Updating a DID with a nym transaction will be written to the ledger if neither ROLE or VERKEY are being changed, regardless of sender.

Moderate **esplinr** published **GHSA-wh2w-39f4-rpv2** on Dec 22, 2020

Package
No package listed

Affected versions
1.12.2

Patched versions
1.12.4

Description

Name

Updating a DID with a nym transaction will be written to the ledger if neither ROLE or VERKEY are being changed, regardless of sender.

Description

A malicious DID with no particular role can ask an update for another DID (but cannot modify its verkey or role). This is bad because:

1. Any DID can write a nym transaction to the ledger (i.e., any DID can spam the ledger with nym transactions).
2. Any DID can change any other DID's alias.
3. The update transaction modifies the ledger metadata associated with a DID.

Expected vs Observed

We expect that if a DID (with no role) wants to update another DID (not its own or one it is the endorser), then the nodes should refuse the request. We can see that requirements in the Indy default auth_rules in Section "Who is the owner" in the last point of "Endorser using".

We observe that with a normal DID, we can update the field `from` for a random DID, for example, the one of a TRUSTEE. It creates then a new transaction on the ledger.

Explanation of the attack

We first begin to connect to the pool and open a wallet. Then, we will use a TRUSTEE (but can also be a STEWARD or an ENDORSER) DID `V4SGRU86Z58d6TV7PBUe6f`. We ask the information about `V4SGRU86Z58d6TV7PBUe6f` with a get-nym. We create a new DID `V4SGRU86Z58d6TV7PBUe1a` signed by `V4SGRU86Z58d6TV7PBUe6f` with no role. For the rest of the attack, we will use `V4SGRU86Z58d6TV7PBUe1a` to sign new transactions. We send a `ledger nym did=V4SGRU86Z58d6TV7PBUe6f extra=hello` to see if `V4SGRU86Z58d6TV7PBUe1a` can send an update of a TRUSTEE identity. When we ask information to the ledger about `V4SGRU86Z58d6TV7PBUe6f`, it answers that the `from` field is `V4SGRU86Z58d6TV7PBUe1a` (to compare with the first get-nym we did with `from` field = `V4SGRU86Z58d6TV7PBUe6f`). To see the log of the attack, I modified my indy-cli to print the json request and the json response directly on the terminal. You can find the log file `indy.log` in this archive.

Implementation notes

NymHandler method `update_state`, line 62. I think that we need to check if the DID which signs the transaction, owns the DID or is its endorser.

Steps to Reproduce

Environment

Ubuntu 18.04
Docker version 19.03.8
indy-cli
indy-ci Dockerfile is copied in this archive
To install indy-cli, run `./install_indy_cli.sh`

Command

Here is the script to create the container, run the attack and remove the container and the image. Find below the command to execute each step separately.

```
./full_attack.sh
```

Installation of the environment

Install indy-cli and create an image with tag `test` from Dockerfile

```
./install.sh
```

Exploit

```
indy-cli proof_of_concept
```

Uninstallation of the environment

Suppress the container `test` and remove the image `test`

```
./uninstall.sh
```

Analysis

We are grateful to **@alexandredeleze** for discovering and responsibly disclosing the issue.

We were previously aware that any DID on the ledger can "update" the state (seqNo + txnTime) if it doesn't change the state data itself. We considered this a minor bug because only the seqNo and txnTime changed. But seeing that this can also affect the "parent" DID means that it has a higher severity.

Severity

Moderate

CVE ID

CVE-2020-11093

Weaknesses

No CWEs

Credits

 **alexandredeleze**