Supply Chain Integrity, Transparency, and Trust (SCITT)

This session is being recorded
Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam ([https://www.ietf.org/contact/ombudsteam/](https://www.ietf.org/contact/ombudsteam/)) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- BCP 9 (Internet Standards Process)
- BCP 25 (Working Group processes)
- BCP 25 (Anti-Harassment Procedures)
- BCP 54 (Code of Conduct)
- BCP 78 (Copyright)
- BCP 79 (Patents, Participation)
- [https://www.ietf.org/privacy-policy/](https://www.ietf.org/privacy-policy/) (Privacy Policy)
Note Really Well

- IETF meetings, virtual meetings, and mailing lists are intended for professional collaboration and networking, as defined in the IETF Guidelines for Conduct (RFC 7154), the IETF Anti-Harassment Policy, and the IETF Anti-Harassment Procedures (RFC 7776). If you have any concerns about observed behavior, please talk to the Ombudsteam, who are available if you need confidentiality to raise concerns confident about harassment or other conduct in the IETF.

- The IETF strives to create and maintain an environment in which people of many different backgrounds and identities are treated with dignity, decency, and respect. Those who participate in the IETF are expected to behave according to professional standards and demonstrate appropriate workplace behavior.

- IETF participants must not engage in harassment while at IETF meetings, virtual meetings, social events, or on mailing lists. Harassment is unwelcome hostile or intimidating behavior—in particular, speech or behavior that is aggressive or intimidates.

- If you believe you have been harassed, notice that someone else is being harassed, or have any other concerns, you are encouraged to raise your concern in confidence with one of the Ombudspersons.
IETF 119 Meeting Tips

In-person participants

- Make sure to sign into the session via Datatracker or the QR Code in this session.
- Use Meetecho (usually the “Meetecho lite”) client to:
  - join the mic queue
  - participate in shows of hands
- *Keep audio and video off if not using the onsite version.*

Remote participants

- Make sure your audio and video are off unless you are chairing or presenting during a session.
- Use of a headset is strongly recommended.
Resources for IETF 119 Brisbane

- Agenda
  https://datatracker.ietf.org/meeting/agenda
- Meetecho and other information:
  https://www.ietf.org/how/meetings/preparation
- If you need technical assistance, see the Reporting Issues page:
  http://www.ietf.org/how/meetings/issues/
Any Volunteers?
Agenda

- Welcome and Introduction (5 min): Chairs
- SCITT Overview (10 Mins) Steve
- Document Status (5 Mins) Orie
- Recap since 118 (10 Mins) Henk
- SCRAPI (10 Mins) Jon
- Hackathon Report (20 min): Jon, Corey, Ray
- Next Steps (10 min): Chairs
- AOB Open Mic (20 min – BE CONCISE!): All
- Wrap-up and Conclusion (5 min): Chairs
SCITT Overview

Steve
What is SCITT

- **Working Group Charter**
  The Supply Chain Integrity, Transparency, and Trust (SCITT) WG will define a set of interoperable building blocks that will allow implementers to build integrity and accountability into software supply chain systems to help assure trustworthy operation.
Identities make Statements about Artifacts
Signed Statement = #6.18(COSE_Sign1)

Envelope : COSE_SIGN1
protected : bstr . cbor Protected_Header, 
unprotected : Unprotected_Header, 
payload : bstr / nil, 
signature : bstr

Protected_Header
&(CWT_Claims : 15) => CWT_Claims
? &{alg : 1} => int
? &{(content_type: 3)} => tstr / uint
? &{kid : 4} => bstr
? &{(x5chain : 33)} => COSE_X509
? &{(x5t : 34)} => COSE_CertHash
* int => any

Unprotected_Header
? &{(receipts : 394)} => [+ Receipt]
* int => any

Issuer

Statement

Artifact
SCITT Documents

Orie
- The Architecture draft was refactored, pulling out Receipts, COSE Hash Envelope and SCRAPI into separate drafts.
- The Architecture has been our focus, building consensus to focus on SCRAPI
- Debate amongst the authors if we should publish the Use Cases, or use them as reference to conversations
Recap Since 118

Henk
Architecture Updates: Shaping a Simple Core Document

- Clarify: An Append-only Log is a Verifiable Data Structure, returning Receipts
  - COSE Receipts
- Returning Receipt are Async Operations
- Terminology Simplification: Consumer, Verifiers, Relying Parties
- Focusing on X.509 Identity (RFC9360)
- Clarifying Initialization of the Append-only Log
- Refined the Auditor Role
- Clarifying Feed and Subject
- Deferred Federation
- Examples for Remote Statement Signing
SCRAPI:

SCITT REST API

Jon
SCRAPI: Why do it?

- A charter deliverable
- Helps address confusion regarding "Transparency Services" and "Adjacent Services"
- Enable interoperability in at least 1 protocol
SCRAPI: What does it do?

- HTTP APIs for Transparency Service-related Operations
  - POST Signed Statement (SBOMs, Attestations, End Of Life (EOL), New Versions, …)
  - GET Receipt (Signed Proof of Inclusion)
  - GET Signed Statements (Signed SBOMs, Signed Attestations, Signed EOL, Signed New Versions, …)
  - GET Statements (SBOMs, Attestations, EOL, New Versions, …)
  - GET Issuer Metadata
    - GET Verification Keys (for Signed Statements)
  - GET Service Metadata (operating profile)
    - GET Registration Policy
    - GET Verification Keys (for Receipts)
Only 2 mandatory endpoints

**GET** /*.well-known/transparency-configuration HTTP/1.1
Host: transparency.example
Accept: application/json

**POST** /entries HTTP/1.1
Host: transparency.example
Accept: application/json
Content-Type: application/cose
Payload (in CBOR diagnostic notation)

```
18([             // COSE Sign1
  h'a1013822',   // Protected Header
  {},            // Unprotected Header
  null,          // Detached Payload
  h'269cd68f4211dffc...0dcb29c' // Signature
])
```
POST /signed-statements/issue HTTP/1.1
Host: transparency.example
Accept: application/json
Content-Type: application/vc+ld+json

Payload

{
    "@context": [
        "https://www.w3.org/ns/credentials/v2",
        "https://www.w3.org/ns/credentials/examples/v2"
    ],
    "id": "https://transparency.example/credentials/1872",
    "type": ["VerifiableCredential", "SensorCredential"],
    "issuer": "https://transparency.example/device/1234",
    "validFrom": "2010-01-01T19:23:24Z",
    "credentialSubject": {
        "type": "Feature",
        "geometry": {
            "type": "Point",
            "coordinates": [125.6, 10.1]
        },
        "properties": {
            "name": "Dinagat Islands"
        }
    }
}
Statement Resolution

GET /statements/urn...qnGmr1o HTTP/1.1
Host: transparency.example
Accept: application/pdf

GET /signed-statements/urn...qnGmr1o HTTP/1.1
Host: transparency.example
Accept: application/cose

GET /receipts/urn...qnGmr1o HTTP/1.1
Host: transparency.example
Accept: application/cose
GET /well-known/issuer/tenant/1234 HTTP/1.1
Host: transparency.example
Accept: application/json

HTTP/1.1 200 Ok
Content-Type: application/json

{  "issuer": "https://transparency.example/tenant/1234",  "jwks": [{  "keys": [    {      "kid": "urn:ietf:params:oauth:jwk-thumbprint:sha-256:DgyowWs04gfVRim5i1WLQ-HFFFKI6ltqulj1rXPagRo",      "alg": "ES256",      "use": "sig",      "kty": "EC",      "crv": "P-256",      "x": "p-kZ4u0AT9IjQ9TrWkelnbGb-z3LUIltwRjZa059w",      "y": "ymXElytJcQgjQ5Re9Nwe5TlSUALYzTzy83NVfdg0"    },    {      "kid": "urn:ietf:params:oauth:jwk-thumbprint:sha-256:4FzxBk01W0ob9CZNC3RjX28Ipgy93AFM8jyXKKW0CLE",      "alg": "HPKE-Base-P256-SHA256-AES128GCM",      "use": "enc",      "kty": "EC",      "crv": "P-256",      "x": "Vreuil95vzR6ixutqBFf2ota-rj97MrKfuJWB4qpp5w",      "y": "NKJTea0N1LRRsVRxHGDA-RsA0ex2tSp9d3G-45mKXbs"    }  ]  }]
GET /tenant/1234/did.json HTTP/1.1
Host: transparency.example
Accept: application/did+ld+json

HTTP/1.1 200 Ok
Content-Type: application/did+ld+json

{
  "@context": [
    "https://w3.org/ns/did/v1",
    {
      "@vocab": "https://vocab.transparency.example#"
    }
  ],
  "id": "did:web:transparency.example:tenant:1234",
  "verificationMethod": [
    {
      "id": "did:web:transparency.example:tenant:1234\#urn:ietf:params:oauth:jwk-thumbprint\:sha=256:5b30k1hBWBMCggA9GIJIMUqj4pxQrJ9EOWV6MDcstA",
      "type": "JsonWebKey",
      "controller": "did:web:transparency.example:tenant:1234",
      "publicKeyJwk": {
        "kid": "urn:ietf:params:oauth:jwk-thumbprint\:sha=256:5b30k1hBWBMCggA9GIJIMUqj4pxQrJ9EOWV6MDcstA",
        "alg": "ES256",
        "use": "sig",
        "kty": "EC",
        "crv": "P-256",
        "x": "9ptuW0PBHTSN7bVexWd7xM5kmSPGRCu-K35Lt1yvNc",
        "y": "l7NvF6QbovicSciZqu_M_xy4JT2wtnUbn2SNdMKoaAk"
      }
    },
    {
      "id": "did:web:transparency.example:tenant:1234\#urn:ietf:params:oauth:jwk-thumbprint\:sha=256:5b30k1hBWBMCggA9GIJIMUqj4pxQrJ9EOWV6MDcstA",
      "type": "JsonWebKey",
      "controller": "did:web:transparency.example:tenant:1234",
      "publicKeyJwk": {
        "kid": "urn:ietf:params:oauth:jwk-thumbprint\:sha=256:5b30k1hBWBMCggA9GIJIMUqj4pxQrJ9EOWV6MDcstA",
        "alg": "ES256",
        "use": "sig",
        "kty": "EC",
        "crv": "P-256",
        "x": "9ptuW0PBHTSN7bVexWd7xM5kmSPGRCu-K35Lt1yvNc",
        "y": "l7NvF6QbovicSciZqu_M_xy4JT2wtnUbn2SNdMKoaAk"
      }
    }
  ]
}
SCRAPI: Status

- 00 adopted by working group December 2023
- 01 published March 2024
  - We have been focusing on the Architecture to create consensus
  - Needs reviews
  - We expect substantial changes as folks start looking at it.

SCRAPI: Next Steps

- Raise your hand if you can provide a review of SCRAPI

- Tests to align with COSE Receipts
  - [COSE Receipts](https://datatracker.ietf.org/doc/draft-ietf-scitt-scrapi/)
Hackathon Report

Jon, Corey, Ray
Experience from the Hackathon

Jon
Hackathon -119

- Local and Remote Hacking
  - SCRAPI Focus
  - DigiCert Remote Signing
  - National Vulnerability Database
  - Exploration: Election Data Use Case
Jon Geater reviewed the SCITT architecture, SCRAPI documents, and RFC3552 to fill in the required elements of SCRAPI:

- Probably more to add for general usefulness, but looks complete :-)
- Maintains appropriate separation of concerns between the interoperable client and the architecture
- Reinforces the separation between Issuer and client

https://github.com/ietf-wg-scitt/draft-ietf-scitt-scrapi/pull/5
Remote Signing of Statements

- **Integration of DigiCert: Software Trust Manager**
  - Implemented a GitHub SCITT Action to remote sign statements
    https://github.com/digicert/scitt-action/
  - Validated how to integrate the X.509 properties
  - Discussion on whether to include the x5chain, and where to include it

```plaintext
Protected_Header
& (CWT_Claims : 15) => CWT_Claims
? & (alg : 1) => int
? & (content_type: 3) => tstr / uint
? & (kid : 4) => bstr
? & (x5chain : 33) => COSE_X509
? & (x5t : 34) => COSE_CertHash
* int => any

Protected_Header
{
  15: {
    1: '16',
    2: synsation.io/website-v1.0,
  },
  1: 1,
  3: application/json,
  4: h'63363333...33633338',
  34: h'-16',
}

Unprotected_Header
{
  33: [x5chain],
  x5chain
}
```
Identified two security levels for Election Data use-case

- **Level 1: Post Election, aggregated security**
  - Jurisdictions create a hash-manifest file of all election files and then sign and submit this signed statement to a SCITT transparency service.

- **Level 2: Fine-grained, ballot-level security**
  - Jurisdictions create a public-key manifest of all ballot scanners and create hash manifest of this and other files to the SCITT transparency service prior to the election. (This requires new protocol adoption to obtain public keys of air-gapped devices, outside SCITT scope).
  - During the election, images of each ballot are hashed and signed by the ballot scanner using private keys internally generated by that device, and these signed statements are included in the Level 1 data.
Next Steps

Chairs
SCITT Drafts

- Software Supply Chain Uses Cases
  https://datatracker.ietf.org/doc/draft-ietf-scitt-software-use-cases/

- SCITT Architecture
  https://datatracker.ietf.org/doc/draft-ietf-scitt-architecture
  - Looking for more review
  - Consider Working Group Last Call

- SCITT Reference API (SCRAPI) – Next Focus of Authors
AOB (Open Mic)
Wrap-Up