TAXII: An Overview

The MITRE Corporation
What is TAXII?

- **Trusted Automated eXchange of Indicator Information**
- Defines a set of services and message exchanges for exchanging cyber threat information
  - led by DHS
  - managed by MITRE
- **TAXII is NOT**
  - A specific sharing initiative
    - but sharing initiatives can use it
  - A specific tool
    - but tools can use it to share information
  - Mandate particular trust agreements or sharing
    - instead, use it to share what you want with the parties you choose
Why was TAXII Created?

- DHS identified an operational need for sharing
  - Existing standards did not solve the problem
- DHS wanted the solution to be open
  - Solution can improve national cyber security, which is part of the DHS mission
- DHS funded MITRE to facilitate a community-based solution
  - TAXII developed with feedback from a broad array of potential users
  - Ensures that TAXII addresses real challenges in real environments
Design Philosophy

- Facilitate rather than re-architect existing sharing arrangements
- Do not force inclusion of undesired capabilities
- Sharing communities each have their own character – a one-size-fits-all approach will be more disruptive than beneficial
  - Do not impose a single sharing model architecture
TAXII Requirements

- **Minimal requirements imposed on data consumers**
  - Does not require data consumers to field internet services or establish a particular security capability

- **Minimal data management requirements on data producers**
  - Does not require use of particular data management technologies or constrain how producers manage access to their data

- **Flexible sharing model support**
  - Does not force a particular sharing model on users

- ** Appropriately secure communication**
  - Supports multiple security mechanisms without forcing adoption of unnecessary measures

- **Push and Pull content dissemination**
  - Users can exchange data using either or both models

- **Flexible protocol and message bindings**
  - Does not require a particular network protocol or message format
Flexible Sharing Models

- Most sharing models are variants of these three basic models
  - TAXII services can support participation in any of these models or multiple models simultaneously
TAXII Services

- TAXII defines four core services
  - Discovery – A way to learn what services an entity supports and how to interact with them
  - Feed Management – A way to learn about and request subscriptions to data feeds
  - Inbox – A way to receive pushed content (push messaging)
  - Poll – A way to request content (pull messaging)

- Each service is optional – implement only the ones you wish
- Services can be combined in different ways for different sharing models
Hub & Spoke Example

- **Hub**
  - Push new data to the hub
  - Pull recent data from the hub

- **Spoke 1**
  - Get connection info
  - Subscribe to data feeds

- **Spoke 2**
  - Push new data to the hub

- **Spoke 3**
  - Push recent data to a spoke

- **Spoke 4**
  - Pull recent data from the hub

**Key Components:**
- **Discovery**
- **Feed Manage.**
- **Poll**
- **Inbox**
- **Client**
Message and Protocol Bindings

- **TAXII does not require a specific protocol or message format**
  - TAXII 1.0 defines how to express messages in XML
  - TAXII 1.0 defines how to transport messages over HTTP
  - TAXII does not require these and allows future and custom formats and protocols

- **“Binding” specifications**
  - Map the core TAXII Services Specification to an expression using a particular network protocol or message format

- **Gives organizations with strict network/format requirements ability to still use TAXII**
  - When using differing bindings, direct interoperation is not possible
  - Indirect interoperation is possible because all bindings can be mapped to the same core message model
  - Gateways can provide automated translation between bindings
TAXII Specifications and Documentation

- **TAXII Overview**
  - Defines the primary concepts of TAXII

- **Services Specification**
  - Defines TAXII Services, as well as the information conveyed by TAXII Messages and TAXII Message Exchanges.

- **Message Binding Specification**
  - Defines normative requirements for representing TAXII Messages in a particular format (e.g., XML).

- **Protocol Binding Specification**
  - Defines normative requirements for transporting TAXII Messages over some network protocol (e.g., HTTP).

- **Content Binding Reference**
  - Lists Content Binding IDs for use within TAXII.
Normative TAXII Specifications

TAXII Services Specification
- Defines TAXII Services
- Defines TAXII Message Types
- Defines TAXII Message Exchanges

TAXII Protocol Binding Specifications
- Define requirements for network transport of TAXII Messages

TAXII Message Binding Specifications
- Define TAXII Message format bindings
For more information

- **TAXII Website**
  - Contains official releases and other info
  - [http://taxii.mitre.org/](http://taxii.mitre.org/)

- **Sign up for the TAXII Discussion and Announcement mailing lists**
  - [http://taxii.mitre.org/community/registration.html](http://taxii.mitre.org/community/registration.html)

- **Open issues can be discussed on GitHub**
  - [https://github.com/TAXIIProject/TAXII-Specifications](https://github.com/TAXIIProject/TAXII-Specifications)

- **TAXII-related software can be found on GitHub**
  - [https://github.com/TAXIIProject](https://github.com/TAXIIProject)

- **Related sites**
  - [https://stix.mitre.org/](https://stix.mitre.org/)
Additional TAXII Details and Examples
TAXII Feed Management Service

- Hosted by data producers
- Receives queries about offered TAXII data feeds
  - Provides feed names and descriptions
  - How TAXII data feed content can be accessed ("pull" or indicate delivery protocols)
  - Any other information about a TAXII data feed (e.g., membership requirements, payment requirements, etc.)
- Receives requests to manage TAXII data feed subscriptions
  - Subscribe, unsubscribe, pause delivery, resume delivery, modify subscription, status query
  - TAXII does not specify the process for deciding whether to allow the requested action to occur nor how the action manifests
- Note this just deals with arranging subscriptions, not actual data dissemination
TAXII Data Feeds

- TAXII does not dictate how data producers store or organize their data...
  ...but TAXII requires some common handle for communication.
- TAXII Data Feed – a producer-dictated organization of their data
  - A given data record might exist in one or more TAXII data feeds
  - Producers decide what data feeds represent. Examples:
    - Topic – e.g., a feed for spear-phishing, a feed for botnets, etc.
    - Subject – e.g., a feed for each identified STIX campaign
    - Access – e.g., a feed for gold-level subscribers, a feed for silver-level, etc.
    - Or producer might just have one feed with everything in it
- In TAXII, all solicited data distribution (push or pull) occurs relative to a TAXII Data Feed
TAXII Inbox, Poll, and Discovery Services

- **Inbox Service**
  - Hosted by consumers to receive pushed content
  - Basically a listener for incoming content

- **Poll Service**
  - Hosted by data producers
  - Consumers request updates relative to a TAXII data feed
  - To support this, TAXII requires all records within a TAXII data feed to be assigned a timestamp
    - Data producers can decided the meaning, if any, of the timestamp
    - Poll requests indicate a range of timestamps to collect
    - Poll responses identify returned range – recipient can track to avoid re-requesting content

- **Discovery Service**
  - Identify services and how to contact them
Source/Subscriber Walkthrough
Background

- One possible way to use TAXII to implement Source/Subscriber
  - Others may make different choices

- Assume an existing sharing arrangement
  - A vendor (the source) publishes threat alerts as information becomes known
  - Customers (subscribers) can pay to receive these daily updates
    - Multiple levels of access depending on contract costs
  - Currently, customers log into the vendor web site to view updates
Step 1: Source Organizes its Data

- **Vendor organizes data records into TAXII Data Feeds**
  - Decides on “contract level” for feeds
    - Many records will be present in all feeds, but some fields may be stripped before dissemination
  - Access to a feed contingent upon the purchasing of a contract

- **Vendor labels all data within each TAXII Data Feed with a timestamp**
  - Decides to use the time of posting as that timestamp
    - More than one data record may have the same timestamp – not a problem
    - A single record could have the same timestamp in all data feeds – not a requirement
Step 2a: Source Implements TAXII Services

- **Decides to implement a Feed Management Service**
  - Feed Information Requests
    - Lists available feeds
    - Explain what information is provided via each feed (i.e., contract levels)
    - Reference to site where one can purchase necessary contracts
  - Feed Management Requests
    - Forward management requests to back-end for comparison to purchased contracts

- **Decides to implement a Poll Service**
  - Give customers the option to pull content from a feed

- **Decides to interface with Customers’ Inbox Services**
  - Support pushing content to customer Inbox Services

- **Decides NOT to implement a Discovery Service**
  - Vendor decides to continue publishing this information using HTML
May implement an Inbox Service

- If customer wishes have updates pushed, must implement Inbox
- Inbox listens to appropriate port for connections
  - In TAXII 1.0, this would be a (truncated) HTTP server
- May avoid implementing if all content to be pulled via Poll Service

Subscribers may interface with the Vendor’s TAXII Poll Service for pull messaging

For this design, subscribers must interface with the Vendor’s TAXII Feed Management Service
Step 3: Establish Sharing Relationships

1. Customer contacts vendor Feed Management Service to get list of feeds
Step 3: Establish Sharing Relationships

1. Customer contacts vendor Feed Management Service to get list of feeds

2. Customer purchases a contract via Vendor web site
   - Also establishes authentication credentials
Step 3: Establish Sharing Relationships

1. Customer contacts vendor Feed Management Service to get list of feeds

2. Customer purchases a contract via Vendor web site
   – Also establishes authentication credentials

3. Customer contacts vendor Feed Management Service to establish subscription
   – Request verified before acceptance
Step 4: Share

- Content pushed to Customer’s Inbox Service

-- and/or --

- Customer pulls from Vendor’s Poll Service
  - Request verified before being fulfilled
Hub and Spoke Walkthrough
Background

▪ One possible way to use TAXII to implement Hub and Spoke
  – Others may make different choices

▪ Assume an existing sharing arrangement
  – Community exists with a pre-existing intra-group sharing agreement
  – Currently all threat alerts sent via e-mail to the group mailing list
    ▪ Automatically re-distributed to all group members
Step 1a: Hub Implements TAXII Services

- Decide to implement a Inbox Service
  - Used to receive all input from spokes (Hub does not poll)
- Decide to interface with Spokes’ TAXII Inbox Services for message delivery
  - Support pushing of alerts to spokes
- Decide to implement a Poll Service
  - Support spokes pulling current and/or archived alerts
  - Decide on only one TAXII data feed for all information
  - Decide timestamps = the time the alert arrives in Hub’s Inbox
- Decide NOT to implement a Discovery Service
  - Members informed of the Hub’s services via other means
- Decide NOT to implement a Feed Management Service
  - Spokes automatically enrolled when they join the sharing group
Step 1b: Spokes Implement TAXII Services

- Spokes that produce data interface with the Hub’s TAXII Inbox Service
- May implement an Inbox Service
  - If spoke wants pushed info, must implement Inbox
  - May avoid implementing if all content to be pulled via Poll Service
- Some spokes may interface with the Hub’s TAXII Poll Service
  - May avoid this use if all content to be pushed to the spoke’s Inbox Service
Step 2: Share

1. Spoke X pushes new alert to Hub’s Inbox Service
Step 2: Share

1. Spoke X pushes new alert to Hub’s Inbox Service

2. Hub re-sends alert to all spokes that requested push notification
Step 2: Share

1. Spoke X pushes new alert to Hub’s Inbox Service

2. Hub re-sends alert to all spokes that requested push notification

3. Hub archives alert so spokes can poll for the alert at a later time