manySECURED

ManySecured:
Data Sharing Considerations

ManySecured Data Sharing Considerations

26/09/22

Introducing the need for sharing

Data sharing is the bedrock of good cyber security practices. From incident handling, to vulnerability disclosures through to emergent SBOM (software bill of materials) practices; all are examples of public sharing of data for the common good. A cyber threat is a common threat, and therefore it makes eminent sense to collaborate on the solution, sharing the necessary data in the process.

Distributed device descriptors (D3), provide a language for sharing fine grained data, between distributed parties, in a controlled fashion. The cryptographic signing methods employed in D3 provide a method for one party to express information about a threat or device type and another party to act on this data, but with some reasonable assurances of trust, provenance and integrity of the data. And with pre agreed data schemas we also address the interoperability problem.

In this document we will provide an overview of the primary D3 data sharing use cases, highlighting the data sharing and data governance considerations.

To stay consistent with the philosophy of D3 statements (which are built on verifiable credentials) we will use the term “claim” to mean any piece of information asserted by the issuing party, in other words the party initiating the data share and providing the initial piece of information.

Primary Sharing Use cases

One of the first things we need to consider is the overarching use case; who is sharing data with who. Who is the data issuer (disclosing party) and who is the receiving party.

For every scenario we assume that the issuing party is fully identified, and the claim has been digitally signed with by the issuer.

We will consider separately the more advanced use cases where the disclosing party has full or partial anonymity.

For simplicity we will consider three primary data sharing flows.

Public

The first is the simplest. The data issuer places claims in the public domain. There is no protection against data access, and no authentication or authorisation test required to access the information.

From a practical perspective the issuing party can either host the claim themselves, place the information on a public blockchain (or similar) or host the data indirectly through another hosting service (e.g. GitHub).

Public curated

The second scenario is a variation on the first. The issuer has the intention of making the data public (available to all), but this information is mediated through a curating party.

This curating party may make quality decisions on the incoming data and republish the information, with the implied assurances of the curating party.

From a practical perspective the curated assurance could be:

a) Implicit: the curating party confers its assurance though the physical address, and physical access of the claim (ie its hosted on the site)
b) Explicit: the claim is cryptographically countersigned by the curating party

Private club

The third scenario is a disclosure between a privileged set of peers. The disclosing party places the claim within an arena which is protected by some form of access control.

This access control could also be implemented by group based cryptography on the claim.

The assumption is that the information is not in the public domain, but passed to a third party (private club) who will administer the access to this data in the future

Private curated

This is a minor variation on the public curated use case, where the “approved” claims are published to the private audience with the conferred assurances of the curating party.

Peer to Peer

The fourth scenario is a disclosure between the issuer and the receiver. This should be considered a private transaction. No other party is assumed to have access to this information.

Claimant granularity

In the above description we have glibly overlooked the question who exactly is the disclosing party. Here are the primary options we consider:

1. Natural person: a human being who has intentionally made a claim about some piece of security data.
2. Organisation: a claim made by (or on behalf of) an organisation or legal entity
3. Computational agent: a physical electronic device, and/or the software agent running on this device. (This can get quite complex).

Clearly there can be relationships between these claimant levels.

a) At the point of making a claim, a natural person may have a relationship with one or more organisations. (e.g. employed by)

b) Almost always, any claim made by an organisation has been made by, or at least approved by one or more natural persons.

c) Any claim made by a natural person at a point in time will be assembled on and physically signed by a piece of machinery (a computation agent) often in web parlance a user agent.

d) A computational agent: even if “autonomous” is probably owned by an organisation and administered by one or more natural persons.

For reasons of simplicity and practicality, we may often ignore these relationships and accept a coarse grained assertion at face value: e.g. John Smith provided this piece of information. However, part of the elegance of the Verifiable Credential approach is that pertinent security meta data can be either packaged together or cross referenced, allowing us to establish the full “credentials” of the issuing party at the time they made the claim.

These relationships are complex and all are highly material when considering the trustworthiness or security properties of any claim made at any point in time.

Type data

Type data is the lynchpin of the D3 system. An assertion of type is a claim that an abstract physical type of device exists (commonly referred to as a SKU). When asserting a type we provide an immutable GUID and one or more URI, which can be used to refer to this type.

The assertion of type provides a common language for making interoperable security statements: e.g.

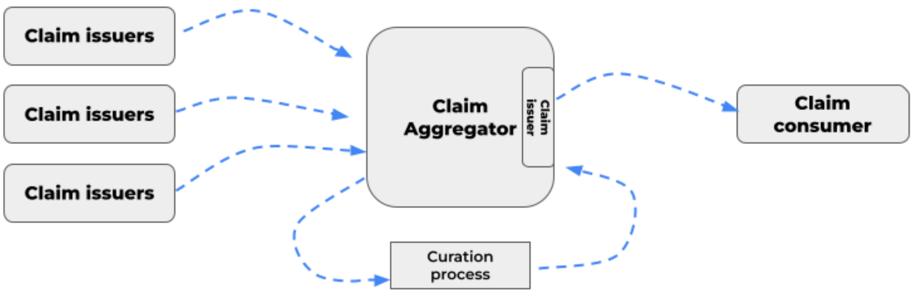
- This device instances is of this **device type**
- This **device type** has this expected behaviour
- A vulnerability has been identified for this **device type**
- This **device type** has these available firmwares

From a data sharing perspective any organisation or natural person can make an assertion that a device type exists.

If we can ascertain that the asserting party is linked to the organisation that physically built the device type, this should confer certain strong assurances to the claim.

However, it is up to the consumer of the claim to establish their own sense of trust in that piece of information.

The claim aggregator becomes a claim issuer in its own right. The aggregated claims may be optionally countersigned by the aggregator



The curation process can arbitrarily complex. In this process the agragrator assesses the claims, resolve any conflicts and essentially selects which claims to accept.

IOTSF statement of intent

It is the clear intent that the IOTSF establish itself as a trusted curator of and publisher of device types, hence a claim aggregator.

The IOTSF ecosystem will support both crowdsourced and manufacture sourced device type assertions.

A governance system will be developed to implement the curation process.

Hierarchy

The hierarchy data is essentially metadata about type. The hierarchy is essentially a relationship between two or more types. The hierarchy relationship is directional with a parent and child type. The instantiation of a hierarchy implies that the child inherits attributes from the parent. Within D3 we assume the hierarchy is typed, which essentially identifies what attributes are to be inherited.

Firmware

From an IOT device perspective we can assume that Firmware is a "special" form of type.

The aggregate behaviour of an IOT device is determined by the physical hardware plus the operational software. (type + firmware). In many senses when a firmware is instantiated (or upgraded) on a physical device it creates a new type variant that many have different behaviours.

A firmware cannot be modelled as just a type however, as a single firmware can be used on many hardware device types.

Data Sharing Considerations - Device Type

For all practical purposes we assume all type data will be public.

Immutable type references are essential for other security data to be dereferenced.

A data aggregator may wish to keep the details of its curation process secretary. And a data issuer may not wish to disclose when, how and the content of any types. But by the time the device type data is consumed we assume that this type information is in the public domain, ideally on a publicly accessible website.

Data Sharing Considerations - Confidential Scenarios

There are a some edge cases where data type information may be held privately

1. When a manufacturer is about to release a new device onto the market, there

may be roadmap sensitivity about when this information is released. But operational security measures may require the information to be shared between parties before release date.

2. Defence equipment. Some device types may be by nature highly confidentials. In these scenarios the type information should only be shared between trusted peers.
3. Value added: there may be scenarios where some combination of the claim issuer and aggregator wish to provide their information on a privileged (for sale basis)

Device instance identity

In contrast to device type, device instance identity data by default should be private.

Device instance identity data refers to all data that identifies or helps identify a specific physical instance of a device. Examples of device instance data could be MAC address, IP address, public key etc.

The two entities that need to know and should know device identity are

- The manufacturer
- The current owner of the device

There are a number of use cases for sharing device instance identity data:

1. Purchase: recording the purchase of advice and tracking ownership

Typically this information should be shared between the manufacturer and device owner.

However, we should understand that by making the device owner a participant in identity based transactions post point of sale, we encourage and enable the mote to rental and licensing based transactions. The common activity of registering a device to activate warranty is an example of using device instance identity information post point of sale. It is possible that anonymised zero knowledge proofs could be used here.

There are also clear use cases for device distributors and retailers having access to device instance identity (or some anonymous proxy thereof), to help manage the supply chain

2. Fraud prevention

The second use case is fraud prevention. A relatively common attack is device identity spoofing, where one device pretends to be another. There can be many incentives for this including counterfeiting. The sharing of device identity instance information can help protect against this. If some central authority can track that two or more device identities are simultaneously active in different locations, then we can assume the integrity of the device identity has been compromised.

There are a number of parties that could act as a “counterfeit protector”, from the originating manufacturer to service provided by intermediate secutroy providers.

3. Downstream licensing

Device instance identity is often used to implement licence protection or forms of DRM. Software for example could be licensed to a single specific device. Or streaming services may be activated to a limited number of client devices. In these examples the device owner will share the device instance identity (or some proxy thereof) with the service provider .

Data Sharing Considerations - Device Instance Identity

From a D3 perspective we assume all device instance identity to be default private and confidential.

Parties may choose disclose device instance identity with third parties under restricted circumstances, for example

1. Sharing instance data for security purposes
2. Sharing in the context of legal transactions eg. purchase

Device instance activity

Device instance activity is highly confidential information. Device instance activity is a record of all data communication made by the device in question whether this be on the internal network or external network (internet), and whether the device instance is the initiator or recipient of the information.

Provisioning access to the complete activity trace of a device could provide a trace of every website visited by a TV for example, or for internal administration traffic (often encrypted) this could provide access to administration passwords in the clear.

Device instance activity can be post processed, removing information in the data flow, which can partially amoloritate any privacy considerations.

For simplicity we will just iterate three levels of device instance activity

1) **Full capture:** this the full and complete capture of all traffic. Typically as an unfiltered PCAP file.
2) **D3 activity schema:** this is the level at which the current D3 system shares information. This is computed from the full capture. It removes the payload (data package content) from the capture. But records essential header data and performs a level of stateful packet analysis identifying the type of traffic.
3) **Connection type profiling:** the third level is a further filtering, where timestamp and individual connection information is removed. But each connection type (source, destination) is recorced, optionally with some form of weighting to indicate the velocity, sequencing or volume of these connections. In this version we can't see the individual connections, but can get a sense of the shape of the connectivity profile of this device instance.

These levels of data sharing are entirely arbitrary, but give us a sense of the continuum.

Sharing data activity is essential in order to identify anomalous behaviour and build profiles of well behaving device types. Working out how much data to share is complex and we need to navigate between:

1) Efficiency: how quickly can the data be stored
2) Storage: how much information do we need to store
3) Privacy: what is the disclose liability of sharing this information

Data Sharing Considerations - Device Instance Activity

The default in D3 is all device instance activity data is private.

In D3 there are two primary use cases for sharing device instance activity

1) Device monitoring: sharing device activity data so that anomalous and threats can be identified
2) Device profiling: sharing data so that profiles/models can be built for device instance and device types

Both of these use cases can be addressed “locally”, requiring no data sharing. However, the router owner could choose to share the activity information to provide better (collaborative) capability of monitoring and profiling.

For operational D3 (VIrtual IOT CyberLab) scenarios we assume the default granularity of data exchange is encapsulated by the formal D3 schema definitions. We assume that this information is shared privately with the aggregator, but derivative information (e.g. device type activity) may be disclosed publicly

Device type behaviour

Device type behaviours are a model of how the device types are expected to behave. Device type behaviours are essential for a good dynamic security implementation.

If a **device instance** behaviour does not map to its known **device type** behaviour THEN

EITHER this **device instance** is behaving anomalously

OR this **device type** behavioural descriptor needs updating

Like device instance behaviours, these behaviours can be modelled at different levels.

Pravicall we will define two crude levels of device behaviour descriptor:

1. **Least privilege behaviour (LPB)**: the least privilege behaviour or the device is a crude delimitation of the behavioural envelope in which we expect all instances of device instances of this device type to sit. Most simply this is a white list of destination source IP addresses including relevant ports. Optionally these connectivity pairs may have a crude type (TCP/DNS/HTTP) etc. This from of type behaviour is specified by IETF as https://datatracker.ietf.org/doc/rfc8520/. Elsewhere this type of descriptor is sometimes referred to as a device intent.
2. **Stochastic descriptor of behaviour (SDB)**: a stochastic descriptor behaviour starts to model deeper aspects of a devices expected behaviour that start to include frequency of connection, volume of connection, sequence of connection and other pertinent attributes. We use the term stochastic descriptor as this model needs to have an explicit or implicit modelling of statistical distribution. These modells could take many forms: Ngram analysis, bayesian model, marcov models through to transformer neural networks.

Data Sharing Considerations - Device Type Activity

It is the clear intent of D3 to help bootstrap the ecosystem to make least privilege descriptors of device type public. (ie make MUD statements ubiquitous and relaibile)

This can be achieved by encouraging manufacturers to publish their own or encourage aggregators to publish reliable LPB.

In general we assume LPB are mostly public. Public good is served by being open on this data

There are a number of ways of generating public reliable LPBs through aggregators.

1. Single submitter: a single submitter may provide a LPB for a device type and the aggregator determines the trustworthiness of this submission. A submission of LPB from the oringiatvin manufacture is a special case of this
2. Multiple LPB submitter: an LPB may be submitted by one or more provides and the aggregator approves the republishing of the approved LPB when a threshold is met
3. Single/multiple SDB submission: in this scenario the submitter provides a richer SDP (statistical description) of device behaviour to the agretnater. This submission flow may be private between the submitter and the aggregator. And the aggregate determines the threshold at which the LPD is published
4. Device instance activity submission: in this scenario the submitter, discloses the activity of the device instance. The aggregator can then build their own SBD and LPB models for the type. This model inherently requires greater trust between submitter and aggregator due to the nature of the information being exchanged

In scenarios 1-3 the submitter is possible for determining the likely device type of a device instance, In scenario 4 the aggregator can perform this function.

Vulnerabilities

The discovery of vulnerabilities is already commonplace in the industry. (see https://cve.mitre.org/ etc)

It is interesting to review the vulnerability information flow from the perspective of data sharing and public disclosure. The existing flows echo some of the security and privacy considerations highlighted above.

Responsible disclosure of a a vulnerability relates to the convention of the claim issuer (security researcher/discoverer of the vulnerability) passing the information directly to the originating manufacture confidentially, usually for a discretionary period. This is to allow the responsible party (manufacture in most instances) time to address the vulnerability before putting in in the public domain where bad actors could exploit the vulnerability before it fixed. Contrariwise public good is served by putting this information in the public domain so that impact parties can deal with the downstream consequence.

Data Sharing Considerations - Device Vulnerabilities

In this document we do not address the pros/cons, timing or audience of vulnerability disclosure. This a complex issue addressed already by industry.

We do strongly advocate the use of immutable, dereferenceable URIs for device types. This is a primary motivation of the D3 system and clarifying these references offers substantive benefits to the community as a whole.

It is possible the models of issuer, aggregate and the distributed nature of D3 Claims, and the ability to support selective disclose and encryption could offer substantive benefits, or augment current disclosure models. This is an area of further research.

Suspicious activities

Where as vulnerabilities are security concerns attributed to **device types**; a suspicious activity database provides us with **patterns of behaviour** matches we can apply to **device instance**, A positive match against one of these "anti-patterns" providers a strong indicate that something has been compromised.

There are many potential source of this information including

https://www.misp-project.org/

Data Sharing Considerations - Suspicious activities

Suspicious activity databases may be open source or proprietary. In general we assume public good is best served by making this formation public.

An active D3 system can benefit by including such pattern matching capabilities, and this can be incorporated into dynamic security assessments.

We would encourage reuse of the underlying D3 primitives, e.g declaration of types, instances and description of behaviour to help with integration and data

Conclusion

In the above document we examine the primarily data flows and privacy considerations for the D3 ecosystem. We have the following high level summary of key convulsions. ##

1. The D3 system supports data claims to be issued by originator or an aggregator.
2. It is always up to the D3 consumer to attribute their own sense of trust in the data provide.
3. The two primary objective of D3 are to encourage
    a. Use of interoperable D3 Device types URIS to make reasoning about the security status of individual device instance
    b. User D3 least privilege behaviours (MUD statements) to limit device isn't behaviours to secure envelopes
4. D3 Device Types and D3 Device Type Behaviours (LPB) should be open and public.
5. There are a number of ecosystems that lead to the outcomes declared in 3 and 4, some of these will need a secure ecosystem to share data between data issues and aggregators. This document highlights some of the requirements and considerations on these data flows.