STIX and TAXII 2.0

Looking Ahead

HS SEDI
Homeland Security Systems Engineering and Development Institute

Homeland Security

STIX, TAXII, and CybOX in OASIS

- **All decisions are made via the OASIS process**
  - Either consensus or explicit votes
- **DHS or MITRE are not “in charge”**
  - Though, still very active and in some leadership positions in the OASIS committee
- **Aggressive timelines for next versions**
  - Early summer for STIX 2.0, TAXII 2.0, and CybOX 3.0

We need your help!

- **If you can or do belong to OASIS**
  - Join the TC and contribute
  - The more comments, the better the specifications
- **If you don't and can't join OASIS**
  - You can still review the specs and submit comments via cti-comment@lists.oasis-open.org
- **Or just come see me after ☺**

Where it's headed

- **Simplification and intuitiveness**
  - One way of doing things
  - Less complicated approaches and terminology
  - Less flexibility, more standardization
  - Less abstraction, more top-level objects
- **More expressive analysis**
  - Better support real analysis use cases
  - Explicitly model as a graph

Switching to JSON

- **XML is fine**
  - STIX also used fairly complicated XML
- **JSON is more natural for many developers**
  - And we'll use simple JSON
  - Validate with JSON Schema
- **Better because:**
  - No element/attribute distinctions
  - No namespaces

Note

JSON is "Mandatory to Implement", but other serializations are still possible

What do **you** think?

Things are required

- **Almost all fields in 1.x are optional**
  - Easier for producers, very difficult for consumers

- **Examples**
  - id
  - created_at
  - (more to come)

What do **you** think?

Splitting apart TTP and Exploit Target

- **In STIX 1.x, TTP and Exploit Target were containers for types**
  - But this wasn’t clear in the spec
- **In STIX 2.0, these containers are removed and types become top-level objects**
- **Better because:**
  - More intuitive to create and use a Malware object than a TTP with a Behavior, that has a Malware Instance
  - Prevents you from creating ambiguous content

What do **you** think?

Extract Relationship to the top-level

- **In STIX 1.x, relationships were embedded in top-level objects**
- **2.0 is explicitly graph-based, with relationships at their own object**
- Better because:
  - Easier to parse
  - Can be represented separately and created by many producers
  - Prevents the “embed vs. reference” debate

What do **you** think?

Consolidate CybOX Patterning

- **In STIX 1.x, CybOX patterns were embedded in all CybOX object fields and had significant duplication of functionality**
- **In 2.0, patterns are extracted out of objects and consolidated to a single (yet to be defined) approach**
- Better because:
  - Less duplication
  - Less pollution of the CybOX object fields model with things used only for patterning

What do **you** think?

TAXII Collections and Channels

- **In TAXII 1.1, everything was a data set**
- **In 2.0, there are two patterns:**
  - Collections, which are data sets for sharing content
  - Channels, which are used for sharing "topic" messages
- Better because:
  - Explicitly supports two design patterns in optimized ways

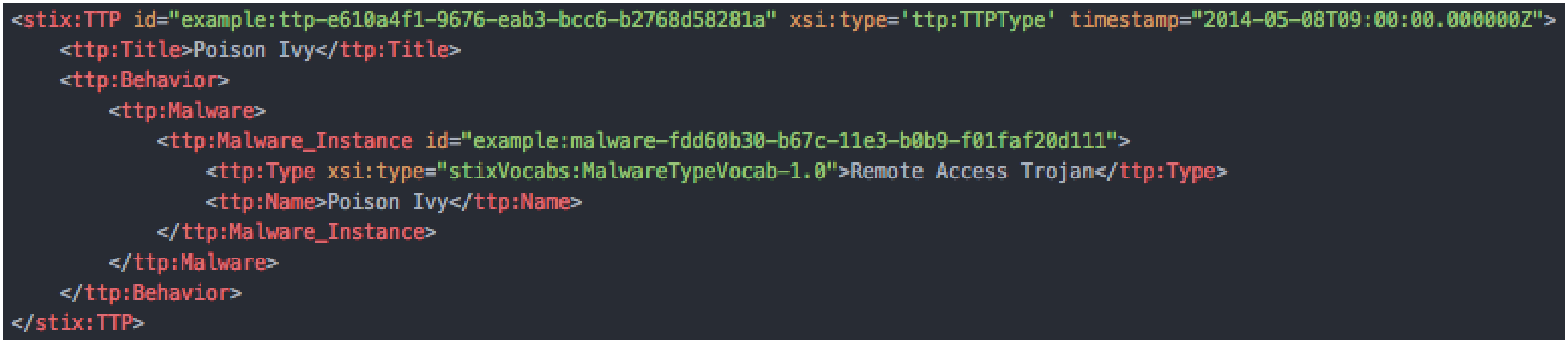
What do **you** think?

TAXII is HTTP and JSON

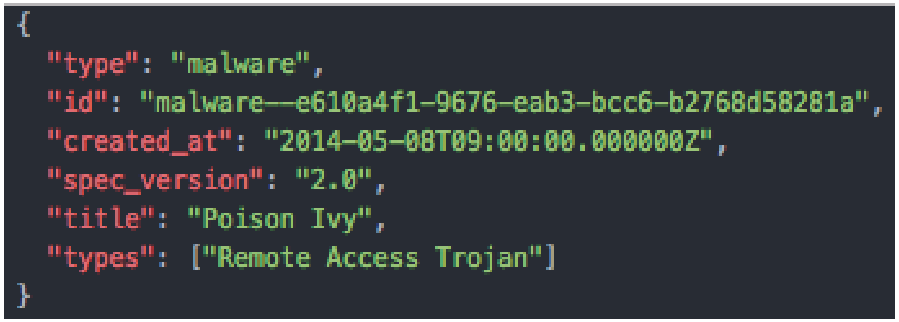
- **In TAXII 1.1, HTTP and XML were bindings of an abstract model**
- **In 2.0, TAXII is explicitly tied to HTTPS and JSON**
- Better because:
  - Everyone is using HTTPS, so it’s less complicated to have the abstraction layer
  - Allows it to take advantage of native HTTPS features

What do **you** think?

Before & After: Malware

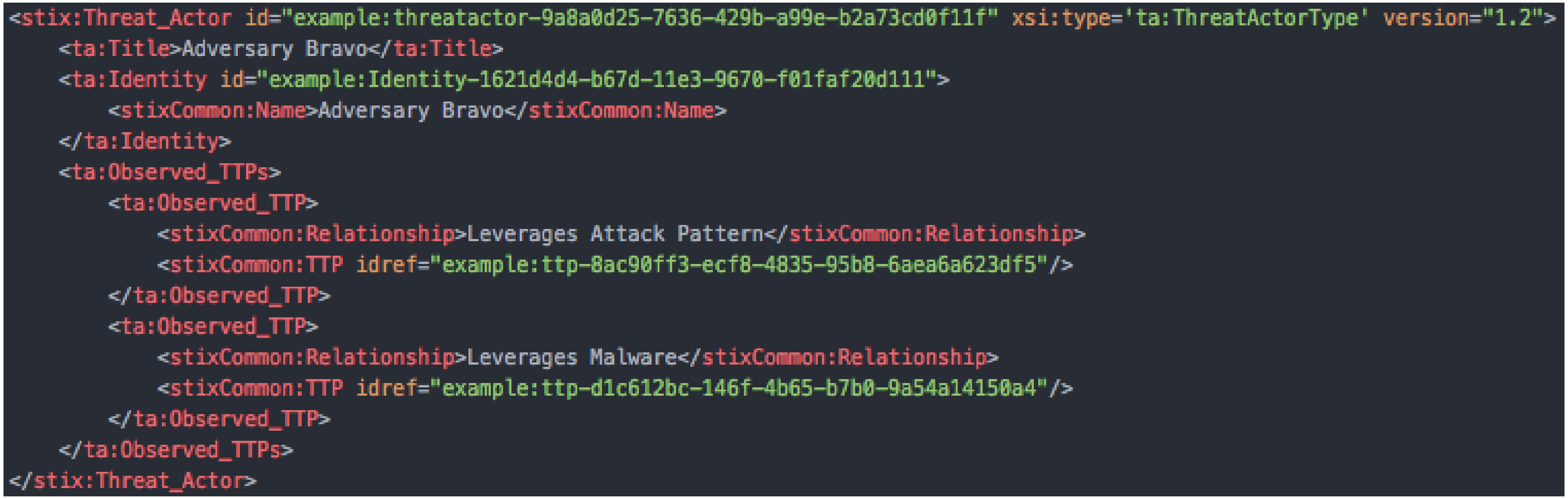


```
<stix:TTP id="example:ttp-e610a4f1-9676-eab3-bcc6-b2768d58281a" xsi:type='ttp:TTPType' timestamp="2014-05-08T09:00:00.000000Z">
    <ttp:Title>Poison Ivy</ttp:Title>
    <ttp:Behavior>
        <ttp:Malware>
            <ttp:Malware_Instance id="example:malware-fdd60b30-b67c-11e3-b0b9-f01faf20d111">
                <ttp:Type xsi:type="stixVocabs:MalwareTypeVocab-1.0">Remote Access Trojan</ttp:Type>
                <ttp:Name>Poison Ivy</ttp:Name>
            </ttp:Malware_Instance>
        </ttp:Malware>
    </ttp:Behavior>
</stix:TTP>
```



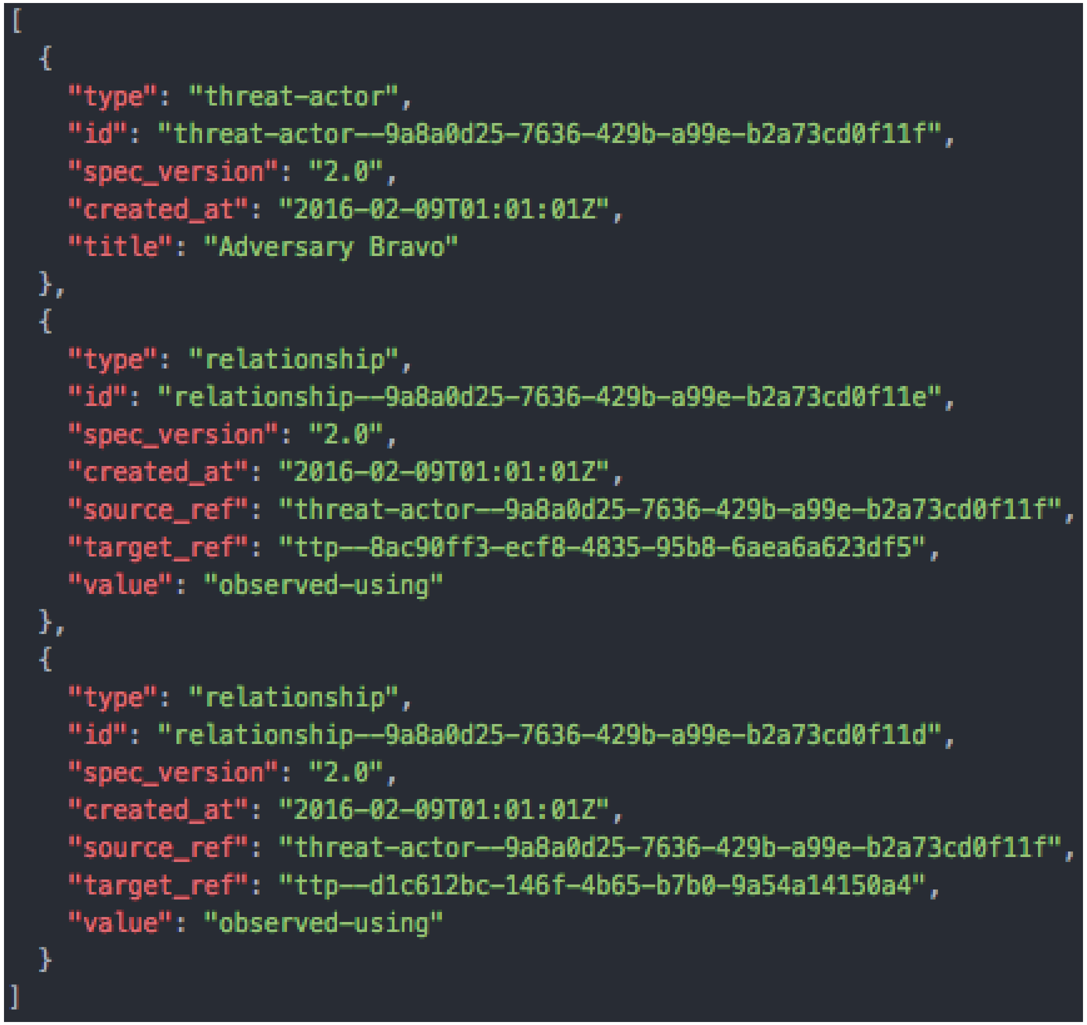
```
{
  "type": "malware",
  "id": "malware--e610a4f1-9676-eab3-bcc6-b2768d58281a",
  "created_at": "2014-05-08T09:00:00.000000Z",
  "spec_version": "2.0",
  "title": "Poison Ivy",
  "types": ["Remote Access Trojan"]
}
```

Before: Relationships



```
<stix:Threat_Actor id="example:threatactor-9a8a0d25-7636-429b-a99e-b2a73cd0f11f" xsi:type='ta:ThreatActorType' version="1.2">
    <ta:Title>Adversary Bravo</ta:Title>
    <ta:Identity id="example:Identity-1621d4d4-b67d-11e3-9670-f01faf20d111">
        <stixCommon:Name>Adversary Bravo</stixCommon:Name>
    </ta:Identity>
    <ta:Observed_TTPs>
        <ta:Observed_TTP>
            <stixCommon:Relationship>Leverages Attack Pattern</stixCommon:Relationship>
            <stixCommon:TTP idref="example:ttp-8ac90ff3-ecf8-4835-95b8-6aea6a623df5"/>
        </ta:Observed_TTP>
        <ta:Observed_TTP>
            <stixCommon:Relationship>Leverages Malware</stixCommon:Relationship>
            <stixCommon:TTP idref="example:ttp-d1c612bc-146f-4b65-b7b0-9a54a14150a4"/>
        </ta:Observed_TTP>
    </ta:Observed_TTPs>
</stix:Threat_Actor>
```

After: Relationships



```
[
  {
    "type": "threat-actor",
    "id": "threat-actor--9a8a0d25-7636-429b-a99e-b2a73cd0f11f",
    "spec_version": "2.0",
    "created_at": "2016-02-09T01:01:01Z",
    "title": "Adversary Bravo"
  },
  {
    "type": "relationship",
    "id": "relationship--9a8a0d25-7636-429b-a99e-b2a73cd0f11e",
    "spec_version": "2.0",
    "created_at": "2016-02-09T01:01:01Z",
    "source_ref": "threat-actor--9a8a0d25-7636-429b-a99e-b2a73cd0f11f",
    "target_ref": "ttp--8ac90ff3-ecf8-4835-95b8-6aea6a623df5",
    "value": "observed-using"
  },
  {
    "type": "relationship",
    "id": "relationship--9a8a0d25-7636-429b-a99e-b2a73cd0f11d",
    "spec_version": "2.0",
    "created_at": "2016-02-09T01:01:01Z",
    "source_ref": "threat-actor--9a8a0d25-7636-429b-a99e-b2a73cd0f11f",
    "target_ref": "ttp--d1c612bc-146f-4b65-b7b0-9a54a14150a4",
    "value": "observed-using"
  }
]
```