**Draft Recorded Futures / Sumo Logic Integration**

1. **Strategy**
   * We use a data source as record keeper and threat intelligence source.
   * The data source is unique to each file record (5 in the script)
   * We have a query looking for the last timestamp on specific threat feeds
     + It calculates the delta between last seen and current date
     + If the delta is more than a ¥threshold it fires a script or lambda function
     + This script is the download script which can:
       1. Get and Upload directly to the Web Collector
       2. Get, Store, and then upload to the Web Collector
2. **Download Script Logic**
   * Use an existing API key to connect to Recorded Futures
   * Stream the data from Recorded Futures to into a Web Collector
   * Optionally persist the files locally for a replay cache, as well as other purposes
   * Script streams the data in CSV format (or others) into Sumo-Logic Web Collector
   * Using the Web Collectors, we index threat intelligence as well
3. **Working Parts Needed:**
   * Script to collect data plus credentials to access the
   * Client Hosted Collector Web hosted HTTP collector
   * Web Sources and Partition/Category for the Recorded Futures Data. Examples:
     + recordedfutures/ip
     + recordedfutures/hash
     + recordedfutures/url
     + recordedfutures/vulnerability
     + recordedfutures/domain
4. **Benefits:**
   * We can keep change records of threat intelligence for days
   * We can keep storage low with retention periods, tuned to each mapping
   * We can use analytics for complex queries against all or some of the maps

**Example of Web Collector URL (URL is only an example):**

<https://collectors.jp.sumologic.com/receiver/v1/http/ZaVnC4dhaV0MnCOwJ5fk69I5ucUjRTnUfAqKCW7TJpvHHk37oR8b5BAK76tIWb7OKmXgbQ9CZxziLSfhI9RkH5oIDZMU859ekRe1UlGDHzpwodmsoZZZ9920309==>

**Client Setup**

1. **Recorded Future Subscription and Recorded Futures API key**
2. **Client Defines a Sumo-Logic HTTP Hosted Collector with source category setup**
3. **Client Defines an Installed Collector if possible (local or cloud resource)**
4. **Client Defines a Sumo-Logic Partition for Recorded Future Maps (recommended)**
5. **Client Sets up the Script to Collect the Recorded Future data**
6. **Client Sets up the Query to trigger the script to collect Recorded Future data**

**NOTE:** while this can be a lambda it is recommended to use an installed collector running a local script to avoid costs, as the download times can be in the minutes and the memory footprint can be large in size.