**Structured Threat Information eXpression (STIX) v0.3**

**Frequently Asked Questions (FAQ)**

**What is the Structured Threat Information eXpression (STIX)?**

|  |
| --- |
| The Structured Threat Information eXpression (STIX) is a standardized language, being developed in collaboration with any and all interested parties, for the specification, capture, characterization and communication of cyber threat information in a structured fashion to support more effective cyber threat management processes and application of automation. A wide variety of high-level cyber security use cases rely on such information including: computer network defense (CND), cyber threat identification, cyber threat characterization, cyber threat detection, cyber threat prevention, cyber threat response, cyber threat attribution, cyber threat indicator generation/application, cyber threat information sharing, incident response/management, etc. STIX provides a common mechanism (structure and content) for addressing structured cyber threat information across and among this full range of use cases improving consistency, efficiency, interoperability and overall situational awareness. To enable such an aggregate solution to be practical for any single use case, existing standardized languages are leveraged where appropriate and numerous flexibility mechanisms are designed into the language. In particular, almost everything is optional such that any single use case could leverage only the portions of STIX that are relevant for it (from a single field to the entire language or anything in between) without being overwhelmed by the rest.  The Structured Threat Information eXpression provides a unifying architecture tying together a diverse set of cyber threat information including: cyber observables, incidents, indicators, adversary TTP (including attack patterns, malware, exploits, kill chains, tools, infrastructure, targeting, etc.), cyber attack campaigns, cyber threat actors, exploit targets (e.g. vulnerabilities & weaknesses), and courses of action (e.g. incident response or vulnerability/weakness remedies).  STIX v0.3 is an XML Schema (XSD) schematic implementation of the STIX language for practical use as well as for concrete collaborative evolution of the language targeted for eventual abstraction to an implementation-independent form. Once this level of maturity is achieved, other potential implementations are foreseen including possibilities such as semantic web (RDF/OWL) and/or JSON-centric implementations. |

**Where did STIX come from?**

The Structured Threat Information eXpression (STIX) language evolved out of a rough structured threat information architecture diagram that was created from collaborative discussions among the experts on the US-CERT hosted idxwg (indicator exchange working group) email list. The original purpose of this architecture diagram was to clearly define the scope of what sorts of information should be included within a structured cyber threat indicator and what sorts of information should be defined in other related structures. This architecture diagram helped to clarify scope such that initial cuts at a standardized language for cyber threat indicators (Indicator Exchange eXpression (IndEX)) could be successfully drafted. As the concept and initial structure for cyber threat indicators matured, there was increasing interest from numerous parties in fleshing out the rest of the structured threat information architecture in a concrete schematic implementation to advance its collaborative maturation and provide an initial structure for real-world prototyping and proof of concept implementations in structured threat information management and sharing. STIX v0.3 is an initial public draft of this concrete schematic implementation of the structured threat information architecture.

**What other schemas does STIX import?**

Version 0.3 of STIX imports and leverages the following constituent schemas:

* Cyber Observable eXpression (CybOX) v1.0 (Draft)
* Indicator Exchange eXpression (IndEX) v0.4
* Common Attack Pattern Enumeration and Classification (CAPEC) v2.5
* Malware Attribute Enumeration and Characterization (MAEC) v2.1
* Incident Object Description and Exchange Format (IODEF) v1.0
* Data\_Marking v0.3

**What is the intent of v0.3 of STIX?**

Version 0.3 of STIX is primarily intended as a concrete strawman for ongoing collaborative development of a structured threat information expression language among a community of relevant experts. It is also targeted to provide an initial practical structure for early real-world prototyping and proof of concept implementations in structured threat information management and sharing. Only through appropriate levels of collaborative iteration among a relevant community of experts and vetted through real-world data and use cases can a practical and effective solution evolve.

***Version 0.3 of STIX is explicitly NOT asserted to be a complete or finished solution but rather an initial instantiation to serve as a seed for the evolution of an eventual complete and finished solution.***

**What is in STIX v0.3?**

Version 0.3 of STIX contains complete initial draft implementations for the following elements of the guiding structured threat information architecture diagram:

* Observables construct
* Indicator construct
* TTP construct
* Campaign construct
* ThreatActor construct
* Incident construct
* ExploitTarget construct
* CourseOfAction construct
* Namespace segregation for the above constructs and for imported schemas
* Appropriate relationships among the above constructs
* Common structure for Confidence
* Structure for Sightings
* Generalized structure for handling markings (split out into its own schema and namespace)
  + Exemplar specific instantiation for TLP handling markings
* Generalized structure for Kill Chain definition
* Common structure for Producer or Information Source
* Common structure for Intent
* TTP Tools structure
* TTP Infrastructure structure
* Extensive refinement of the previous version (v0.3) of the IndEX schema to align with STIX. Some portions were left as-is for now to minimize effect on those already working with IndEX.
  + Addition of IndicatedTTP assertion structure
  + Addition of Kill Chain Phases adornment
  + Addition of Test Mechanism structure
  + Addition of an Impact field
  + Addition of Suggested COA (Course of Action) structure
  + Addition of a Producer structure
  + Refinement of Handling, Confidence, Sightings and relationship structures
* Import of as-yet-unpublished namespace segregated CAPEC schema

Almost all attributes, elements and types should be fully annotated.

**What is not yet in STIX v0.3?**

Version 0.3 of STIX explicitly does not contain complete initial draft implementations for the following elements of the guiding structured threat information architecture diagram. Development of initial draft implementations for these elements is planned for the near future through collaborative discussion and refinement among the community of experts.

* TTP:Exploit characterization structure
* STIX baseline handling markings structure (specific implementation leveraging the currently defined generalized structure)
* Indicator temporal dimension decisions or structures: status, versioning, history
* ThreatActor:Identity structure
* TTP:Targetting structure
* STIX baseline Kill Chain definition (specific implementation leveraging the currently defined generalized structure)
* IndEX Impact structure

**What sort of expert contributions are desired?**

STIX welcomes expert feedback and contributions to ensure that the language is expressive and flexible enough to support the appropriate variety of use cases within the cyber threat operations, management and information sharing domains. Expert contributions should not seek to limit the capabilities of the language just because it contains expressive capabilities that may not be germane to the expert’s use cases. It is likely that other use cases require those expressive capabilities. Testing the expressiveness of the language through simulated scenarios and example data is ideal and contributions which share the utilized scenario descriptions and example data are most welcome.