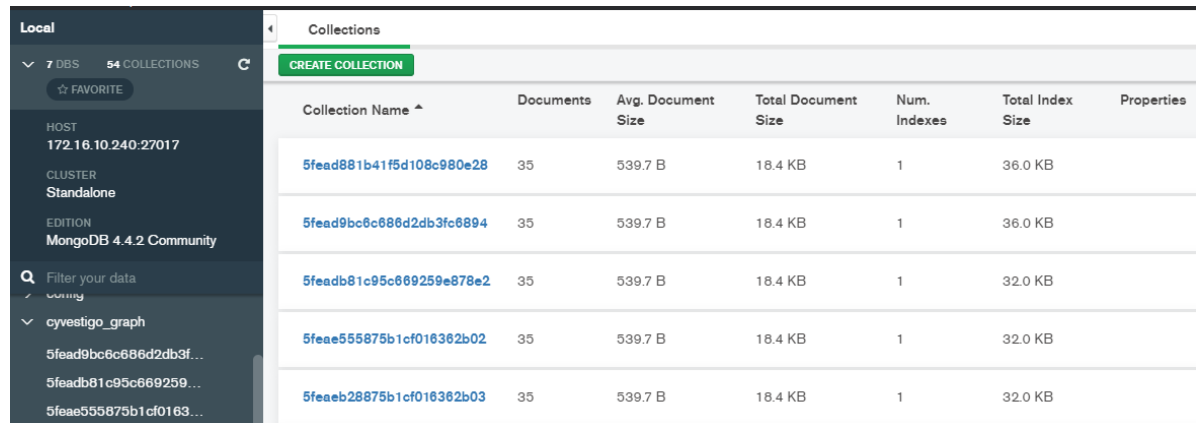


JSON Schema Explained

For graph drawing / generation and the relation to event log in SIEM.

Query ID => MongoDB Collection

For every new graph query, a new collection will be created in MongoDB, in the database `cyvestigo_graph` database.



The screenshot shows the MongoDB Compass interface. On the left, the 'Local' sidebar displays the database 'cyvestigo_graph' with several collections listed. The main panel shows a table of collections with the following data:

| Collection Name | Documents | Avg. Document Size | Total Document Size | Num. Indexes | Total Index Size | Properties |
|--------------------------|-----------|--------------------|---------------------|--------------|------------------|------------|
| 5fead881b41f5d108c980e28 | 35 | 539.7 B | 18.4 KB | 1 | 36.0 KB | |
| 5fead9bc6c686d2db3fc6894 | 35 | 539.7 B | 18.4 KB | 1 | 36.0 KB | |
| 5feadb81c95c669259e878e2 | 35 | 539.7 B | 18.4 KB | 1 | 32.0 KB | |
| 5feae555875b1cf016362b02 | 35 | 539.7 B | 18.4 KB | 1 | 32.0 KB | |
| 5feae555875b1cf016362b03 | 35 | 539.7 B | 18.4 KB | 1 | 32.0 KB | |

JSON data for graph drawing

- JSON data that is used for graph drawing is contained within the `{queryid}` collection with `type: "graphData"`.
- Additional details for each nodes and links located in one of the `property` identifiable by `Id: "{id}"` field.

```
_id: ObjectId("6004ff527d6be0bd340fc2d8")
type: "graphData"
> data: Object
```

```
_id: ObjectId("6004ff527d6be0bd340fc2d9")
type: "property"
Id: "1"
> Properties: Object
```

```
_id: ObjectId("6004ff527d6be0bd340fc2da")
type: "property"
Id: "2"
> Properties: Object
```

Graph Data

The nested `data` object of `type: "graphData"` contains the entire JSON object necessary for graph drawing.

```

>
_id: ObjectId("6004ff527d6be0bd340fc2d8")
type: "graphData"
data: Object
  type: "LinkChart"
  items: Array
    > 0: Object
    > 1: Object
    > 2: Object
    > 3: Object
    > 4: Object
    > 5: Object
    > 6: Object
    > 7: Object
    > 8: Object
    > 9: Object
    > 10: Object
    > 11: Object
    > 12: Object
      id: "user_ports_13.13.13.10"
      parentId: ""
      type: "node"
      t: "1024-49151"
      d: Object
        labels: Array
          0: "Port"
          tag: ""
    > 13: Object
      id: "1-2-a1-Create Process"
      type: "link"
      id1: "1"
      id2: "2"
      w: 2
      c: "grey"
      dt: Array
        t: "Create Process"
        g: null
        a1: false
        a2: true
      d: Object
        ids: Array
        score_ttp: Array
    > 14: Object
    > 15: Object
    > 16: Object
    > 17: Object
    > 18: Object

```

- `type: "Linkchart"` is required for the graph library to understand and parse the array if `items` for nodes and links.
 - **note: next major version of CyvestiGO is projected to shift `items: Array` up 2 levels and each node and link will be on their own Mongo Document to avoid hitting maximum Document size constraint**
- `items: Array` contains all nodes and links
- each item in this array contains
 1. `id` - a unique ID for the node or link
 2. `type` - to indicate if it is a node or link
 3. `t` - to display text on the node or link
 4. `d` - to show additional data / details such as labels or tag that must be attached to the graph data to visually present it on the UI and cannot be lazy loaded into from `type: "property"`
 5. `parentId` - if present will create a grouping of multiple nodes of the same `parentId`

6. `id1` and `id2` - these two fields will always appear together and indicate that this object is a link. They identify both node that are linked
7. `a1` and `a2` - indicate the directions of the link
8. `ids` nested in `d` indicate an array of properties that this link contains
9. `score_ttp` nested in `d` indicate the score CyvestiGO enrichment engine had given for this particular link
10. `g` indicate the glyph on the node. In our case, uppercase "I" indicate IOC enriched nodes and uppercase "T" indicate TTP enriched links

Properties

The nested JSON object contained in `Properties` extracted from the event logs retrieved from the SIEM (Splunk for this instance).

```
_id: ObjectId("6004ff527d6be0bd340fc2d9")
type: "property"
Id: "1"
Properties: Object
  eventID: 4688
  computerName: "Q053.stella.local"
  recordNumber: 34389622
  time: 2020-11-02T10:40:05.000+00:00
  epochTime: 1604284805
  accountDomain: "STELLA"
  newProcessName: "C:\\Users\\SUMITK~1.STE\\AppData\\Local\\Temp\\ipnet32.exe"
  creatorProcessName: "C:\\Users\\SUMITK~1.STE\\AppData\\Local\\Temp\\ipnet32.exe"
  accountName: "sumit.k"
  creatorProcessID: "0xc9c"
  newProcessID: "0x12a8"
  processCommandLine: "C:\\Users\\SUMITK~1.STE\\AppData\\Local\\Temp\\ipnet32.exe"
  originatingComputer: "169.254.215.99"
  path: "C:\\Users\\SUMITK~1.STE\\AppData\\Local\\Temp\\ipnet32.exe"
  Title: "ipnet32.exe"
  tag: ""
```

- only data that after research deem necessary for the analyst are extracted
- for Windows Security Event logs, `recordNumber` will uniquely identify the exact event log in Splunk
- Each `eventID` will have a different set of key, value pairs
- for Windows Security Event logs we are supporting event code:
 - 4688
 - 4663
 - 4657
 - 4689
 - 5156
 - 5154