Review of model

**STIX (structured threat information eXpression)**

STIX was selected for use in this project within the Attack Flow builder. STIX is a standardised machine-readable language used to express cyber threat intelligence (CTI). It is expressed in JSON. Some benefits to using STIX have been identified and are as follows:

* Easy processing for different programming languages and computer architectures due to JSON format.
* STIX is used by the Attack Flow builder. Attack Flow is defined as an extension of STIX. Since the builder that is being used relies on STIX, it would be simplest to use it.
* STIX has a formal extension mechanism and software can add support for it, allowing for any necessary future adjustments.
* Encompasses details such as IP addresses, file hashes to threat actor profiles and TTPs. These components are linked together in a graph.
* Has ability to be extended, would be simpler than building a new information format from scratch.

A downside to STIX’s ability to be read by different software leads to the issue of different interpretations on how to store and represent STIX data. However, as STIX is limited to use in the Attack Flow builder, this should not present a major issue.

There are also other potential options that may have been considered. For example, there is the MISP event model. MISP was designed to focus on past incidents through events. In MISP, events are the core data object. However, MISP can still describe CTI concepts. MISP can also produce graphs, however the relationships between elements within the graph represents correlations. In comparison with STIX, the representation of relationships is less meaningful and less explicit (for example a connection on the graph could just represent reuse by different threat actors). MISP is also largely bound to the MISP software. This makes it use in the project much more limited. Since it is highly significant to demonstrate the relationships between cyber activities and influence operations, it is important that the standard that is being used can represent meaningful relationships and is expandable (to add influence operations via DISARM). MISP, like STIX, is also represented in JSON. Due to some of the downfalls of MISP, STIX can still be considered as more appropriate for this project. To summarise:

* MISP has less meaningful graph relationships (an important feature for graphs in this project due to significance of relationships between different activities)
* MISP is largely bound to MISP software, making it difficult to use in the Attack Flow builder

Another potential alternative is presented by OASIS Open is the Defense Against Disinformation Common Data Model (DAD-CDM). This model extends STIX with the aim of combatting misinformation. DAD-CDM is intended to “build a bridge between efforts to counter cyberattacks and efforts to counter information manipulation”. This aim is very similar to that of this project, which focuses on creating a model for cyber-enabled influence operations. DAD-CDM extends STIX to model hybrid threats. An example is provided where STIX currently can model a DDoS attack but cannot model coordinated inauthentic behaviour. DAD-CDM extends STIX by including new disinformation objects that can be used within models. It is currently an open project, so is being developed as it is used. As DAD-CDM is in development, there may be future changes which could affect any work produced using the model. There is no specific framework being used to implement the disinformation objects, so it may not be very relevant to the DISARM framework.

DAD-CDM may be more suited to this project due to its ability to also include disinformation objects. However, if DAD-CDM was to be used for the project, it would affect the ability to use the MITRE Attack Flow Builder (as this uses STIX). It is also possible to extend STIX to add objects from the DISARM Framework. As the Attack Flow Builder uses STIX, and this is the software that will be used for the project, it would still be more beneficial to use STIX, and extend it where necessary to incorporate DISARM with the existing ATT&CK Framework objects.

**Use of STIX:**

In the MITRE ATT&CK Framework, each domain (Enterprise, Mobile, ICS) is represented as a series of STIX 2.1 collection bundles. The framework uses STIX as it “enables organizations to share CTI with one another in a consistent and machine-readable manner, allowing security communities to better understand what computer-based attacks they are most likely to see and to anticipate and/or respond to those attacks faster and more effectively.” (MITRE, 2024). Using STIX is an appropriate choice as the goal of creating the models is to have a machine readable standard for CEIOs.

**Sources:**

Cosive 2024, *MISP vs. STIX — Cosive*, Cosive.com, viewed 27 October 2024, <https://www.cosive.com/misp-vs-stix>.

Deibler, D 2024, *‘Navigating the Nexus Between Cyber Operations and Influence Operations’*, RSA Conference, viewed 27 October 2024, <https://www.rsaconference.com/library/blog/navigating-the-nexus-between-cyber-operations-and-influence-operations>.

Koen Van Impe 2015, *Comparing Different Tools for Threat Sharing*, Security Intelligence, viewed 27 October 2024, <https://securityintelligence.com/comparing-different-tools-for-threat-sharing/>.

MITRE 2024, *attack-stix-data*, GitHub, viewed 27 October 2024, <https://github.com/mitre-attack/attack-stix-data/tree/master>.

MITRE Engenuity 2022, *Language — Attack Flow v2.2.7 documentation*, Github.io, viewed 27 October 2024, <https://center-for-threat-informed-defense.github.io/attack-flow/language/>.

Oasis Open 2016, *Introduction to STIX*, Github.io, viewed 27 October 2024, <https://oasis-open.github.io/cti-documentation/stix/intro.html>.