LEDGER DONJON

← All Bulletins LSB 020

Ledger Security Bulletin 020

Ledger Live incorrectly parses some EIP-712 messages

20 November 2023: Ledger Live incorrectly parses some EIP-712 messages

Summary

When requesting to sign typed EIP-712 messages through Ledger Live and a Ledger device, the message which was actually signed could be different from the intended one.

More precisely, if an integer field contained a hexadecimal number with an odd number of characters, its value was reduced. For example, requesting the signature of a message meaning “I want to sell this asset for 0x123 tokens”, the signature of “I want to sell this asset for 0x12 tokens” would have been produced.

This issue is fixed in `ledgerhq/hw-app-eth` version 6.34.7, included in Ledger Live 2.70.0.

Description

EIP-712 is a standard which defines a way to sign messages with several fields. This can be used to authorize some actions without sending a transaction on Ethereum. For example, let’s imagine a marketplace where users can sell their assets and the Ethereum transaction fees are paid by the buyer. A way to do implement this mechanism would consist in signing a message containing some identifier of the asset, the address of the buyer and the value:

1. The seller crafts a EIP-712 message with all the needed information and signs it.
2. The seller shares this message with the buyer.
3. The buyer invokes a method of the smart contract of the marketplace to claim the asset. This invocation is an Ethereum transaction which also transfers enough tokens to the seller.

(N.B. This is a simple example. A real marketplace protocol is likely to include more fields to be secure, like a way for the seller to cancel the trade if the buyer takes too long to make the transaction.)

In practice, the first step can be implemented by using WalletConnect's package `walletconnect/sign-client` and invoking the method `eth_signTypedData_v4`. The signed message can have a field named `value` using type `uint256`.

If the seller uses Ledger Live with a Ledger Nano X device, the message to be signed goes through several libraries before hitting the device. A problem occurs when the integer in field `value` is encoded in hexadecimal: one of these libraries (`ledgerhq/hw-app-eth`) decodes it incorrectly and propagates this wrong value. For example `0x123` (which is the number 291) is decoded to `0x12`, which is 18. This leads to the Nano X displaying a hash which does not match the initial message. If the user approves this hash, the signed message is incorrect and an attacker who manages to get this signature can abuse it to buy the asset for 18 tokens instead of 291.

Thankfully, Ledger Live also displays the right hash of the message, so users can find out that something wrong is going on. If you encounter a situation where the data displayed by Ledger Live does not match the one on your device, please reject the signature and contact Ledger's support instead, as this could be a bug in our products.

Root cause of the issue

The issue was caused by using Node.js's function `Buffer.from` to decode a hexadecimal number in bytes. When this function encounters an odd number of characters, it ignores the last one. This can be reproduced in Node.js' interactive command-line interface:

```
$ node
> Buffer.from("1234", "hex")
<Buffer 12 34>

> Buffer.from("123", "hex")
<Buffer 12>
```

To fix this, `hw-app-eth/src/utils.ts` was modified to automatically prepend a `0` to the string before decoding it, in Ledger Live's Pull Request #4687.

Mitigations

Version 6.34.7 of `ledgerhq/hw-app-eth` fixed the issue. It is recommended for all projects using it to update to the newest version.

Credits

We would like to thank Ian Fisher, who discovered the vulnerability and disclosed it through our bug bounty program.

← PREVIOUS
LSB 019

NEXT →
LSB 021