Ledger Receive Address Attack

Overview

Crypto wallets consist of a private key for spending funds, and a public key for receiving funds.

Modern Crypto clients usually create a new receive address after every transaction.

This is done to better protect the privacy of the user, by spreading his funds across multiple addresses, rather than one.

Receive addresses are normally generated automatically and are transparent to the wallet owner.

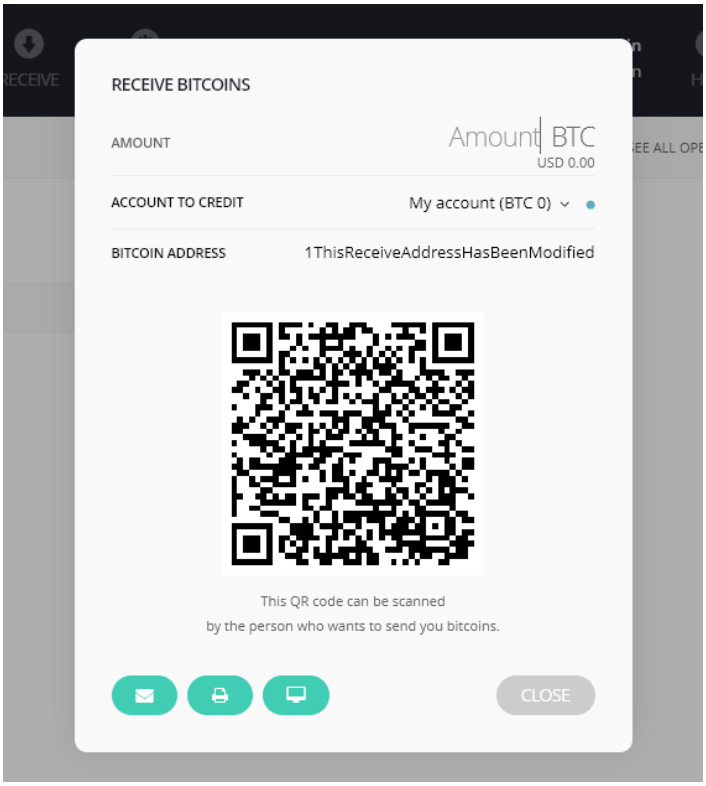
The Attack

Ledger wallets generates the displayed receive address using JavaScript code running on the host machine.

This means that a malware can simply replace the code responsible for generating the receive address with its own address, causing all future deposits to be sent to the attacker.

Because receive addresses are consistently changing as part of the usual activity of the wallet, the user has no trivial way (like recognizing his address) to verify the integrity of the receive address.

As far as he knows, the displayed receive address is his actual receive address.



What Makes This Even Worse

- All the ledger wallet software is located in the AppData folder, meaning that even an **unprivileged** malware can modify them (no need to gain administrative rights).
- The ledger wallet doesn't implement any integrity-check/anti-tampering to its source files, meaning they can be modified by anyone.
- All the malware needs to do is replace one line of code in the ledger software, this can be achieved with less than 10 lines of python code.
- New ledger users would typically send all their funds to the wallet once initialized.
  If the machine was pre-infected, this first transaction may be compromised causing the user to lose all of his funds.
- The attack changes the receive address during its generation, causing even the automatically generated QR to be updated to the attacker's address. Meaning that both the string and QR representations of the address are compromised.

Proof of Concept

Open the file:

> *C:\Users\%USERNAME%\AppData\Local\Google\Chrome\User Data\Default\Extensions\%EXTENSION_ID%\%EXTENSION_VERSION%\src\wallet\wallet.js*

Replace the line:

> *return (_ref = this.wallet.cache) != null ? _ref.get(this.getCurrentPublicAddressPath()) : void 0;*

With:

> *return "MY_MALICIOUS_ADDRESS";*

The next time you receive funds, all the funds will be sent to MY_MALICIOUS_ADDRESS.

Mitigation

Un undocumented feature, that isn't even part of the official "Receiving BTC to your Ledger" article, can in some cases help verify the integrity of the receive address.

On the bottom right part of the receive screen, a small monitor button exists. Pressing this button will cause the receive address to show up on the hardware wallet's screen.

This can be used to verify that the address is valid and has not been tampered.

Note that this process is not part of the default receive process, and is not enforced by the wallet.

A proper solution would be to **enforce** the user to validate the receive address before every receive transaction, just like the wallet **enforces** the user to approve every send transaction.

Also, this undocumented feature only exists in the Bitcoin App.

The Ethereum App (and possibly other apps as well) has no mitigation, the user has no way to validate if the receive address has been tampered.

Advice for Existing Ledger Customers

If you're using the Bitcoin App – Before every receive transaction validate the integrity of the address using the monitor button.

If you're using the Ethereum App – Treat the ledger hardware wallet the same as any other software-based wallet, and use it only on a Live CD operating system that is guaranteed to be malware-free. At least until this issue receives some kind of fix.

Responsible Disclosure

Unfortunately, Ledger doesn't have an organized vulnerability disclosure program.

Nonetheless we contacted the CEO and CTO of Ledger directly in order to privately disclose and fix the issue. We've received a single reply, asking to hand over the attack details. Since then all our mails have been ignored for 3 weeks, finally receiving an answer that they won't issue any fix/change.

Timeline:

4, January, 2018 – First contact with general information.

4, January, 2018 – CTO of Ledger requested the full details of the vulnerability.

4, January, 2018 – Full Details were sent.

10, January, 2018 – We've requested an update, no response.

13, January, 2018 – Again, we've requested an update, not response.

27, January, 2018 – CTO of Ledger replies that no fix/change would be done (our recommendation to enforce the user to validate the receive address has been rejected), but they will work on raising public awareness so that users can protect themselves from such attacks.