CWE/CAPEC Terminology and usage in the REST API

Rich Piazza, MITRE

Steve Christey Coley, MITRE

May 12, 2022

What Are We Talking About?

- **What are the <knowledge concepts> we're trying to share information about?**
  - Weaknesses
  - Attack Patterns
  - IDs
- **What is the <representation> of the concepts we're trying to share information about?**
  - XML
  - JSON
  - "database"

Concepts in CWE

- **Weakness**
  - Each weakness has 15+ additional piece of knowledge, like Name, Description, 0 or more Demonstrative Examples, Potential Mitigations, 0 or more References, etc.
  - Many of these concepts themselves have lower-level knowledge
    - Mitigation: SDLC phase, strategy, description, effectiveness
  - Some are simple freeform English
  - Some are from fixed lists of values (e.g. C, C++, Perl, PHP, Go, …)
  - Some are fairly complex
    - Demonstrative examples: introductory text; bad code example, explanation of why the code is bad; good code example
  - The CWE web site has a "glossary" of terms (freeform text) for key concepts

Concepts common to both CWE and CAPEC

- **Views**
  - Express relationships between weaknesses and other concepts together
  - Different views pose different perspectives
  - Usually only a subset of weaknesses / attack patterns
- **Categories**
  - Groupings of weaknesses that share a common characteristic
- **Reference**
  - author, title, URL, release date, internal reference ID
- **Some common concepts have different XML representations**
  - Mitigations

Concepts in CAPEC

- **Attack Pattern**
  - Each Attack Pattern has <X> additional pieces of knowledge
  - Many have lower-level knowledge
    - Taxonomy Mapping: taxonomy_name, entry_id, entry_name, mapping_fit
  - Some are simple freeform English
  - Some are from fixed lists of values (Confidentiality, Integrity, Availability,...)
  - Some are fairly complex
    - Execution flows – different phases (explore, experiment, exploit), 0+ steps, description, 0+ techniques

Representations involved in CWE/CAPEC REST API Dev

- **CWE/CAPEC primary representation: XML**
  - Consists of “Elements” and “Attributes”
  - <MyElement attribute1=“hello”>
    - <ChildElement>world</ChildELement>
  - </MyElement>
- **JSON**
  - STIX version of CAPEC

Potentially confusing/confounding terms

- **Some concepts can be represented in different ways**
  - In CWE XML, the weakness ID and Name are attributes, but could be implemented as elements
- **"Field"**
  - Sometimes used in CWE/CAPEC to cover how complete an entry is, i.e., how much information it has
  - Roughly means "element or attribute" in XML
- **"Property" – used in STIX, but not a JSON concept**
  - **In JSON: name/value pairs (also called key/value pairs)**
- **Property/Field vs. Type**
  - A CAPEC entry has a property/field called ExecutionFlow, which is populated with an ExecutionFlowType object

STIX Specification / Document Conventions

1.1 Document Conventions

The following color, font and font style conventions are used in this document:

- The `Consolas` font is used for all type names, property names and literals.
  - type names are in red with a light red background – `threat-actor`
  - property names are in bold style – **`created_at`**
  - literals (values) are in blue with a blue background – `malicious-activity`
  - All relationship types are string literals; therefore, they will also appear in blue with a blue background – `related-to`
- In an object's property table, if a common property is being redefined in some way, then the background is dark grey.
- All examples in this document are expressed in JSON. They are in `Consolas` 9-point font, with straight quotes, black text and a `light grey background`, and using 2-space indentation. JSON examples in this document are representations of JSON objects [RFC8259]. They should not be interpreted as string literals. The ordering of object keys is insignificant. Whitespace before or after JSON structural characters in the examples are insignificant [RFC8259].
- Parts of the example may be omitted for conciseness and clarity. These omitted parts are denoted with the ellipses (...).
- The term "hyphen" is used throughout this document to refer to the ASCII hyphen or minus character, which in Unicode is "hyphen-minus", U+002D.

Questions for the WG

- **Where is confusion likely to occur?**
- **How strict do we need to be in conversation? In documentation?**
- **How should we distinguish between property/field and type?**