VSW and VSW Repository

Wenjing.Chu@futurewei.com
10/12/2020

vsw & vsw repo

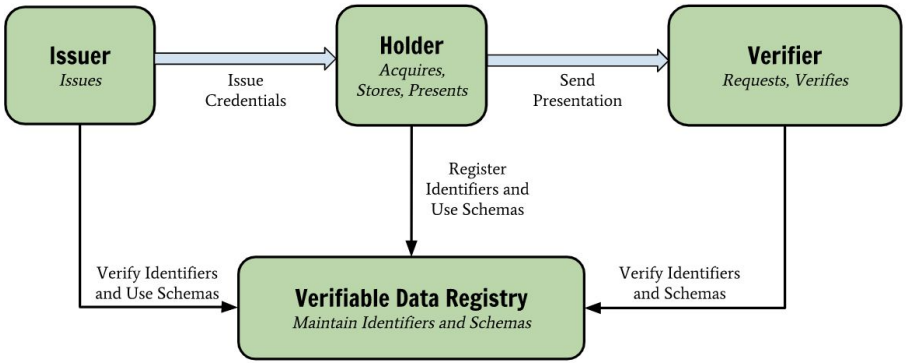
- vsw = verifiable software
  - Software developers, testers, consumers can exchange verifiable information about software therefore it
    - Makes the software ecosystem/supply chain more trustworthy, less prone to various attacks
    - Creates a trusted registry/database of information about software that cuts down the complexity of tracing software and tool dependencies
- vsw-repo = verifiable software repository
  - It is the trusted registry/database of information about software
  - It can also store the software within the repository, or alternatively reference the actual software in another system
  - It is designed as a decentralized system

Decentralized

- Decentralized systems do not have a central “trusted authority”
  - GitHub is not a decentralized system
    - GitHub is owned by a corporation = Microsoft
    - Its proper functioning (as users expect it to function) relies on the good-will and competence of the central operator (e.g. its terms of use policy)
    - The central operator can even go back in time and change data
    - And of course, it can be hacked, or have outages …
  - Sovrin ledger is a decentralized system
    - It is operated by a group of entities (Sovrin Foundation), not just one, who participate in a distributed algorithm to maintain the database
    - Its proper functioning relies on both technical protocol and governance framework of the parties involved, which the user can examine to determine trust levels
    - The data is immutable - no one can go back and change history - unless a big portion of the group collude to commit fraud
    - One party (or small portion of the group) being hacked or suffered outages does not invalidate information or bring down the service

Verifiable

- Verifiable credential relies on cryptographic methods to ensure the information is authentic and trustworthy.
- When coupled with a decentralized ledger and identifiers, it enables the creation and exchange of trusted information without relying on a trusted authority



Briefly compare *vsw* with *npm*

- npm has 3 components
  - Web site npmjs.com
  - Cli command: npm
  - The registry/repo
- A user needs to create an account with npmjs.com (centralized)
  - Npmjs.com determines what user info is required to identify a user, which is often email and domain
- A user publishes a software package with meta information in package.json
  - User's identity is validated only by account login passwd in npmjs.com
  - The info in package.json can be attributed to this user only through trusting npmjs.com
  - The info in package.json uses traditional IDs e.g. URL, email, etc.

- vsw has 3 components
  - Web site verifiablesoftware.com
  - Cli command: vsw
  - The registry/repo
- A user registers with vsw - which creates a DID in Sovrin ledger and binds to this user
  - A published schema in the ledger determines what is required info for a user, which ties to a public key
- A user publishes a software package with meta information in vsw.json (to be named)
  - User's identity is validated through the ledger with no passwd
  - The info in vsw.json is attributed to this user cryptographically
  - The info in vsw.json uses DIDs so that each data points are also verifiable
  - In addition, vsw can support more advanced functions such as zero-knowledge proof for privacy etc.

Briefly compare *vsw* with *npm*

- npm typically adds info about a software package when the developer publishes it
- Enforcement of policies relies on human involvement.

- vsw can be used to publish additional information about an existing software package by anyone, e.g.
  - The original developer can add additional info
  - A tester can publish test result of a third party software
  - An audit can issue certification of a software package
- Because the information in vsw is cryptographically verifiable, it can be used to automatically enforce policies for security or for other reasons, e.g.
  - Only accepts software packages from vendors rated “Gold”
  - Must use software that has up-to-date patches
- More advanced features and also future proof with standards and technology advancements

How vsw solves trust related problems

Motivating Example 1

Threat model:

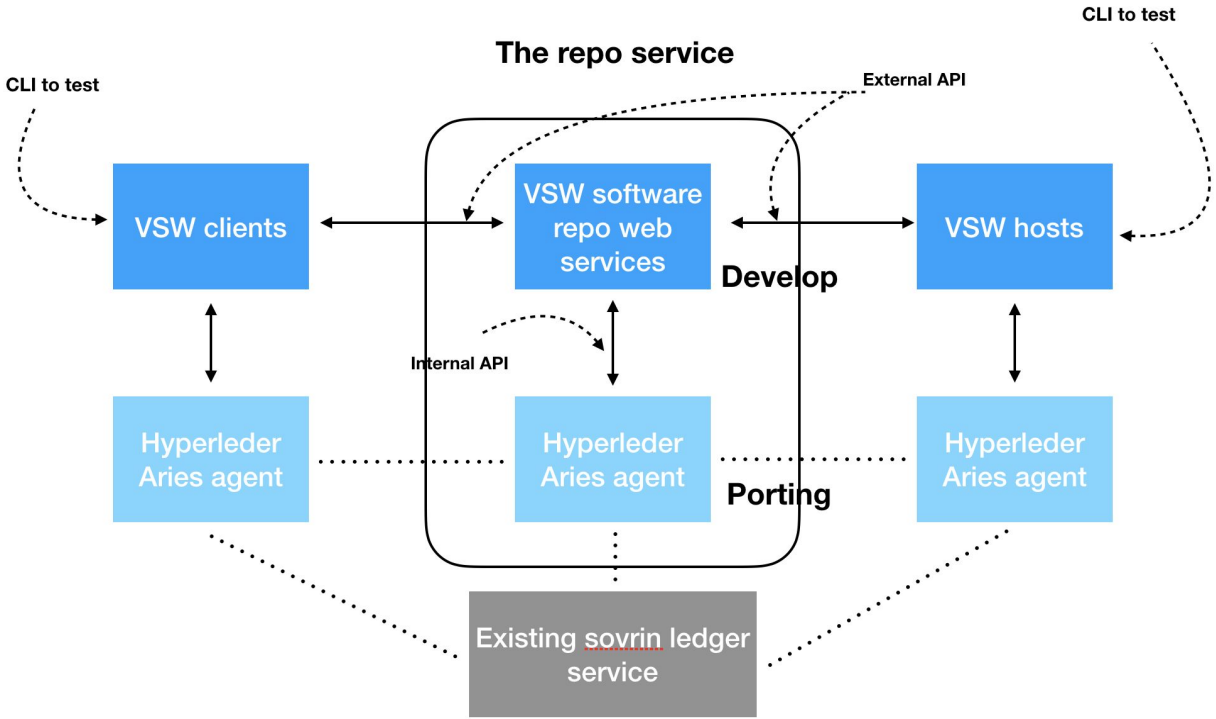
- *Fake developer identity*
- *Avoid reputation of web domain based detection*
- *Avoid human review based detection*

- (*Reuters*) — A newly discovered **spyware** effort attacked users through 32 million downloads of extensions to Google's market-leading Chrome web browser, researchers at Awake Security told Reuters, highlighting the tech industry's **failure to protect** browsers as they are used more for email, payroll, and other sensitive functions.
- It is unclear who was behind the effort to distribute the malware. Awake said the developers supplied **fake contact information** when they submitted the extensions to Google.
- The extensions were designed to **avoid detection** by antivirus companies or security software that evaluates **the reputations of web domains**.
- Malicious developers have been using Google's Chrome Store as a conduit for a long time. After one in 10 submissions was deemed malicious, Google said in 2018 it would improve security, in part by increasing **human review**.

FUTUREWEI Technologies

- The malicious users hide behind one bad domain registrar - it's a weakness of centralized systems
  - They won't be able to in a DID
- Current reputation based detection software is domain based - they can hide behind innocuous domain names
  - Won't be possible in a DID - either they are known with good reputation or they are known to be "anonymous".
- Human review is work intensive - another weakness of centralized system
  - In vsw, the whole community can be reviewers

vsw overview



- vsw has 3 components
  - Web
  - Cli
  - Registry/repo
- The client or host run on local machines (or any endpoint machine)
  - Cli
  - Web api
  - Web api language binding
- The registry/repo/web server runs on cloud
  - Containers in aws
  - Containers in local docker
- Each endpoint (role or actor) integrates Aries agent

Common uses

- vsw register
  - A user (developer, tester, consumer, auditor…) registers its identity with vsw
- vsw publish
  - A developer publishes a new software package
- vsw attest
  - A user (developer, tester, auditor…) publishes additional info for an existing software package
- vsw verify
  - A consumer (or anyone) requests of proof that certain information or condition is true and verifies the replied proof from vsw repo
- vsw list
  - Query the registry
- vsw audit
  - Run a systematic verification
- vsw help

Note that same functionalities are to be implemented in CLI, web API & program language binding.

Note: see github for detailed spec of each.

Q&A

-
-