STIX Incident Time Fidelity

**Purpose:** The addition of fields to the core incident extension that allow for optional recording of the precision of low fidelity time records to enable both humans and machines to understand the degree of estimation and lack of detail being provided to avoid making erroneous analytical assumptions

**Challenges:**

1. Requirements for variable precision in common units
2. Must be easily usable by humans and easy to handle in code with minimal error
3. Time Zones
4. Daylight Savings
5. Leap Years
6. Leap Seconds

**Options:**

1. Do nothing and keep the status quo
   1. Pro: No work required
   2. Con: Analysts will continue to use fuzzy time and will not be able to communicate this effectively. As such confusion and suspicious will govern timestamps for Incidents.
   3. Con: Analysts may use description fields to explain fuzziness, but computer systems will not be able to understand these and analytics will necessarily produce poor results.
2. Have two timestamps for each to show the true start / end
   1. Pro: Avoids problems caused by leap years and seconds.
   2. Con: Requires two fields to always be filled instead of one.
   3. Con: Not backwards compatible
   4. Con: Additional checks are required as start\_start\_time must now be greater than end\_start\_time.
   5. Con: Naming conventions associated with this are painful.
3. Have an enumerated precision field – Proposed Solution
   1. Pro: Makes common mappings obvious for: hour, day, month, and year
   2. Con: Does not allow for multiples of any given time unit. For example “2 days”.
   3. Con: Hard to compute as months and years have variable length so context awareness is required to handle
   4. Con: Requires more resources on ingest and when processing native STIX to convert to a valid end timestamp or numeric offset.
4. Have one timestamp and one numeric precision field
   1. Pro: Allows backwards compatibility by defaulting to 0.
   2. Pro: Is easy to compute range values since you can add precision to the timestamp to generate a range.
   3. Pro: Allows units to be displayed easily for human consumption
   4. Con: Year based precision can be a challenge due to leap years.
   5. Con: Month based precision can be off due to month length not being uniform.
   6. Con: Day based precision can very rarely have errors off due to leap seconds.

**Enumeration Proposal**

1. Introducing the following fields with the core STIX Incident extension to that map to peer level timestamp fields:
   1. start\_time\_fidelity (start\_time)
   2. end\_time\_fidelity (end\_time)
   3. timestamp\_fidelity (timestamp).
2. Record one of the following enumeration values to these fields with the following expected values in other fields
   1. Hour – Seconds should be 0 and minutes should align with the appropriate timezone
   2. Day – Seconds should be 0 and minutes should align with the appropriate timezone
   3. Month – Day should be 1st of the month, hours and minutes should align to the appropriate timezone, second should be 0 *[nice to have but not required]*
   4. Year – Day and month should be 1, hours and minutes should align to the appropriate timezone, second should be 0 *[nice to have but not required]*

**Handling of Known Challenges:**

1. A strict enumeration does not allow for units such as 2 days, but it can easily list all common levels of standard incident time reporting fidelity.
2. Humans and systems can easily handle an enumeration of 3 to 7 items. This option also makes representing this very easy within GUIs to support standard workflows.
3. Time zones are supported by using day fidelity or lower and simply setting the UTC time. For higher fidelity time entries time zones are not necessary given the requirement to put times in UTC.
4. Daylight savings are handled like other time zones so this is not an issue.
5. Leap years only need to be accounted for when the year enumeration is used. This is likely best handled by adding a year on the backend / frontend if calculations are required to generate a time range in which case native date systems should take care of the leap year. For human facing interfaces the enumeration lets them largely ignore the problem.
6. Leap seconds should not be an issue for any level of time fidelity except except minute which is not necessary for this proposal.

**Discussion Items:**

1. Should “second” and “millisecond” be included? Both second and millisecond precision can be accurately described based on the current rules for timestamps so keeping the field null can convey the same information with less code.
2. Is “minute” desired? This would allow for a complete mapping, but it is very rare for anyone to care if a timestamp tied to an incident is accurate to the minute instead of the second?

**Example:**

{

"type": "incident",

"id": "incident--11c4461e-25d8-47f1-bccf-a965ddca8e8f",

"created": "2020-10-19T01:01:01.000Z",

"modified":"2020-10-19T02:12:23.000Z",

"created\_by\_ref": "identity--2242662b-d581-4864-8696-fff719dc0500",

"spec\_version": "2.1",

"name": "Stolen Laptop",

"description": "Someone stole a laptop",

"extensions": {

"extension-definition--ef765651-680c-498d-9894-99799f2fa126": {

"extension\_type": "property-extension",

"investigation\_status": "closed",

"determination": "successful-attempt",

"detection\_methods": ["user-report"],

"incident\_types": ["loss-of-control"],

"defender\_activities": [

{

"activity\_type": "discovery",

"description": "User alerted security",

"timestamp": "2020-10-15T06:00:00.000Z",

"timestamp\_fidelity": "day"

},

{

"activity\_type": "updated-inventory",

"timestamp": "2020-10-19T06:00:00.000Z",

"timestamp\_fidelity": "hour"

},

{

"activity\_type": "flagged-spill-incident",

"timestamp": "2020-10-19T02:12:23.000Z"

}

]

}

}

}