.3 version but still I'm facing this issue in my project.

```
  More info       https://github.com/indutny/elliptic/issues/226,https://medi…



                              Manual Review
            Some vulnerabilities require your attention to resolve

          Visit https://go.npm.me/audit-guide for additional guidance


  Moderate        The Elliptic package 6.5.2 for Node.js allows ECDSA
                  signature malleability via variations in encoding, leading
                  '\0' bytes, or integer overflows. This could conceivably
                  have a security-relevant impact if an application relied on
                  a single canonical signature.

  Package         elliptic

  Patched in      6.5.3

  Dependency of   laravel-mix [dev]

  Path            laravel-mix > webpack > node-libs-browser >
                  crypto-browserify > browserify-sign > elliptic

  More info       https://github.com/indutny/elliptic/issues/226,https://medi…

found 2 moderate severity vulnerabilities in 1830 scanned packages
  run `npm audit fix` to fix 1 of them.
  1 vulnerability requires manual review. See the full report for details.
root@mccoy-dev-5dbfbd6f5c-dglrq:~#
```

Could you please let me know the fix for this?

I tried npm install elliptic@6.5.3
and
npm audit fix
and I played around lot of other ways but still issue persists.

Thanks

adamdaly commented on Nov 12, 2020

Hi, I'm also seeing this error. Or scans (using jFrog) are also picking up this violation, which states that both the infected and fixed versions are 6.5.3.

```
=== npm audit security report ===

                                Manual Review
            Some vulnerabilities require your attention to resolve

          Visit https://go.npm.me/audit-guide for additional guidance

┌─────────────────┬──────────────────────────────────────────────────────────────┐
│ Moderate        │ The Elliptic package 6.5.2 for Node.js allows ECDSA          │
│                 │ signature malleability via variations in encoding, leading   │
│                 │ '\0' bytes, or integer overflows. This could conceivably     │
│                 │ have a security-relevant impact if an application relied on  │
│                 │ a single canonical signature.                                │
├─────────────────┼──────────────────────────────────────────────────────────────┤
│ Package         │ elliptic                                                     │
├─────────────────┼──────────────────────────────────────────────────────────────┤
│ Patched in      │ 6.5.3                                                        │
├─────────────────┼──────────────────────────────────────────────────────────────┤
│ Dependency of   │ react-scripts [dev]                                          │
├─────────────────┼──────────────────────────────────────────────────────────────┤
│ Path            │ react-scripts > webpack > node-libs-browser >                │
│                 │ crypto-browserify > create-ecdh > elliptic                   │
├─────────────────┼──────────────────────────────────────────────────────────────┤
│ More info       │ https://github.com/indutny/elliptic/issues/226,https://medi… │
└─────────────────┴──────────────────────────────────────────────────────────────┘
found 1 moderate severity vulnerability in 2028 scanned packages
  1 vulnerability requires manual review. See the full report for details.
adamdaly in ~/Documents/dev/f/foundation-medicine/ebr-web update/ebr-2440:
```

Is there any movement on confirming this issue still exists/creating a fix?

indutny commented on Nov 12, 2020 — Owner

```
/tmp/aab $ npm init -y
Wrote to /private/tmp/aab/package.json:

{
  "name": "aab",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}


/tmp/aab $ npm install elliptic@6.5.3

added 8 packages, and audited 8 packages in 766ms

found 0 vulnerabilities
/tmp/aab $ npm audit
found 0 vulnerabilities
```

Seems to be working alright on my end...

debricked-staging bot mentioned this issue on Jun 18, 2021

Bulk vulnerability fix - Lockfile fix philomather/test-vue#3

Open

debricked bot mentioned this issue on Jul 9, 2021

Bulk vulnerability fix - Lockfile fix WeDoSoftware/webAppBackend#253

Open

debricked bot mentioned this issue on Aug 9, 2021

Bulk vulnerability fix - Lockfile fix debricked/react#2

Open

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

12 participants