Section: CS

Adaptable Cyber Threat Intelligence Sharing Tool

Cortez McCrary, cmccrary32

**Problem Statement:**

Effective and timely communication of Cyber Threat Intelligence (CTI) is critical to protecting national security, critical infrastructure, and private industry. Without the sharing of CTI, the cyber community becomes silos of critical information in the identification and mitigation of cyber-attacks. The Structured Threat Information Expression (STIX) standard and the Trusted Automated Exchange of Intelligence Information (TAXII) protocols aims to solve the interoperability problem between different government organizations, industries, and cybersecurity software tools by standardizing CTI into relational graph objects. STIX allows for CTI to be communicated and acted upon at machine speed while remaining human-readable.

In theory, the STIX standard should be the perfect vehicle for machine-to-machine threat intelligence sharing, but there is still much to be desired in practice. For example, with each new version of the STIX standard released, the industry must adapt current cybersecurity solutions before they can ingest the new schema, causing unexpected behavior. Furthermore, some industry leaders still decide to stick with their own proprietary CTI structures.

**Solution Statement:**

The current threat landscape calls for more advanced and adaptable methods of threat information sharing while still maintaining the ability to share intelligence at machine speeds. More formally, how does the cyber intelligence community improve the mechanisms used to share CTI at machine speeds that do not break existing tools when standards change and allow for enhanced or extensions of the standard between industry partners? I propose a middleware tool that can be added to the STIX and TAXII server framework that can mitigate the following:

1. Improve the consumption of enhanced STIX objects by allowing CTI producers to publish their schemas so that subscribers can ingest their enhanced STIX objects as a part of a federated system, utilizing this middleware tool.
2. Improve predictable schema conversions by ingesting CTI schemas that either describe exactly what information a specific cybersecurity tool can ingest or describe what enhancements to the STIX standard the publisher will produce in their CTI feeds.
3. Prevent CTI data loss by verifying that the structure of the data meets the standard a specific tool downstream is expecting. If the CTI does not meet the standard, the tool gives a warning of what information will be lost in a visual way (i.e. visual graph). It then conducts a conversion to meet the standard the tool is expecting.
4. Finally, because the tool can visualize any enhanced STIX standard version based on a given schema, it allows analysts to continue to analyze CTI while industry tools catch up to adapting to the STIX objects being produced.

My proposed enhanced TAXII framework is centered around a single tool meant to be middleware between TAXII servers and TAXII clients and looks to be a standardized method for federated CTI sharing. New versions of STIX standards or enhanced standards will no longer break existing tools before they are ready to ingest new objects, while still allowing an analyst to observe all portions of a CTI report. Figures 1 and 2 depict how the middleware tool would fit into both a channel and collections TAXII framework. Along with CTI objects, threat intelligence producers would be expected to publish their schemas that fall outside the STIX standard so that threat intelligence consumers can ingest them before subscribing to the producer's threat feed. If a consumer requires a specific format other than what the producer is using, the schema tool allows an analyst to define a conversion within the tool before it reaches cybersecurity products looking to ingest the CTI downstream, avoiding unexpected behavior. If an ingested CTI does not meet any specific standard and a conversion schema is not defined, then a warning is thrown with a visual representation of the problem CTI object. This allows for centralized predictable behavior when both sharing and ingesting CTI at the machine level.

A diagram of a computer system

Description automatically generated

Figure 1. Enhanced TAXII Server Channel Framework

A diagram of a computer system

Description automatically generated

Figure 2. Enhanced TAXII Server Collections Framework

**Completed Tasks (Last 2 Week):**

From 5 Feb, 2024 to 18 Feb 2024 I have accomplished:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Date | Start Time | End Time | Total Time |
| **Visualizer:** Developing STIX bundle parsing into network graph nodes and links | 3/6/2024 | 21:00 | 23:20 | 2:20 |
| **Visualizer:** Improve node information displayed | 3/7/2024 | 21:50 | 23:00 | 1:10 |
| **Visualizer:** Improve node information displayed | 3/9/2024 | 20:00 | 24:00 | 4:00 |
| **Research:**  Looked into more advanced D3.js functionality | 3/10/2024 | 21:00 | 24:00 | 3:00 |
| **Research:**  Looked into more advanced D3.js functionality | 3/11/2024 | 20:00 | 23:00 | 3:00 |
| **Visualizer:**  Added titles to each node | 3/13/2024 | 20:00 | 23:00 | 3:00 |
| **Visualizer:** Visual Formatting for json view and graph. | 3/15/2024 | 21:00 | 23:30 | 2:30 |
| **Visualizer:** Created Labels for Links. This is the last visualizer Task that will be complete. I plan to focus on backend schema integration from this point on. | 3/16/2024 | 21:00 | 23:30 | 2:30 |
| **Research:** Schema Development**.** How to use schemas to exchange STIX standard enhancements. | 3/17/2024 | 12:00 | 18:00 | 6:00 |
|  |  |  | **Total** | 27.5 hrs |

**Tasks for the Next Project Report:**

Over the next period of reporting I plan to complete the following:

* Complete Progress Video 5.
* Complete peer feedback 5.
* Complete Schema multiple schema validation backend starting with STIX 2.1 and then one other custom schema that includes a feedback extension.
* Setup test environment in a docker container utilizing one TAXII-client and one TAXII-server
* Begin completing writing documentation.

**Questions I have or Issues I’m running into:**

None at this time.

**Methodology Paragraph Summary:**

I plan to use agile methodology over the course of five, 2-week long sprints. During each sprint I will look to refine requirements, design specific solutions, develop those solutions into features, and test newly added features. Each sprint will complete key features that by the end of this course I will have a demonstratable Minimum Viable Product (MVP)/Proof of Concept (PoC). Throughout each phase of each sprint, I will work take a “design/document first, build second” mentality to optimize time required to develop each feature.

**Timeline:**

*Enter tasks for every week of this semester. One task per row and only include tasks related to your project (for example, do not include “Peer Feedback” or “Progress Report”). You’ll most likely have multiple tasks per week. When you submit your first progress report, we’re expecting to see a timeline of all tasks related to your project for the full length of the semester. It’s expected that the tasks will develop and get more detailed over time but you must start with something. Any task you list should be actionable (for example, do not have a task “continue research” or “working on xyz”). For the Status column, let us know if you’ve completed the task, if it’s still in-progress, maybe cancelled or whatever the status is.*

|  |  |  |
| --- | --- | --- |
| **Week #** | **Description of Task** | **Status** |
| W1 (Jan 8 to Jan 14) | Researched STIX Standard and TAXII Protocol | Complete |
| W1 (Jan 15 – Jan 21) | Held meeting with cybersecurity vendors to gain an understanding of how their solutions integrate STIX objects. | Complete |
| W1 | Met with cybersecurity tool developers to discuss the feasibility of my proposed solution | Complete |
| W2 (Jan 15 – Jan 21) | Write an initial Draft of a Software Design Document | Complete |
| W2 | Design a prototype User Interface | Complete |
| W2 | Design a Call Flow Diagram | Complete |
| W2 | Design a Component Diagram | Complete |
| W2 | Define User Stories into Functional and Non-Functional Requirements | Complete |
| W3 (Jan 22 – Jan 28) | Develop Initial User Interface | **Complete** |
| W3 | Develop Functionality for ingesting and viewing Schemas | **In Progress** |
| W3 | Develop functionality for selecting an output schema. | **In Progress** |
| W4 (Jan 29 – Feb 4) | Improve User Interface | **Complete** |
| W4 | Develop Functionality for ingesting and viewing STIX 2.1 objects | **In Progress** |
| W4 | Develop API for requesting STIX objects. | Not Complete |
| W5 ( Feb 5 – Feb 11) | Improve User Interface | **Complete** |
| W5 | Develop Functionality for comparing STIX objects to expected Schema. Starting with STIX 2.1. | **In Progress** |
| W5 | Visualize data loss between two conflicting Schemas | Not Complete |
| W6 (February 12 -18) | Improve UI | Not Complete |
| W6 | Develop functionality for defining a schema conversion method. | Not Complete |
| W6 | Develop functionality to complete STIX conversions | **Not going to get to Complete** |
| W7 (Feb 19 – Feb 25) | Develop functionality for verifying produced CTI fitting a given schema on the producer’s side. | **In Progress** |
| W7 | Develop the inclusion of the MITRE ATT&CK enhanced STIX schema. | **Not Going to get to Complete** |
| W8 (Feb 26 – Mar 3) | Develop the inclusion of the STIX 2.0 standard STIX schema. | Not Complete |
| W8 | Develop my own enhanced version of STIX 2.1 Schema for testing and demonstration | **In Progress** |
| W9 (Mar 4 – Mar 10) | Containerize a test and demonstration infrastructure using STIX clients, TAXII Servers, and my middleware tool | Not Complete |
| W9 | Begin documenting User and Developer Guide | Not Complete |
| W10 (Mar 11 – Mar 17) | Develop test suite for containerized environment | Not Complete |
| W10 | Begin Evaluation Testing | Not Complete |
| W11 (Mar 23 – Mar 31) | Complete test suite | Not Complete |
| W11 | Complete Evaluation Testing | Not Complete |
| W12 (Apr 1 – Apr 2 | Document Features for future Development | Not Complete |
| W12 | Document limitations of current implementation | Not Complete |
| W12 | Document deployment instructions | Not Complete |
| W13 (Apr 8 – Apr 14) | Complete user and developer guides | Not Complete |
| W14 – W15 (Apr 15 – Apr 23) | Complete Final Report | Not Complete |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Evaluation:**

My evaluation methodology will consist of setting up a test architecture in a containerized environment. I plan to use a minimal TAXII client as my validator in my test suite. The TAXII client will reach out through TAXII proxy to poll a custom built TAXII server, which will host 3 types of STIX versions: STIX 2.1, STIX 2.0, and a custom STIX 2.1 enhanced which will include an extension for providing feedback about a piece of CTI.

A diagram of a computer

Description automatically generated

I will use the TAXII client to host the test suite that will verify the results TAXII Proxy identifies as data loss based on the schema chosen for validation.

**Report Outline:**

My Final Report will include:

* Finalized Project Proposal
* Software Development Documentation and User Guide.
* TAXII Proxy source code (Hosted on Public Github)
* Docker Container with included Test Suite for repeatability. This will also include evaluation results that can be run again within the Docker Container.

**References:**

Attestation: I did use generative AI technology in producing this assignment.

Bostock, M. (2024). *Force ​*. d3. https://d3js.org/d3-force

Cowles, J. (2009). The Internet as Utopia: Reality, Virtuality, and Politics. *Oshkosh Scholar*, *IV*, 81–89.

Dulaunoy, A., Taylor, M., & Cullinane, K. (2023, September 26). Introduction to STIX. https://oasis-open.github.io/cti-documentation/stix/intro.html

Ginn, J. (2020, April 19). *Overview of STIX 2.1 for Cyber Threat Analysis*. YouTube. https://www.youtube.com/watch?v=SCWLssIKGwg&t=5s

MITRE. (n.d.). *Mitre ATT&CK®*. MITRE ATT&CK®. https://attack.mitre.org/

Walke, J. (2024). *React reference overview*. React. https://react.dev/reference/react

Xianwu. (2019, May 31). *Clickable force-directed graph(network graph)*. Observable. https://observablehq.com/@xianwu/clickable-force-directed-graph-network-graph

**Appendix**

*If there are notes, sources, figures, or draft text you want to reference please include that here.*

All code and documentation can be found at the following GitHub repository: <https://github.com/cortezcodes/Taxii-Proxy-React/tree/UI-build-V1.0.0>