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 21 Jan, 2024 to 4 Jan 2024 I have accomplished:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Date | Start Time | End Time | Total Time |
| **Progress Video** | 1/23/2024 | 22:00 | 23:00 | 1 hr |
| **Software Development Document** – Begin to write out a SDD for TAXII Proxy, which included the following sections:   * Purpose * Scope * Intended Audience * System Components * System Overview – Features * System Architecture – Architecture Overview * Component Diagram * Data Flow Diagram | 1/24/2024 | 21:00 | 23:00 | 2 hrs |
| **Call Flow Diagram -** Started developing the first call flow diagram for modeling how a user will request import from their TAXII client and TAXII proxy will validate and return the STIX bundle. | 1/25/2024 | 21:00 | 23:00 | 2 hrs |
| **Peer Feedback** | 1/27/2024 | 21:00 | 22:00 | 1 hr |
| **Component Diagram** – Started developing component diagram to identify how TAXII Proxy fits within a CTI TAXII architecture. | 1/28/2024 | 15:00 | 17:00 | 2 hrs |
| **TAXII Client build –** Utilizing a python library to build a minimal TAXII client for testing TAXII Proxy. | 1/28/2024 | 17:00 | 22:00 | 5 hrs |
| **TAXII Server Research –** Researched OSINT TAXII servers I could use to pull real threat feeds from for testing. Identified two: AlienVault OTX and Cyware. Cyware seems to be the better option | 1/28/2024 | 22:00 | 23:00 | 1 hr |
| **Wireframe Diagraming –** Began to develop wireframe for GUI in Figma tool | 2/1/2024 | 21:00 | 23:00 | 2 hrs |
| **Wireframe Diagraming –** Continued to develop wireframe for GUI in Figma tool | 2/2/2024 | 21:00 | 23:00 | 2 hrs |
| **Requirements Matrix –** Wrote a requirements matrix to keep track of traceability to user stories and features | 2/3/2024 | 17:00 | 18:30 | 1.5 hrs |
| **GUI Development –** Started developing the GUI interface for TAXII Proxy utilizing python Flask framework. Create a page for each section of the GUI including STIX View, Schema View, Homepage, Login, and Configuration pages. | 2/3/2024 | 18:30 | 19:30 | 1 hr |
| **Research Technologies –** Researched the technologies for GUI development for faster build to a minimum viable product. These included STIX2 library, networkx, bootstrap 5, and Flask-WTF | 2/3/2024 | 19:30 | 21:30 | 1 hr |
| **GUI Development –** Continued | 2/3/2024 | 21:30 | 00:00 | 2.5 hrs |
| **GUI Development –** Continued | 2/4/2024 | 8:00 | 11:00 | 3 hrs |
| **Progress Report 2** | 2/4/2024 | 11:00 | 12:00 | 1 hr |
| **Class Diagram –** Started developing a class diagram for modeling STIX objects in a more flexible manor based on schemas | 2/4/2024 | 12:00 | 15:00 | 3 hrs |
|  |  |  | **Total** | 32 hrs |

The comment I received from progress report 1 stated:

“The tool and the other deliverables are clearer now and i am more comfortable with project scope and its feasibility. However, the demonstration and testing goals are less clear. How will they relate the goals outlined at the start of the proposal "more advanced and adaptable methods"? Do you specific metrics that will demonstrate these metrics are met?” - Mustaque Ahamed

I plan to demonstrate my proposal can provide enhanced STIX sharing capabilities by developing a full environment using TAXII Proxy and multiple TAXII clients and servers. I will provide a step-by-step guide demonstrating key features that enhances the STIX/TAXII standard for CTI sharing. A few examples are:

1. Display how enhancements can be added to a schema that is shared amongst organizations, imported into TAXII Proxy, and immediately provide the ability to consume enhance objects.

2. Display how TAXII proxy can be used to identify data loss of STIX components from downstream client based on schema mismatch.

3. Display how TAXII proxy can be used to convert STIX to meet expected schema for downstream clients both manually and automatically.

The Metric will be built into test cases using the schema validators built into the tool. A good example is how OpenCTI cannot consume STIX extensions and will drop entire bundles if a STIX extension is present. One of my validator schemas can be for the OpenCTI client and I can demonstrate how STIX Proxy can convert a STIX bundle so that it is compatible with OpenCTI. I can allow the bundle to go through unmodified and show how it will be dropped. And then I can allow a second bundle to go through conversion and show how OpenCTI can ingest the new object, while still showing data loss as a expected behavior.

**Tasks for the Next Project Report:**

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

* Complete Progress Video 2.
* Complete peer feedback 2.
* Continue GUI Development
* Data Modeling
* Schema writing manual
* Finish STIX View webpage
* Finish Schema View webpage

**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 | In Progress |
| 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 | In Progress |
| W3 | Develop Functionality for ingesting and viewing Schemas | Not Complete |
| W3 | Develop functionality for selecting an output schema. | Not Complete |
| W4 (Jan 29 – Feb 4) | Improve User Interface | Not Complete |
| W4 | Develop Functionality for ingesting and viewing STIX 2.1 objects | Not Complete |
| W4 | Develop API for requesting STIX objects. | Not Complete |
| W5 ( Feb 5 – Feb 11) | Improve User Interface | Not Complete |
| W5 | Develop Functionality for comparing STIX objects to expected Schema. Starting with STIX 2.1. | Not Complete |
| 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 Complete |
| W7 (Feb 19 – Feb 25) | Develop functionality for verifying produced CTI fitting a given schema on the producer’s side. | Not Complete |
| W7 | Develop the inclusion of the MITRE ATT&CK enhanced STIX schema. | Not 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 | Not Complete |
| 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:**

[Include any evaluation plans and/or results by Progress Report 5. This may expand as you finalize the report.]

**Report Outline:**

[Include an outline of your final report by Progress Report 5. This may expand as you finalize the report.]

**References:**

[List sources you have reviewed for your project]

**Appendix**

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

[The inclusion of draft work product should begin being included by Progress Report 3.]