CWE REST API Working Group Meeting – April 28, 2022

1) Call to order
2) Agenda approval
3) Regular order
   - Steve Christey Coley to update on email list reflector
   - GitHub: https://github.com/CWE-CAPEC/REST-API-wg
   - "First principals" e.g., high performance or interoperability or API backward compatibility
     - Performance (low data content)
       - Data field filtering
       - SQL "select" statement, for example
       - Entire CWE = 11M, but simple request is at least 10x less
       - Date range as a filter
       - Latency requirement: single item reply should be sub-second response
       - Use cases
         - Drop-down list of all IDs → click one → get further detail
         - Sensor ap: what CWEs apply to this sensor (e.g., safety sensor) for a lightweight mitigation set ("select" statement might work for this)
         - CWE related to a CVE when code analysis tool flagged issues
       - Sorting would NOT be a feature
         - Top 25 would ok
         - Easiest to observe CWE could be retrieved (ordered)
     - Interoperability
       - STIX may add 256 more bytes per request, as STIX includes a wrapper around the underlying CWE information
       - Weaknesses easily expressed with STIX
       - STIX has "extensibility" feature
       - STIX is the schema
       - TAXII is an access method, but not the only way to the STIX data
     - API backward compatibility
       - CWEs released quarterly
       - 50% of the time, schema changes
       - New fields, info, elements…
       - Idea: don't tie the API to the schema
       - Don't delete columns… simply add new fields
       - Provide enumerations for fields – this would keep some means to digest new items
       - Schema and content versioning already exists
       - Would the API ALSO have a version?
       - Schema might have several data release versions, but only ONE schema survives at a time
       - Could some "rules of engagement" be leveraged to handle SOME schema flexibility?
         - Example: new enumeration added to field
         - Perhaps some mechanism to "delay" rollout of "latest" schema by API request
         - Balance against MITRE desire to limit available versions
   - DB/tool chain – deferred to later meeting

- STIX 2.1
- TAXII
- Swagger (language: API Documentation & Design Tools for Teams | Swagger) can be used as a design front-end to the API, code, and docs, and supports port to YAML
- XPATH link to XML
- GraphQL and GraphQL Playground (Charles, Ray)
- JSON (schema example: https://github.com/oasis-open/cti-stix2-json-schemas)

4) Action item review
   - Luke to send question about STIX use and a link to STIX information (https://oasis-open.github.io/cti-documentation/resources#stix-21-specification)
   - Steve CC. and Rich to deliver database nomenclature (vocabulary, glossary)
   - Metrics on API use, yet protect security of users' information
   - Means to restrict use (to avoid DoS issues, etc.) to registered users; keys?
5) Adjourn