

INSTALL  
GUIDE

 Recorded Future<sup>®</sup>

**RECORDED FUTURE FOR  
MICROSOFT AZURE**

Security operations and incident response analysts are trying to stay afloat with identifying, triaging and responding to threats targeting the organization before damage to the business occurs. Or at the very least, minimizing that damage as much as possible. With too many alerts, too little time and not enough information, it's difficult to determine which alert represents a critical incident and which may just be a redundancy or a false positive.

Recorded Future combines sophisticated machine and human analysis to fuse open source, dark web, and technical sources with original research. This approach automatically creates outcomes that can be consumed by analysts easily and integrated with security systems to support three primary uses cases for security operations and incident response:

**Threat Prevention:** Block Threats with high confidence for Less Business Disruption

**Threat Detection:** Correlate Recorded Future intelligence with your internal data to detect previously undetected threats

**Sentinel Alert/Incident Triage/Enrichment:** Confidently Prioritize and Resolve Alerts

*Obs.: The integration capabilities are not limited to the areas mentioned above but cover other areas according to the coverage provided by Recorded Future intelligence and data*

Recorded Future Alerts (configured via Recorded Future Portal/UI)

## SECOPS AND RESPONSE

A growing attack surface and abundance of alerts slows detection and response.

<h3>Use Cases</h3>  <ul style="list-style-type: none"> <li> <b>Alert Triage</b></li> <li> <b>Threat Detection</b></li> <li> <b>Threat Prevention</b></li> </ul>	<h3>FEATURES</h3> <ul style="list-style-type: none"> <li>• Broadest source coverage</li> <li>• Real-time risk scores and context</li> <li>• Block-grade indicators</li> <li>• 10+ SIEM and SOAR integrations</li> </ul>	<h3>HOW RECORDED FUTURE HELPS</h3> <ul style="list-style-type: none"> <li>• 50% more alerts reviewed</li> <li>• Fewer false positives</li> <li>• Detection of previously undetected threats</li> <li>• Threat blocking without business disruption</li> </ul>
--	---	---

# RECORDED FUTURE FOR MICROSOFT AZURE

## Table of Contents

<b>Threat Prevention</b> .....	<b>4</b>
Importing Recorded Future Security Control Feed for blocking in Microsoft Defender ATP.....	4
Command & Control IPs.....	4
Weaponized Domains.....	4
Weaponized URLs.....	4
Building the Logic App workflow.....	4
<b>Threat Detection</b> .....	<b>4</b>
Importing Recorded Future IOCs & context for alerting in Microsoft Azure Sentinel.....	4
Detection based on Recorded Future IP Risklists.....	5
Detection based on Recorded Future Domain Risklists.....	5
Detection based on Recorded Future URL Risklists.....	5
Detection based on Recorded Future HASH Risklists.....	6
Building the Logic App workflow.....	6
Using Recorded Future tilndicators for alerting/detection.....	6
Building/Modifying a rule focused on Recorded Future TI.....	7
<b>Alert Triage</b> .....	<b>8</b>
Enriching indicators (one-by-one) with Recorded Future Context.....	8
Enriching indicators (in bulk) with Recorded Future Context.....	11
<b>Recorded Future Custom Connector</b> .....	<b>18</b>
<b>Creating the tilndicators Batching Logic App</b> .....	<b>19</b>
<b>Creating the Recorded Future Intelligence importing Logic App</b> .....	<b>23</b>
<b>APPENDIX</b> .....	<b>32</b>
A1 - JSON Parsing Schemas.....	32
A1.1 - Recorded Future RiskLists (Detection) - Full List.....	32
A1.2 - Recorded Future RiskLists (Detection) - Only one indicator.....	33
A1.3 - Recorded Future Security Control Feed - Command & Control IPs.....	34
A1.3 - Recorded Future Security Control Feed - Weaponized Domains.....	36
A1.4 - Recorded Future Security Control Feed - Weaponized URLs.....	37
A1.5 - Recorded Future Risk related context and Intelligence Card Link (Enrichment Action).....	38
A1.6 - Recorded Future Risk related context (SOAR API - Bulk Enrichment).....	40
A1.7 - Recorded Future Alert Notifications Search.....	47
A2 - JSON Structure for tilndicators creation action.....	49
<b>REFERENCES</b> .....	<b>50</b>

## Threat Prevention

### Importing Recorded Future Security Control Feed for Blocking in Microsoft Defender ATP

Recorded Future provides the following Security Control Feeds for prevention purposes in Microsoft Defender ATP:

- [Command & Control IPs](#)
- [Weaponized Domains](#)
- [Weaponized URLs](#)

### Building the Logic App Workflow

In order to implement any of the above mentioned use cases please follow the next steps:

1. [Create the tilIndicators Batching Logic App](#)
2. [Create the Recorded Future Intelligence import Logic App](#)

You can create an indicator for:

- Files
- IP addresses, URLs/domains

#### 📌 Note

There is a limit of 15,000 indicators per tenant.

## Threat Detection

### Importing Recorded Future IOCs & Context for Alerting in Microsoft Azure Sentinel

Recorded Future provides multiple types of datasets, called Risklists, for detection purposes.

<https://support.recordedfuture.com/hc/en-us/articles/115000897248-Recorded-Future-Risk-Lists>

**Detection based on [Recorded Future IP Risklists](#)**

- Default
- IPs with Score 90+ (very malicious)
- Current C&C Server
- Actively Communicating C&C Server
- Recent Botnet Traffic
- Phishing Host
- Recently Reported by Insikt Group

**Detection based on [Recorded Future Domain Risklists](#)**

- Default
- Domains with Score 90+ (very malicious)
- C&C DNS Name
- Recently Reported by Insikt Group
- Recent COVID-19-Related Domain Lure: Malicious
- Recent Phishing Lure: Malicious
- Ransomware Payment DNS Name
- Recently Active Weaponized Domain

**Detection based on [Recorded Future URL Risklists](#)**

- URLs with Score 90+ (very malicious)
- C&C URL
- Ransomware Distribution URL
- Recently Reported by Insikt Group
- Positive Malware Verdict
- Compromised URL

## Detection based on [Recorded Future HASH Risklists](#)

- Recently Active Targeting Vulnerabilities in the Wild
- Observed in Underground Virus Testing Sites
- Malware SSL Certificate Fingerprint

## Building the Logic App Workflow

In order to implement any of the above mentioned use cases please follow the next steps:

1. [Create the tilndicators Batching Logic App](#)
2. [Create the Recorded Future Intelligence import Logic App](#)

## Using Recorded Future tilndicators for Alerting/Detection

### Leveraging Existing Microsoft Azure Sentinel Rule Templates

Microsoft provides in Azure Sentinel an extensive list of “Rule Templates” that make use of available and active tilndicators. They can be found by searching for “TI map” in the Azure Sentinel Analytics section:

NAME	RULE TYPE	DATA SOURCES	TACTICS
(Preview) TI map IP entry to AzureActivity	Scheduled	Threat Intelligence Platfo... +1	Impact
(Preview) TI map URL entry to PaloAlto data	Scheduled	Palo Alto Networks +1	Impact
(Preview) TI map Domain entry to PaloAlto	Scheduled	Palo Alto Networks +1	Impact
(Preview) TI map Email entry to SigninLogs	Scheduled	Threat Intelligence Platfo... +1	Impact
(Preview) TI map URL entry to SecurityAlert data	Scheduled	Microsoft Cloud App Sec... +2	Impact
(Preview) TI map File Hash to CommonSecurityLog E...	Scheduled	Palo Alto Networks +1	Impact
(Preview) TI map Email entry to SecurityAlert	Scheduled	Azure Security Center +1	Impact
(Preview) TI map Email entry to CommonSecurityLog	Scheduled	Palo Alto Networks +1	Impact
(Preview) TI map File Hash to Security Event	Scheduled	Security Events +1	Impact
(Preview) TI map Domain entry to DnsEvent	Scheduled	DNS (Preview) +1	Impact
(Preview) TI map IP entry to AWSCloudTrail	Scheduled	Threat Intelligence Platfo... +1	Impact
(Preview) TI map URL entry to AuditLogs	Scheduled	Azure Active Directory +1	Impact

By selecting any of these templates (depending on what use case you want to enable for detection based on the Threat Intelligence data) clicking “Create Rule” you can start the process of creating and configuring rules that start from any of these.

## Building/Modifying a rule focused on Recorded Future TI

In this section we will present an example of Rules Logic focused only on indicators associated with a particular Recorded Future dataset, imported via the mechanism mentioned in the previous sections. We will use the name of the risklist, "Recorded Future - Actively Communicating C&C Server IPs" according to the string that was provided in the "Description" field for the indicators, at the moment of their creation:

```
let dt_lookBack = 1d;
let ioc_lookBack = 1d;
ThreatIntelligenceIndicator
| where TimeGenerated >= ago(ioc_lookBack) and ExpirationDateTime > now()
| where Active == true
| where Action == 'alert'
//Picking up only IOCs that are provided by Recorded Future and are part of the dataset "Actively Communicating C&C Server IPs"
| where Description == 'Recorded Future - Actively Communicating C&C Server IPs'
| extend TI_IpEntity = NetworkIP
| join (
    DnsEvents | where TimeGenerated >= ago(dt_lookBack)
    | where SubType =~ "LookupQuery" and isnotempty(IPAddresses)
    | extend SingleIP = split(IPAddresses, ",")
    | mvexpand SingleIP
    | extend SingleIP = tostring(SingleIP)
    // renaming time column so it is clear the log this came from
    | extend DNS_TimeGenerated = TimeGenerated
)
on $left.TI_IpEntity == $right.SingleIP
| summarize LatestIndicatorTime = arg_max(TimeGenerated, *) by IndicatorId
| project LatestIndicatorTime, Description, ActivityGroupNames, IndicatorId, ThreatType, Url, ExpirationDateTime,
ConfidenceScore, DNS_TimeGenerated,
TI_IpEntity, Computer, EventId, SubType, ClientIP, Name, IPAddresses, NetworkIP, NetworkDestinationIP, NetworkSourceIP,
EmailSourceIPAddress
| extend timestamp = DNS_TimeGenerated, IPCustomEntity = ClientIP, HostCustomEntity = Computer, URLCustomEntity = Url
```

## Alert Triage

### Enriching Indicators (one-by-one) with Recorded Future Context

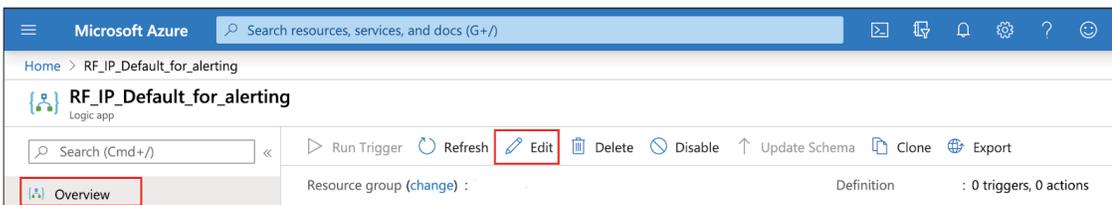
In this section we will present, via a Logic App, how to extract indicators from a Microsoft Sentinel Alert and enrich (with Recorded Future data via the standard Connect API) the Incident, via adding a comment, with Recorded Future Risk Score, Risk Rules, Evidence Details and Link to the Recorded Future Intelligence Card.

1. Create a new Logic App by clicking on “Add” in the Logic Apps section



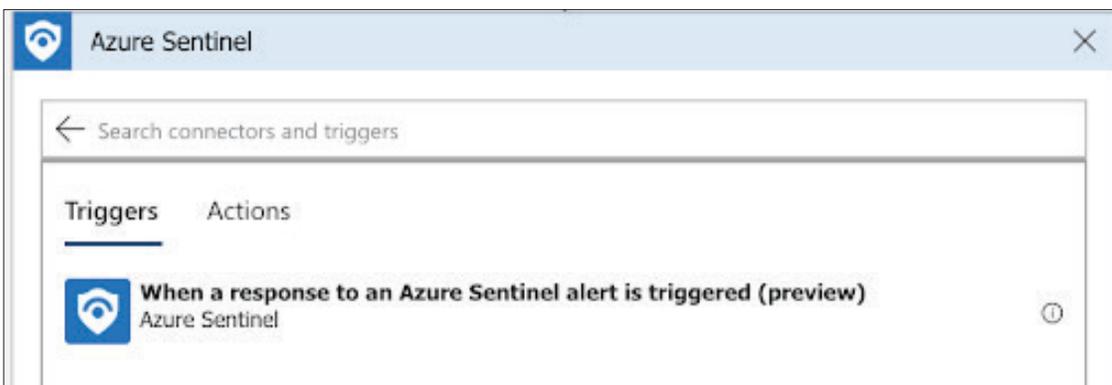
2. Provide a name for your Logic App and additional required dependencies and finalise the creation

3. Once the Logic App has been created, access it and in the “Overview” section click on the “Edit” button to access the Logic App Designer



4. Start with a “Blank Logic App” template

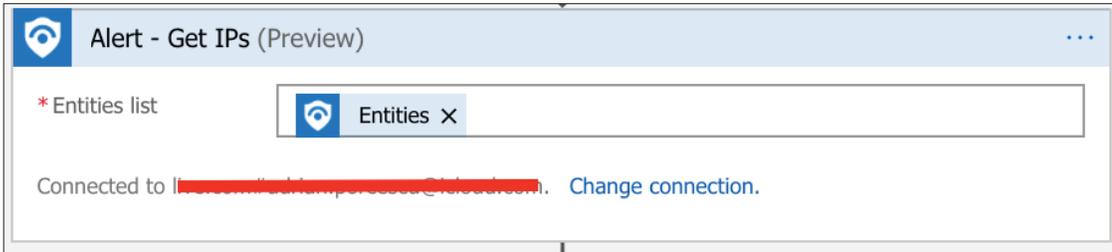
5. In order to run this Logic App when an response to an Azure Sentinel alert is triggered we need to add as a trigger “When a response to an Azure Sentinel alert is triggered” action block



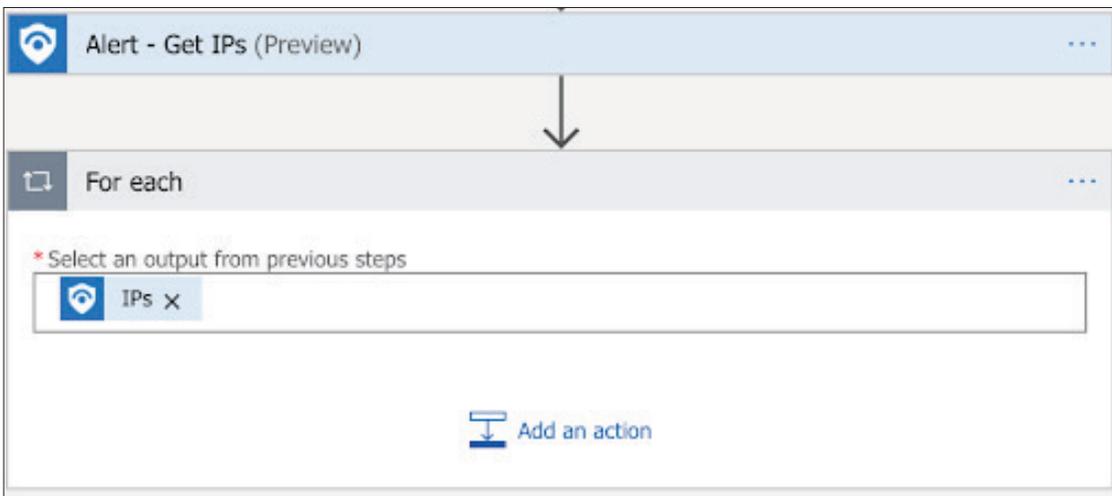
6. When configuring the trigger, chose a connection to use that has permissions to read from Azure Sentinel

7. In order to extract the indicators (we will focus on the indicators of IP type) that are reported in the incident we will add the “Alert - Get IPs” action.

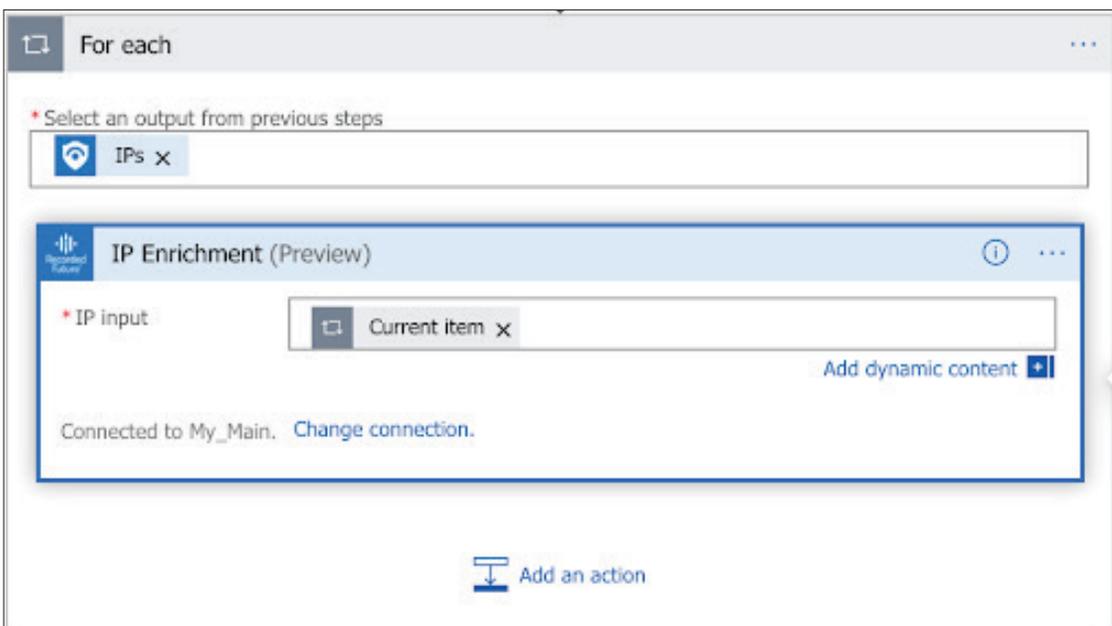
8. Select via Dynamic Content the “Entities” from the previous block as the input for the “Entities List” field:



9. Add a “For Each” block to cycle through all the IPs that were extracted from the incident. The block will have as input the output from the previous “Alert - Get IPs” block.



10. Inside the previous “For each” block, add a Recorded Future IP Enrichment action and configure it’s input as each one of the IP from the iterations of the “For Each” block:



11. To push the newly enriched content as a comment, back into the incident, add a “Add comment to incident” action. Configure the fields according to your environment and the values you want to make available in the comment:

**Add comment to incident (V2) (Preview)**

- \* Specify subscription id: Subscription ID
- \* Specify resource group: Resource group
- \* Specify workspace id: Workspace ID
- \* Identifier: Incident
- \* Specify alert / incident: System alert ID
- \* Specify comment:
  - Recorded Future Enrichment for: Current item
  - Link to the Intelligence Card: intelCard
  - Risk Score: risk
  - Evidence Details: evidenceDetails

Connected to test1@adrianporcescucloud.onmicrosoft.com. [Change connection.](#)

12. Save the Logic App

In order to have this Logic App run automatically when an alert is generated by an Analytic Rule you have to select the Logic App as an “Automated response” Playbook for a particular rule:

**Analytic rule wizard - Edit existing rule**

General | Set rule logic | Incident settings (Preview) | **Automated response** | Review and create

Select a playbook to be run automatically when your analytic rule generates an alert.

You only see playbooks in your selected subscriptions and for which you have permissions.

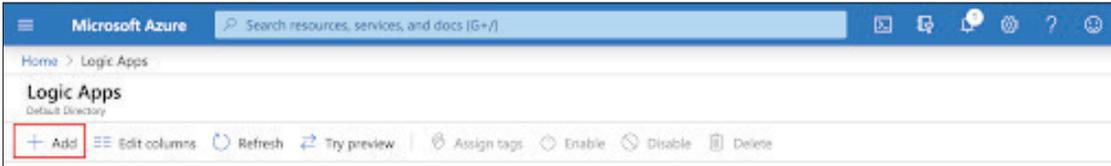
Selected playbooks: RF\_IP\_Enrichment\_2

Name	Status	Trigger kind
<input type="checkbox"/> RF_IP_Enrichment	Disabled	Azure Sentinel
<input checked="" type="checkbox"/> RF_IP_Enrichment_2	Enabled	Azure Sentinel

## Enriching Indicators (in bulk) with Recorded Future Context

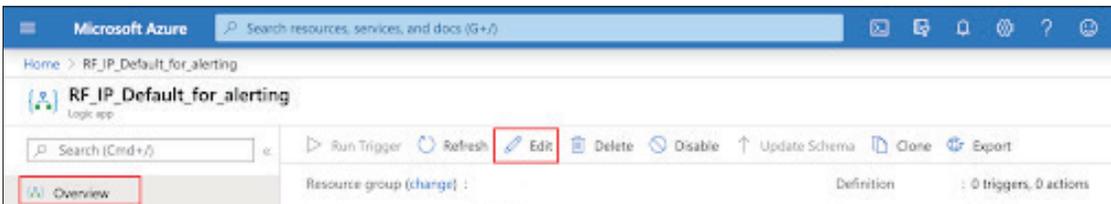
In this section we will present, via a Logic App, how to extract indicators from a Microsoft Sentinel Alert and enrich (with Recorded Future risk context via the SOAR API) the Incident, via adding a comment, with Recorded Future Risk Score, Risk Rules, Evidence Details and Link to the Recorded Future Intelligence Card.

1. Create a new Logic App by clicking on “Add” in the Logic Apps section



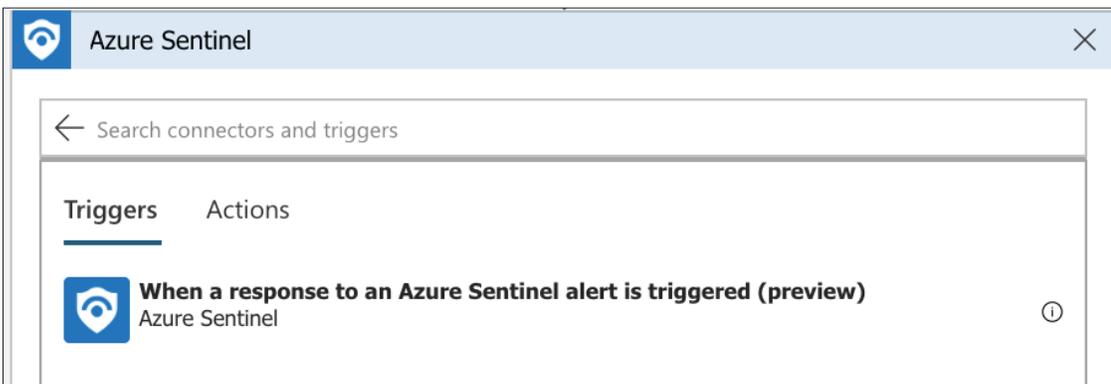
2. Provide a name for your Logic App and additional required dependencies and finalise the creation

3. Once the Logic App has been created, access it and in the “Overview” section click on the “Edit” button to access the Logic App Designer



4. Start with a “Blank Logic App” template

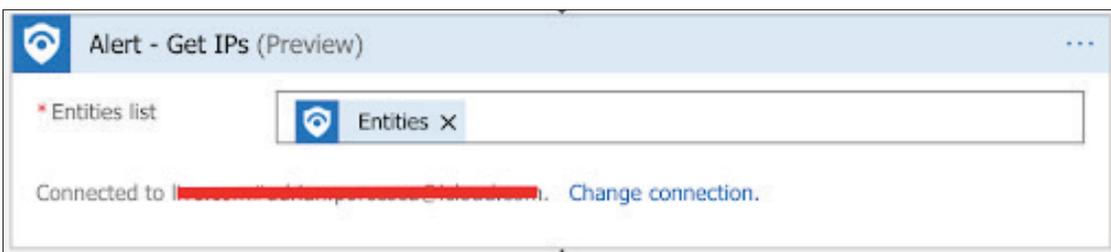
5. In order to run this Logic App when an response to an Azure Sentinel alert is triggered we need to add as a trigger “When a response to an Azure Sentinel alert is triggered” action block



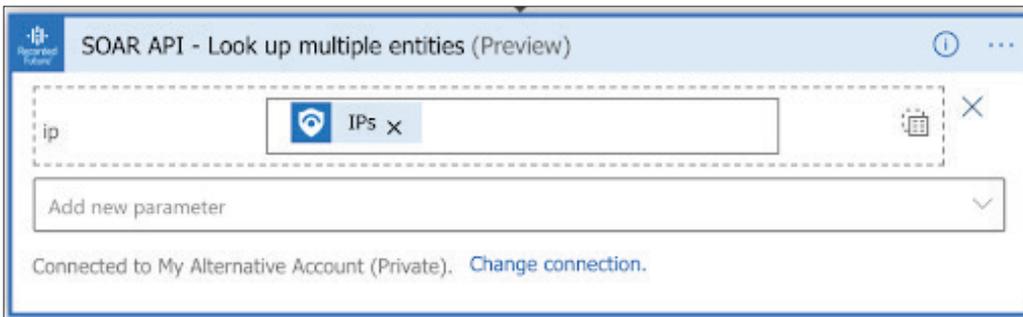
6. When configuring the trigger, chose a connection to use that has permissions to read from Azure Sentinel

7. In order to extract the indicators (we will focus on the indicators of IP type) that are reported in the incident we will add the “Alert - Get IPs” action.

8. Select via Dynamic Content the “Entities” from the previous block as the input for the “Entities List” field:

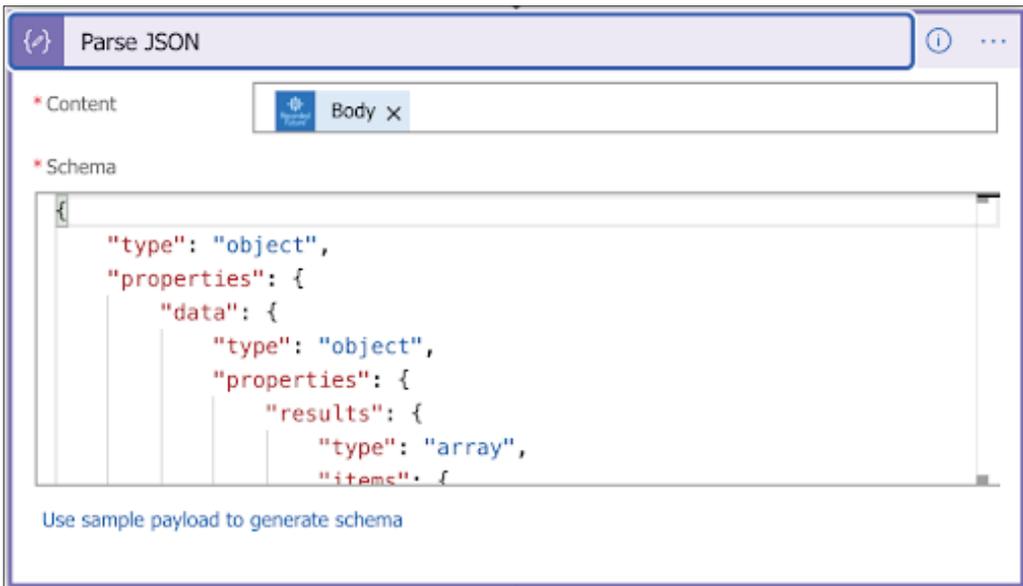


9. Use the “SOAR API - Look up multiple entities” action block in order to enrich in bulk, with Risk related information, all the selected indicators from the Sentinel Alert

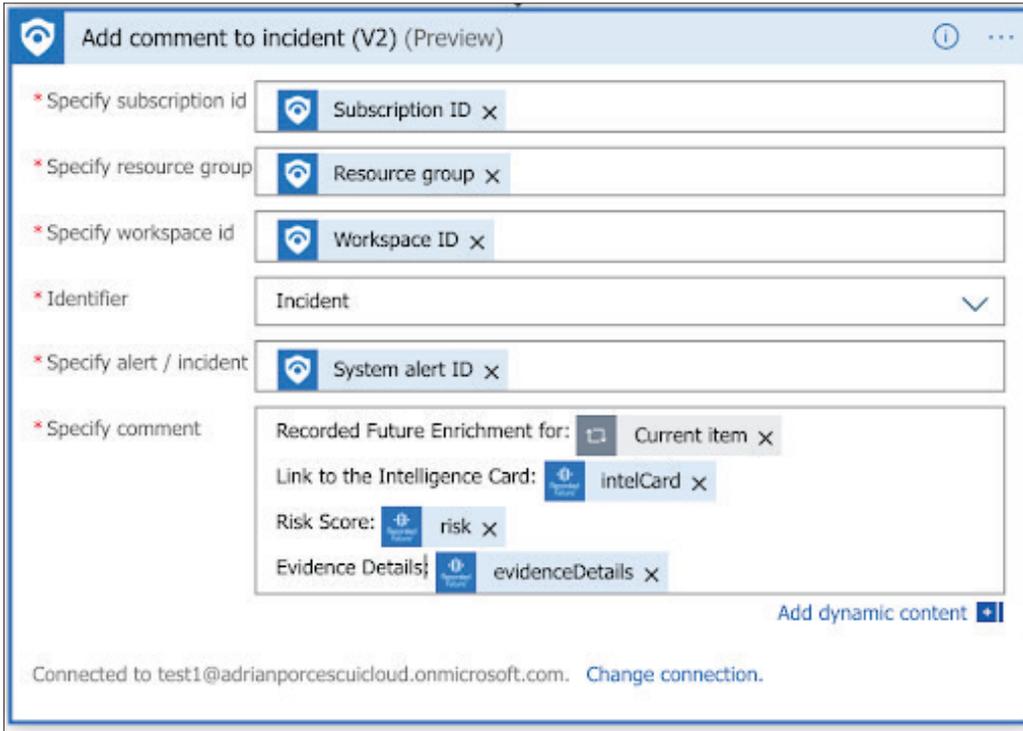


You can click “Add new parameters” to add other types of inputs (domain, url, hash, vulnerability) to the list of indicators for the action to enrich in bulk.

10. Add a Parse JSON block to parse the output of the bulk enrichment action block (parser available in section [A1.6 of the APPENDIX](#))

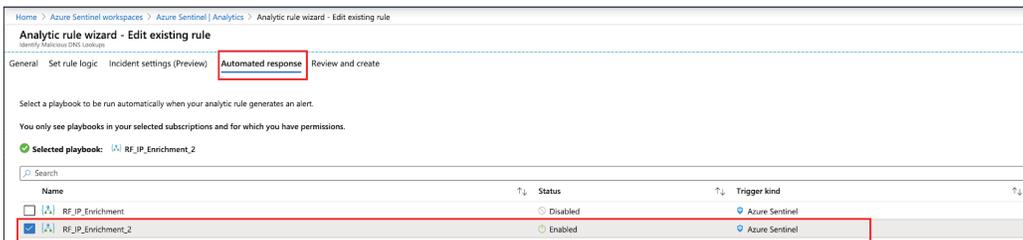


11. To push the newly enriched content as a comment, back into the incident, add a “Add comment to incident” action. Configure the fields according to your environment and the values you want to make available in the comment:



## 12. Save the Logic App

In order to have this Logic App run automatically when an alert is generated by an Analytic Rule you have to select the Logic App as an “Automated response” Playbook for a particular rule:



Name	Status	Trigger kind
RF_IP_Enrichment	Disabled	Azure Sentinel
RF_IP_Enrichment_2	Enabled	Azure Sentinel

## Recorded Future Alerts (Portal/UI)

Recorded Future Connector in Microsoft Azure contains dedicated actions that allow users to configure Logic Apps to pull from the Connect API, Recorded Future alerting rules and associated alert notifications. The Recorded Future Connector actions are following the standard behavior of the Recorded Future Alert API which is presented in more details [here](#).

In order to access/pull the JSON content of a triggered alerting notifications the following workflow needs to be implemented in your own Logic App:

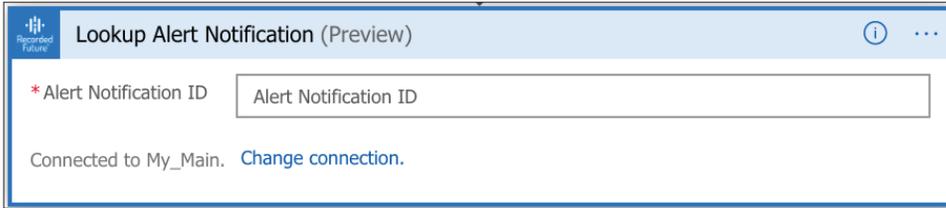
1. Identify the Alert Rule ID of the Alerting Rule you would like to pull triggered notifications. The Alert Rule ID can be obtained from the Recorded Future Portal/UI or using the dedicated action of the Recorded Future Connector, named "Search Alert Rules":

2. You can modify the total number of records to be returned and you can also input a string that the action will search for in the name of the Alerting Rules. If no text is inputted the search will return all rules available, up to the maximum value of records.

3. The next step is to search for any triggered notifications, for a particular Alert Rule ID (identified in the previous step). This can be achieved via the "Search Alert Notifications" action of the Recorded Future Connector:

4. For the Alert Rule ID field you can use the Dynamic output of the previous "Search Alert Rule" action block. You can filter by triggered datetime (All Elasticsearch compatible date formats are valid). Use the limit parameter to specify the maximum number of alerts. Use from to set the pagination offset of the request. Note there is a limit of returning only the top 1000 results from a search; queries requesting more than 1000 records will return an error.

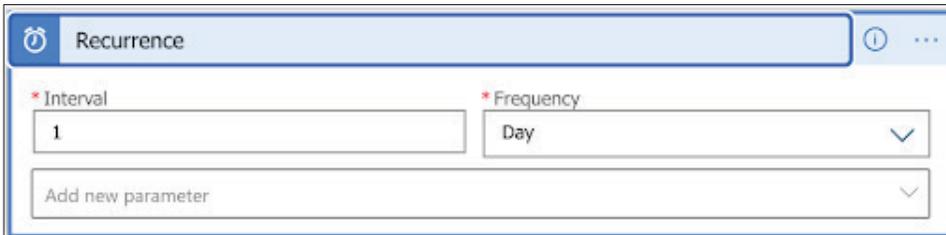
5. Once the Alert Notification IDs are presented in the output of the previous search block, you can use the “Lookup Alert Notification” action, to pull the full content of the Alert Notification:



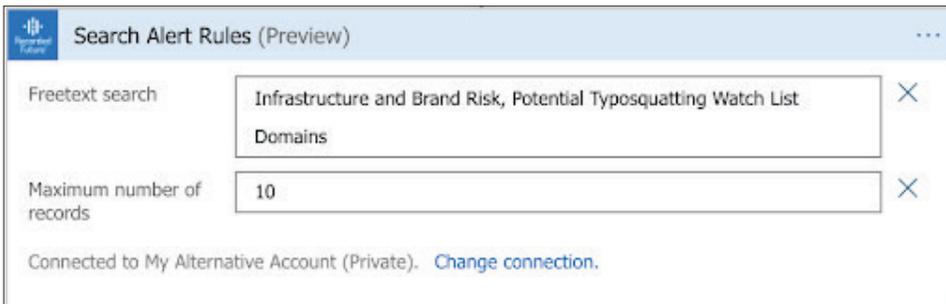
6. The response of the Lookup Notification action is presented as JSON formatted content.

### Use Case - Alert Notification on Domain Typosquat Registration

1. Add a Recurrence block configured according to the intervals off pulling the Alert Notifications. For this example i am pulling alert notifications once per day so the time to run the Logic App will be once per day



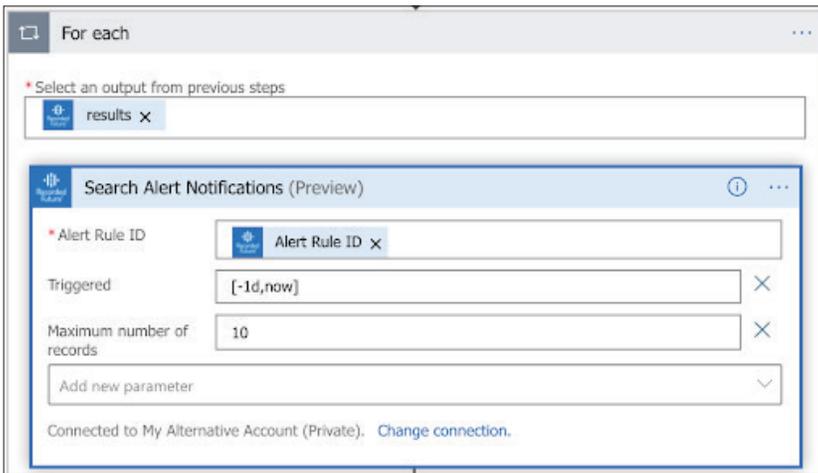
2. Search for the Alerting Rule that is associated with the Use Case:



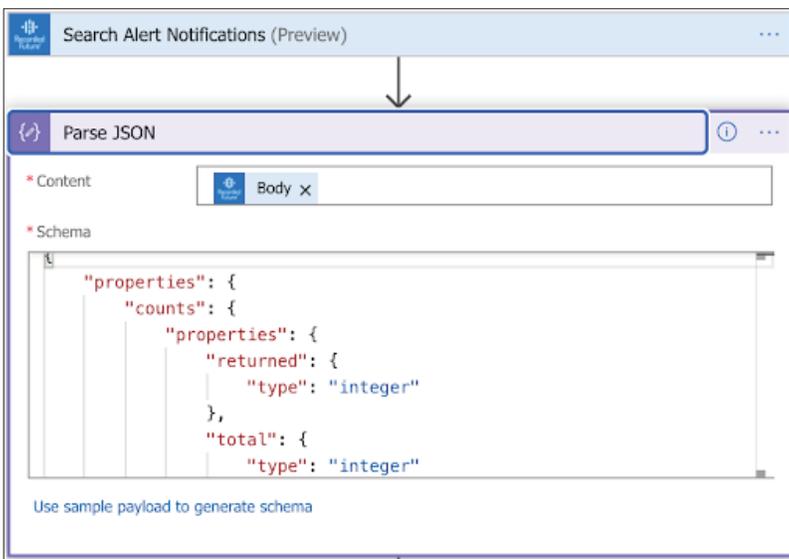
3. Cycle through all the resulting rules with a “For Each” block (as it could be that you have multiple alerts you want to cover in 1 workflow). If you are only working with one Alerting Rule and you have identified the Alerting Rule ID, you can skip Step 2 and Step 3



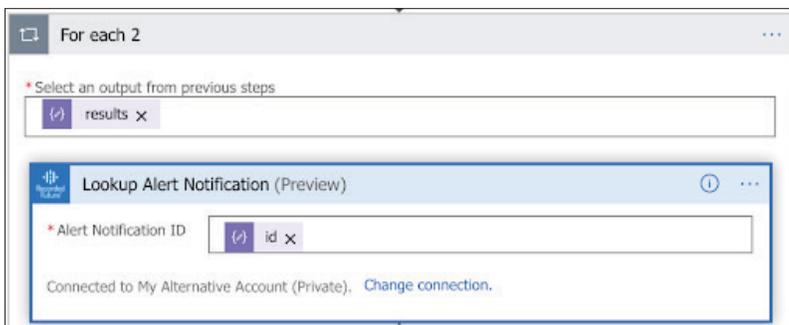
4. Search for the notifications that triggered for the Alert Rule ID of interest, in the last 1 day:



5. Add a "Parse JSON" block in order to parse the response from the Search Alert Notification block (the JSON schema can be found at [A1.7 APPENDIX section](#))



6. Add a "Lookup Alert Notification" action and add the "id" from the previous parsed content as the input for the block (it will generate an automatic add of an additional "For Each" block so it can cycle through multiple potential notifications of the same alert)



7. Depending on the requirements you can now manipulate the output of the Lookup Alert Notification action to perform any actions with the data inside.

For our example we will extract the reported Typosquatting Domain and the criticality level of it and add them to a custom table in Log Analytics

The screenshot shows a workflow configuration in Recorded Future. It starts with a 'Lookup Alert Notification (Preview)' block, followed by a 'Parse JSON 2' block. Below these is a 'For each 3' loop containing a 'Send Data (Preview)' block. The 'Send Data' block is configured with the following settings:

- \*Select an output from previous steps:** entities
- \*JSON Request body:**

```
{
  "Typo_Domain": "name",
  "Typo_Domain_Risk": "criticalityLabel"
}
```
- \*Custom Log Name:** RF\_TypoSquatting\_Domain
- Time-generated-field:** utcNow()
- Connected to Create Log 1:** [Change connection.](#)

8. The data will be delivered to a Log Analytics Custom Log with the name RF\_TypoSquatting\_Domain\_CL according to the name inserted in the configuration of the "Send Data" block.

>	7/13/2020, 2:35:50.130 PM	policies.google.com	Suspicious	RF_TypoSquatting_Domain_CL
>	7/13/2020, 2:36:19.905 PM	google.com.498721811506483.window-updates-service.com	Suspicious	RF_TypoSquatting_Domain_CL
>	7/13/2020, 2:36:33.415 PM	google.com.490705860985649.window-updates-service.com	Suspicious	RF_TypoSquatting_Domain_CL
>	7/13/2020, 2:35:56.521 PM	google.com.digitalmediahomebusiness.net	Suspicious	RF_TypoSquatting_Domain_CL

## Recorded Future Custom Connector

For enabling the connection between Microsoft Azure and Recorded Future API a Custom Connector was created that contains multiple configurable actions, that will be leveraged across this document for covering a wide range of use cases.

The available actions in the last version of the Recorded Future Connector are:

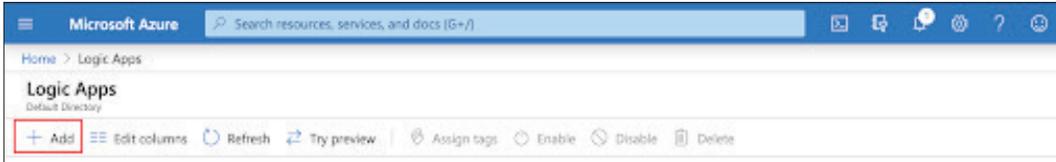
NAME	VISIBILITY	DESCRIPTION
Recorded Future RiskLists & SCF Download	important	Recorded Future RiskList & Security Control Feeds Download
IP Enrichment	important	IP Enrichment with Recorded Future data
IP Extension Enrichment	advanced	IP Enrichment with Recorded Future Extension Partner data
Domain Enrichment	important	Domain Enrichment with Recorded Future data
Domain Extension Enrichment	advanced	Domain Enrichment with Recorded Future Extension Partner data
URL Enrichment	important	URL Enrichment with Recorded Future data
URL Extension Enrichment	advanced	URL Enrichment with Recorded Future Extension Partner data
Hash Enrichment	important	Hash Enrichment with Recorded Future data
Hash Extension Enrichment	advanced	Hash Enrichment with Recorded Future Extension Partner data
Vulnerability Enrichment	advanced	Vulnerability Enrichment with Recorded Future data
Vulnerability Extension Enrichment	advanced	Vulnerability Enrichment with Recorded Future Extension Partner data
SOAR API - Look up multiple entities	important	SOAR API - Look up multiple entities (Specific Access is Required)
Search Alert Rules	advanced	Search Recorded Future UI Alert Rules
Search Alert Notifications	advanced	Search Alert Notifications
Lookup Alert Notification	advanced	Lookup Alert Notification

## Creating the tilndicators Batching Logic App

For optimisation purposes we will create separate Logic Apps, one for importing the Recorded Future indicators and one for publishing them as Microsoft Azure tilndicators (current section).

**Obs.** Please create dedicated tilndicators Batching Logic App for any workflow created for importing Recorded Future RiskLists into Microsoft Azure, associated with either Prevention or Detection use cases.

1. Create a new Logic App by clicking on “Add” in the Logic Apps section



- 2. Provide a name for your Logic App (for example “tilndicators\_Batching\_App”) and additional required dependencies and finalise the creation
- 3. Once the Logic App has been created, access it and in the “Overview” section click on the “Edit” button to access the Logic App Designer
- 4. Start with a “Blank Logic App” template
- 5. Add a “Batch Messages” trigger:



6. Configure the trigger using the following values:

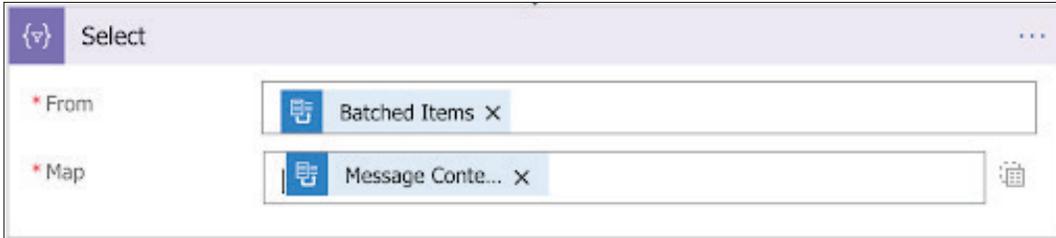
- Batch Mode: Inline
- Batch Name: (provide a name for your Batch. Ex: "RFIP90Plus")
- Release Criteria: Select from the list "Message Count Based" & "Schedule Based"
- Message Count: 100
- Schedule interval: Select an interval lower than the interval of importing the Recorded Future RiskList from the Importing Logic App (Example: If you are pulling Recorded Future Risklists every 1 hour, then this interval should be set for 45 min)

7. Add a "Select" Data Operations action as the next New Step

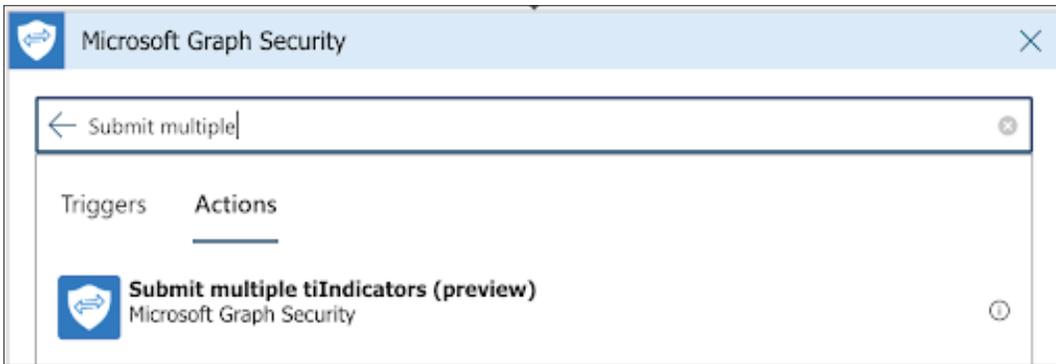
8. Press "Switch to text mode" in the newly added action

9. Configure the action as follows:

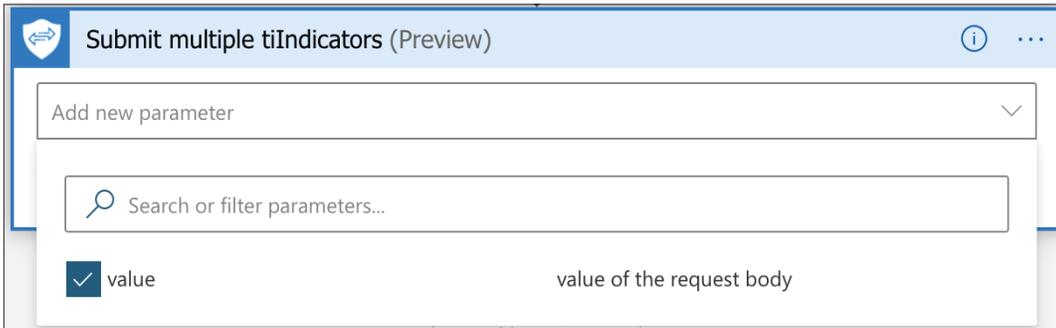
- In the “From” field select “Batched Items” from “Dynamic Content” window
- In the “Map” field select “Message Content” from “Dynamic Content” window



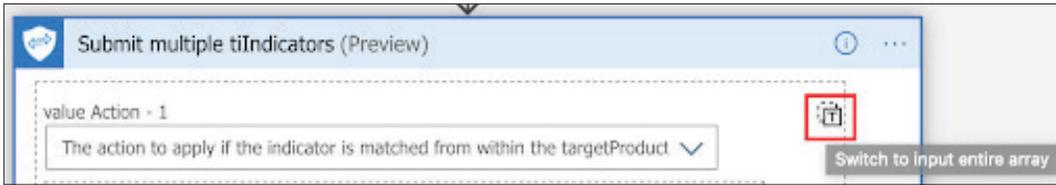
10. Add a “Submit multiple tiIndicators (preview)” Microsoft Graph Security action as the next New Step



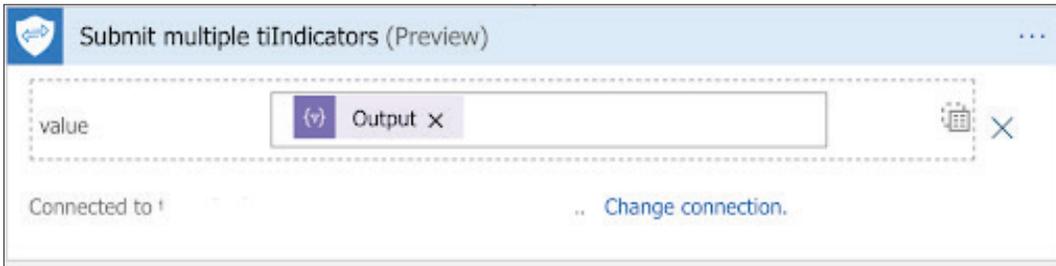
11. Enable the “value” parameter in the newly added action



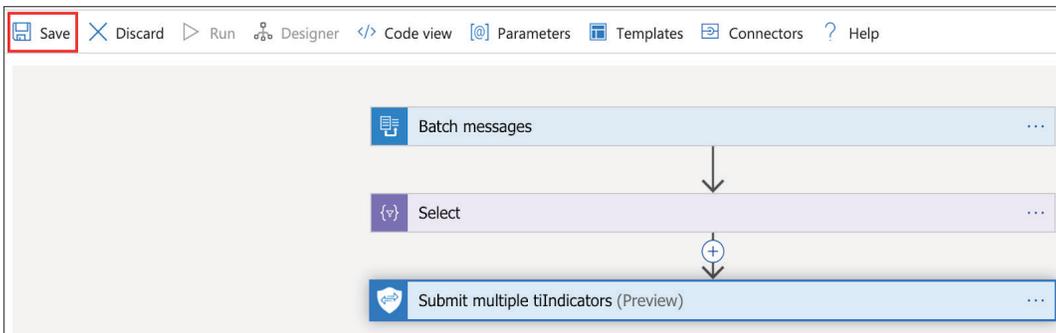
12. Press “Switch to input entire array”



13. In the “value” field select the “Output” of the previous “Select” action, from the “Dynamic content” window



14. Save the Logic App

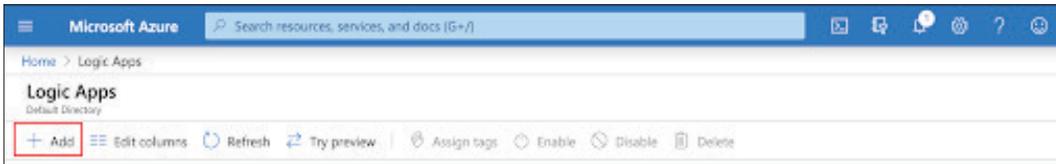


## Creating the Recorded Future Intelligence importing Logic App

For optimisation purposes we will create separate Logic Apps, one for importing the Recorded Future indicators (current section) and one for publishing them as Microsoft Azure indicators.

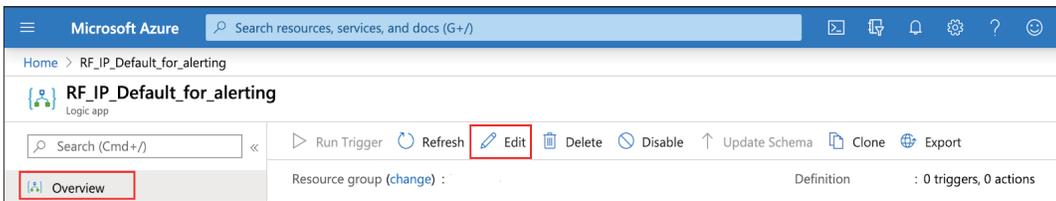
**Obs.** Please create the dedicated indicators Batching Logic App before creating this import Logic App, as this one will contain references to the indicators Batching Logic App.

1. Create a new Logic App by clicking on “Add” in the Logic Apps section



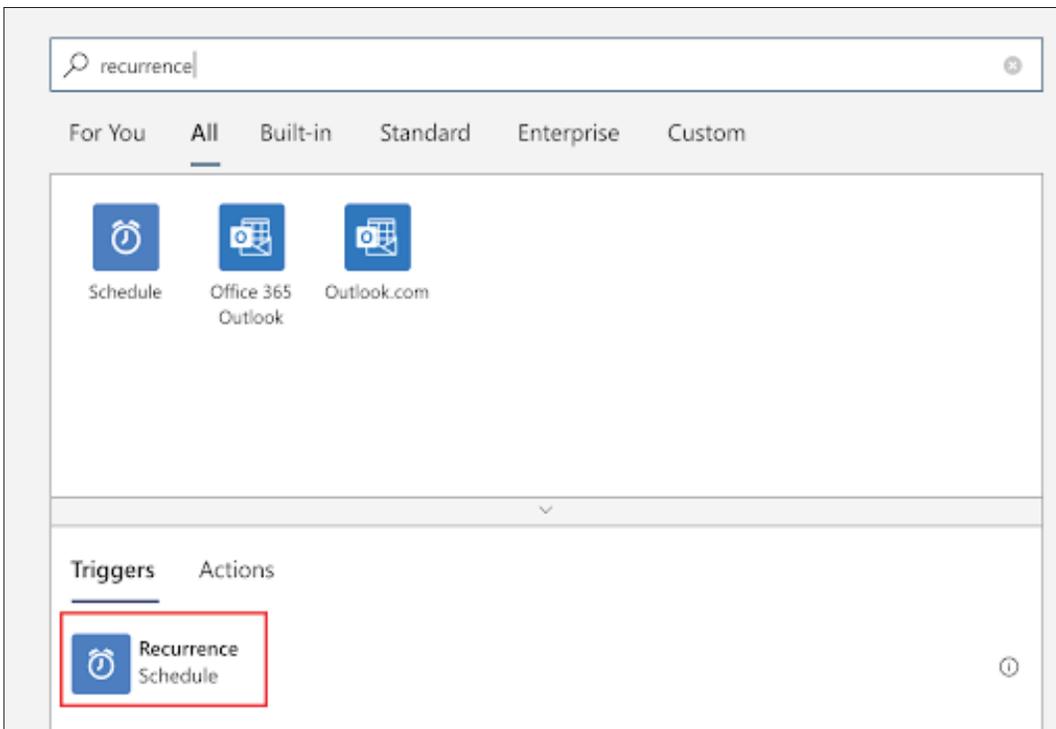
2. Provide a relevant name (that indicates the type of imported indicators) for your Logic App and additional required dependencies and finalise the creation

3. Once the Logic App has been created, access it and in the “Overview” section click on the “Edit” button to access the Logic App Designer



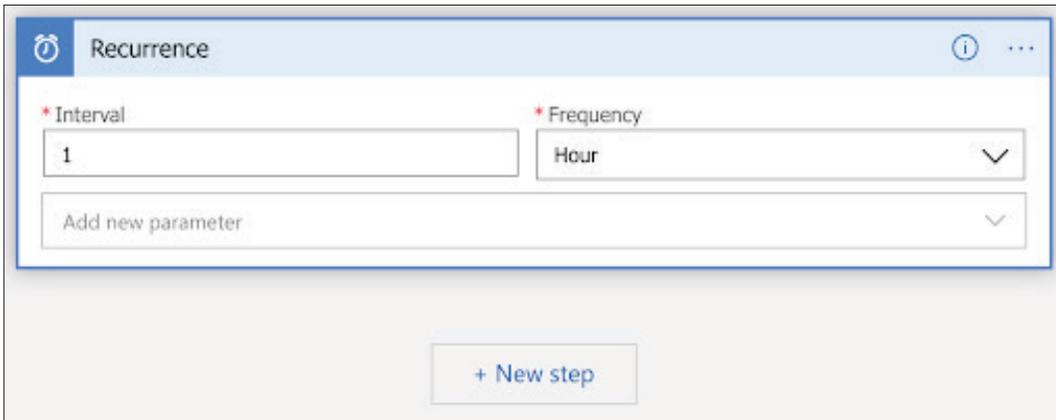
4. Start with a “Blank Logic App” template

5. In order to import the Recorded Future Intelligence at scheduled intervals we add a “Recurrence” trigger



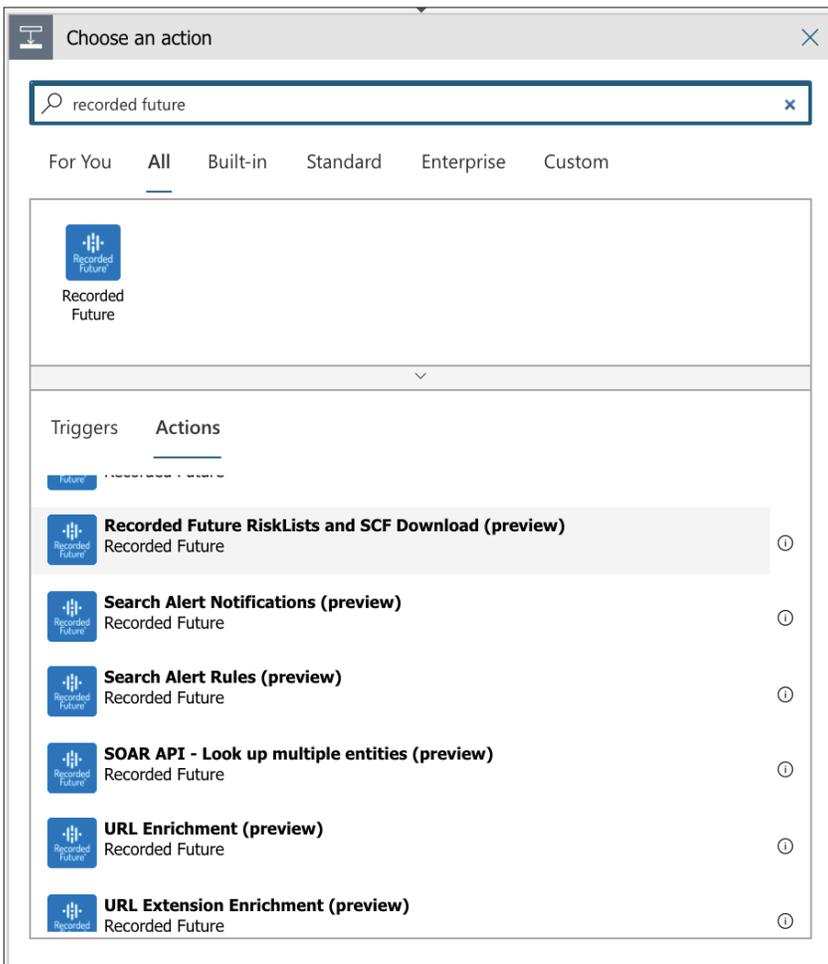
6. Configure the time interval for pulling the Recorded Future Risk List according to the recommended intervals for each type of indicator (more information around the recommended intervals can be found [here](#)).

In the case of an IP Risk Lists the recommendation is every 1 hour so we will set the “Recurrence” trigger to 1 hour

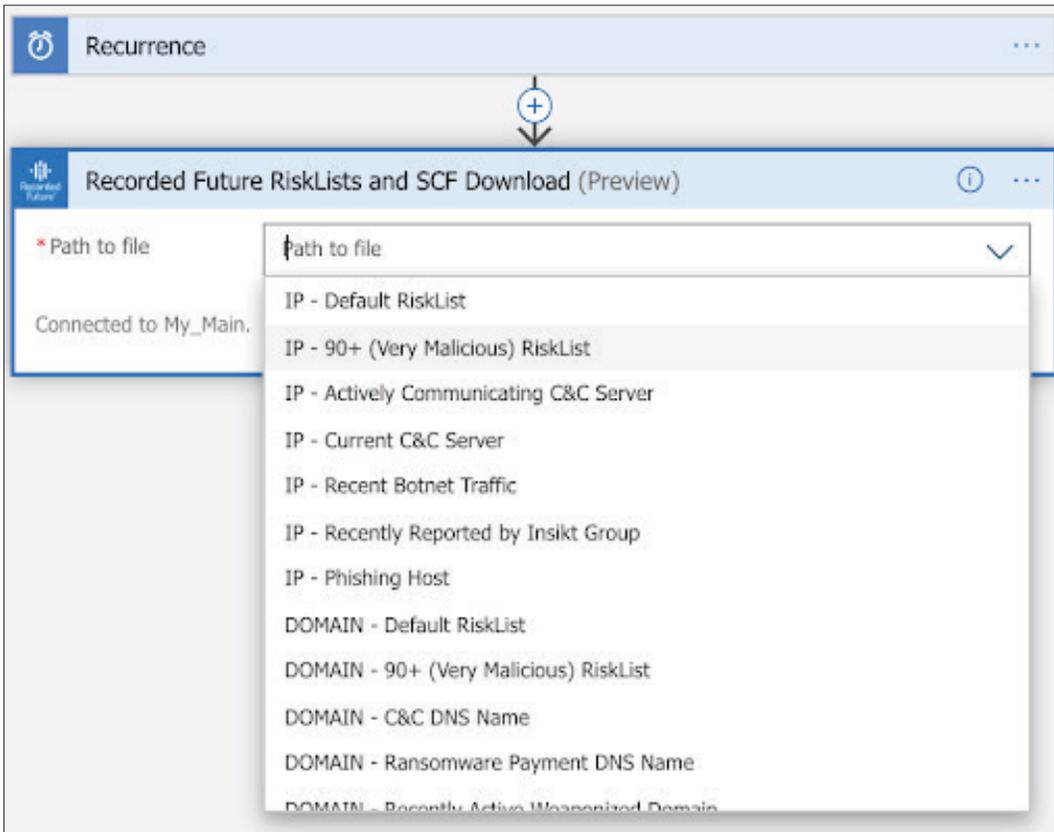


7. Click “New Step” to add the next connector

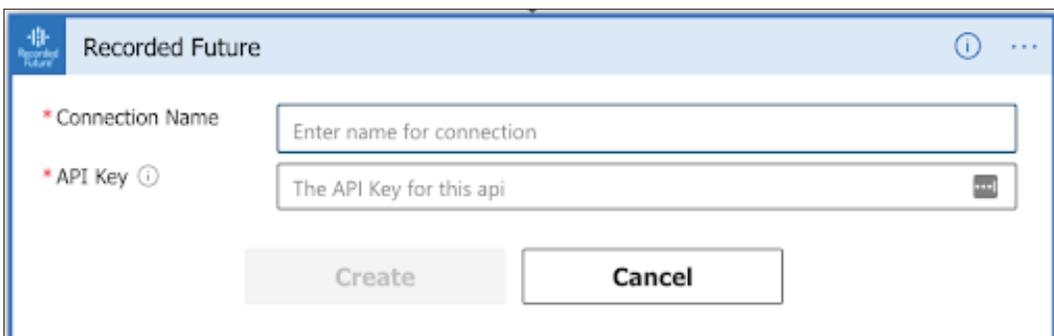
8. To enable the pull of a specific Recorded Future dataset we will use the “Recorded Future RiskLists & SCF Download” action



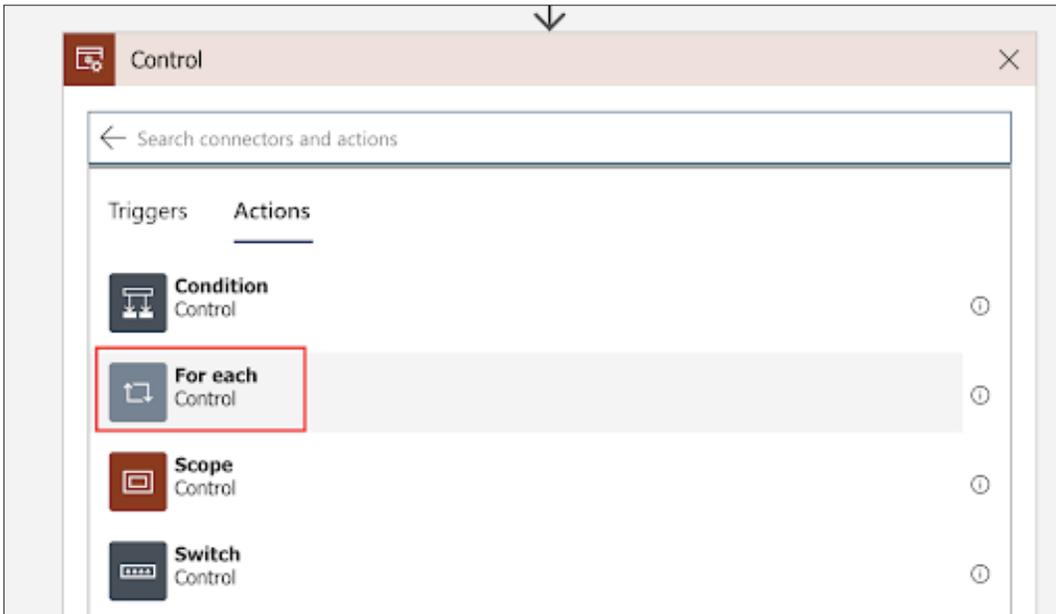
9. Select the dataset you want to download according to the desired use case. Details about the available datasets are presented in the Threat Prevention and Threat Detections sections at the beginning of the document:



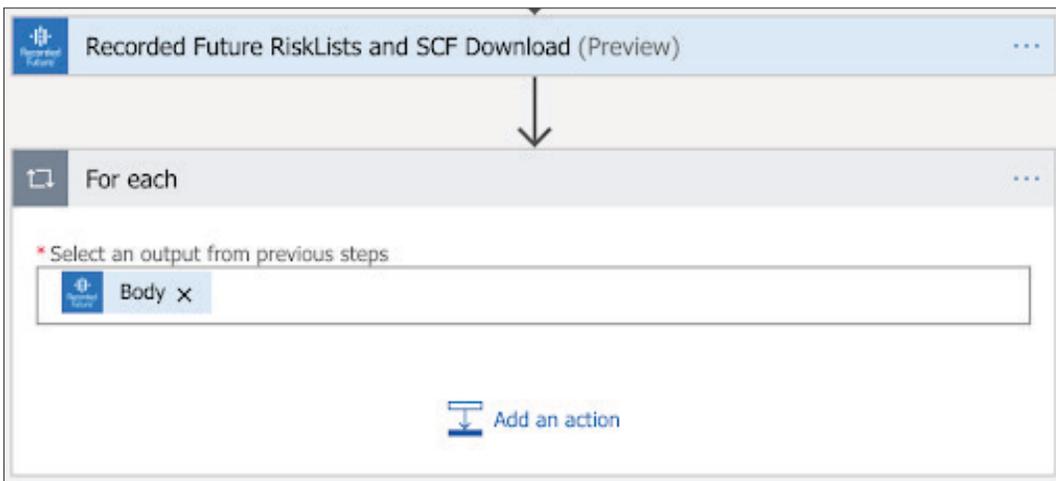
10. Select an existing connection or create a new one by introducing the Recorded Future API Token:



11. Recorded Future provides the data sets in JSON format, so as the output (Body) of the previous action contains a list/array of JSON objects we will add next a “For Each” (Control) action that allows us to cycle through all the elements of the list:

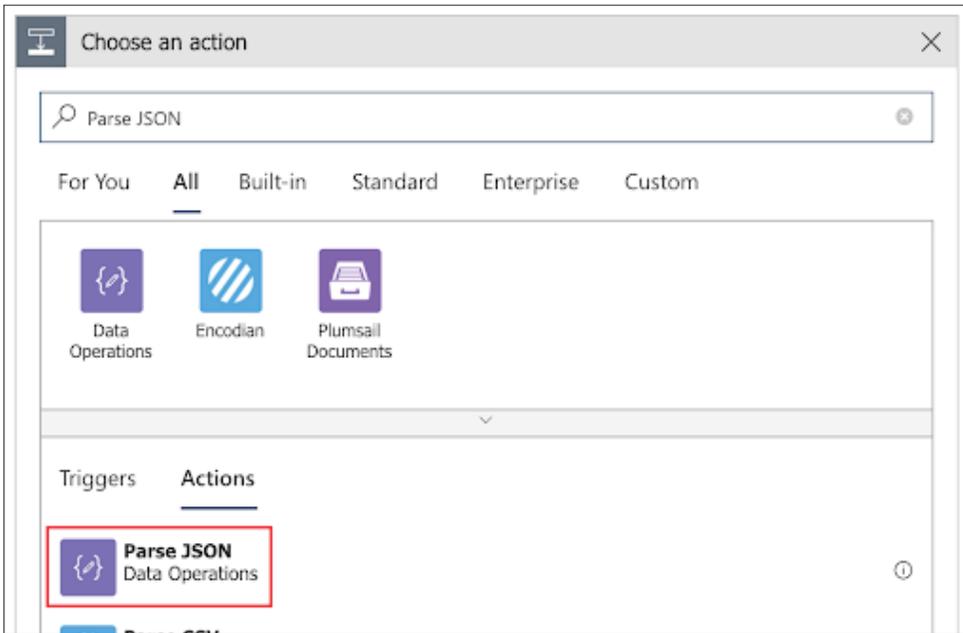


12. Select as the input for this action, the output (Body) of the previous “Recorded Future RiskLists & SCF Download” action:

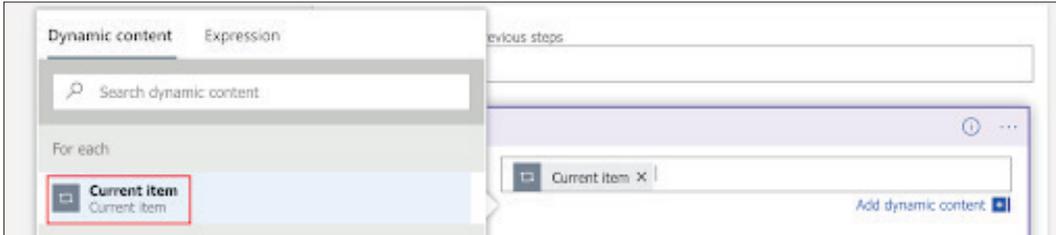


13. Click on “Add new action” inside the “For each” action block to add a new action to be performed on each one of the cycle steps.

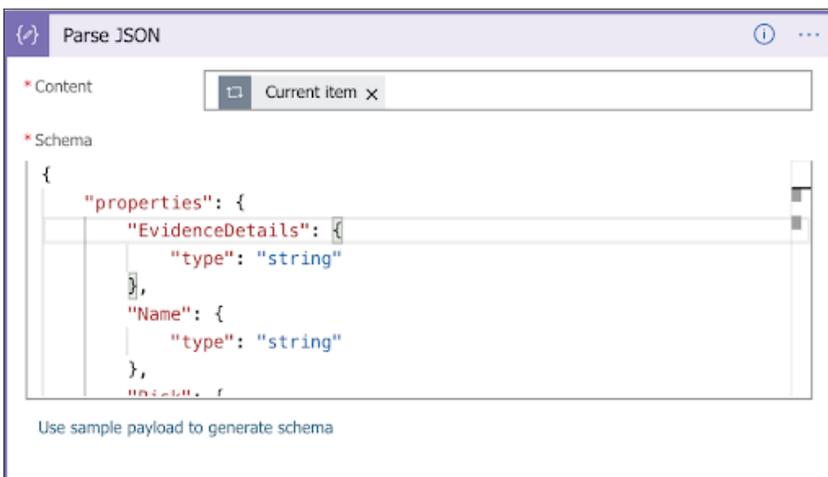
14. Add the “Parse JSON” as the new action, in order to properly parse each JSON object containing the indicators and the associated context. This enables us further on to refer to each one of the attributes of the indicators, based on the type of imported dataset:



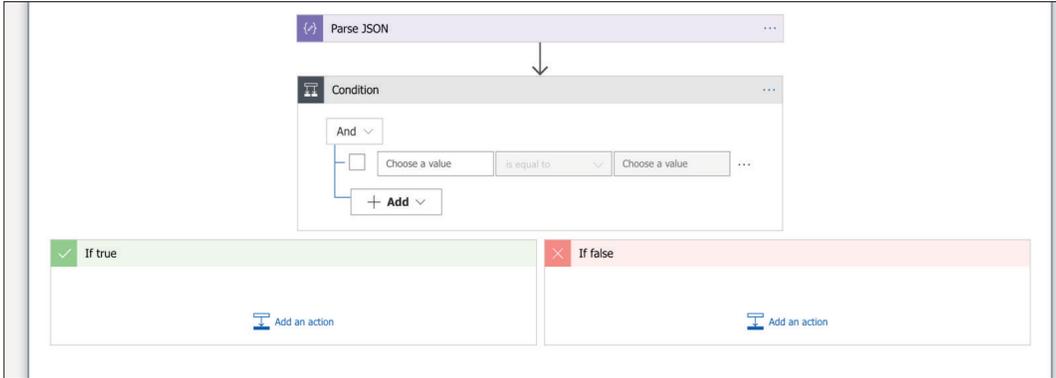
15. Select “Current item” from the “For each” action as the input for “Parse JSON”:



16. The next step is inputting the schema for parsing the JSON. Samples of schemas for this section can be found in the [Appendix Section](#). Depending on the type of data that is imported (at step 9 of this section) a corresponding JSON Schema will be provided.



17. (Optional) If we want to filter some indicators based on the risk score (or any other parameter provided by the previous JSON Parser action) we can add a “Condition” action

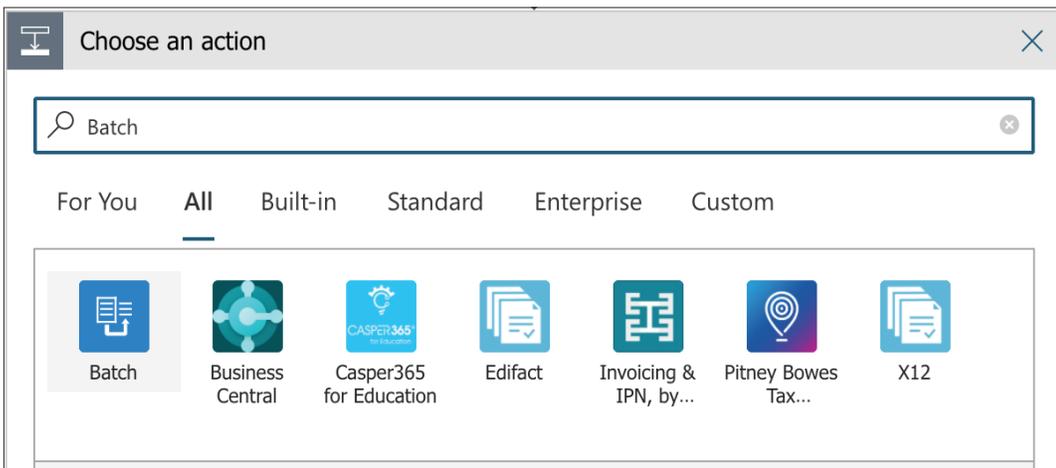


For example if we want to set the condition to filter the indicators with a score higher or equal than 80 the condition block will have the following settings:

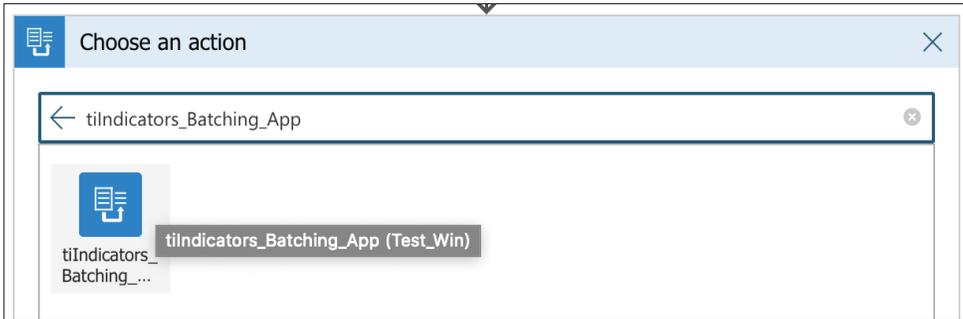
- The value 1 should look similar to “int(body('Parse\_JSON')?['Risk'])”
- The value 2 should be 80
- The condition should be “is greater than or equal to”

18. The last step is to define the full set of indicator properties that will be sent for batching to the “tiIndicators Batching Logic App”. **Obs.** If the condition was added then this new step will be created under “If true” so it will only happen if the condition is fulfilled.

19. Click on “Add an action”, search for “Batch” and click on “Batch”

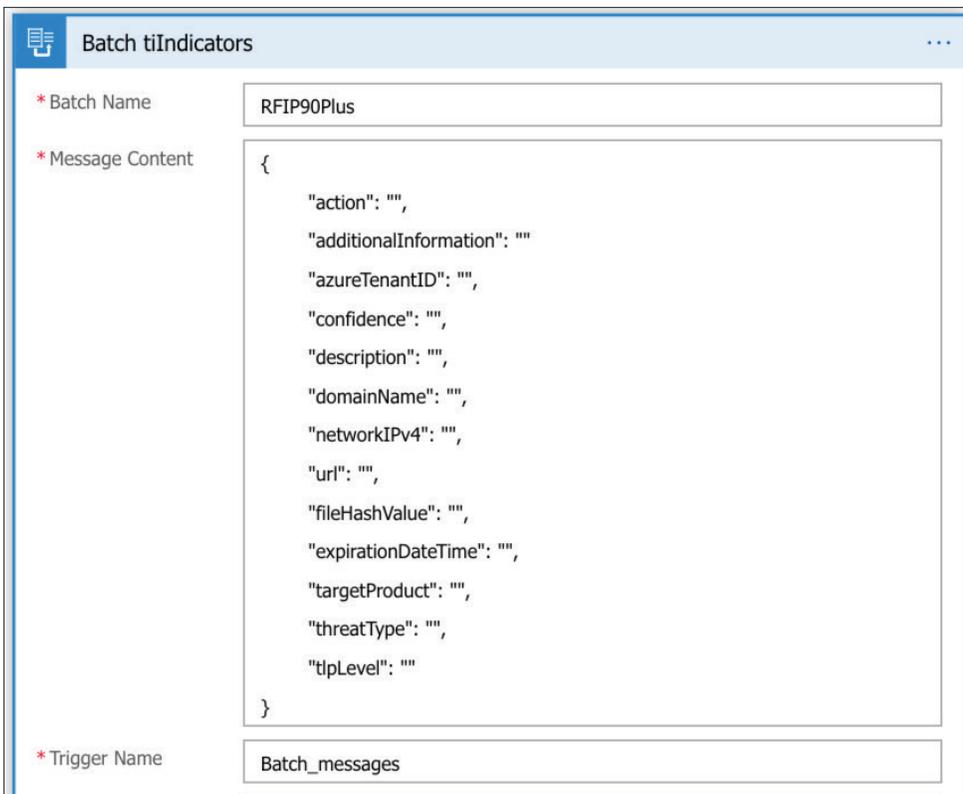


20. Select the option according to how you named your Logic App for Batching (In our case the name was: tiIndicators\_Batching\_App):



21. Configure the “tiIndicators Batching App”action accordingly:

- Batch Name: Input the name defined for the Batching App trigger (set in the previous section at step 6)
- Trigger Name: “Batch\_messages” (automatically populated)
- Workflow: path to the “tiIndicators Batching App” Logic App (automatically populated)
- Message Content: Different based on what types of indicators are imported by the workflow and what is the use case to be covered. Please follow the next steps in order to build this message content:
  - Copy the content from [A2 APPENDIX section](#)



- Populate each of the values according to the type of data you are importing, the type of use case etc. Some recommendations are:
  - “action” - “alert” if the use case is detection or “block” if the use cases is prevention
  - “additional Information” - for detection use cases populate this value with the “Evidence Details” from Recorded Future data Dynamic Content (for prevention use cases you can leave this empty)

```
"additionalInformation": "{ EvidenceDetails X ",
"azureTenantID": "108e704 body('Parse_JSON')['EvidenceDetails']?
['EvidenceDetails']"
```

- “azureTenantID” - input the destination Azure Tenant ID
- “Confidence” - for detection use cases populate this value with the “Risk” from Recorded Future data via the Dynamic Content (for prevention use cases you can leave this empty)

```
"confidence": "{ Risk X ",
"Recorded Future - Domain Def body('Parse_JSON')['Risk']"
```

- “description” - populate this with the name of the source Recorded Future dataset (Ex: “Recorded Future IP Actively Communicating C&C Server” when importing Actively Communicating C&C Server IP Risklist)
- “domainName” - populate this value with the “Name” from Recorded Future data via the Dynamic Content only if you are importing domain IOCs (for other you can leave it empty)

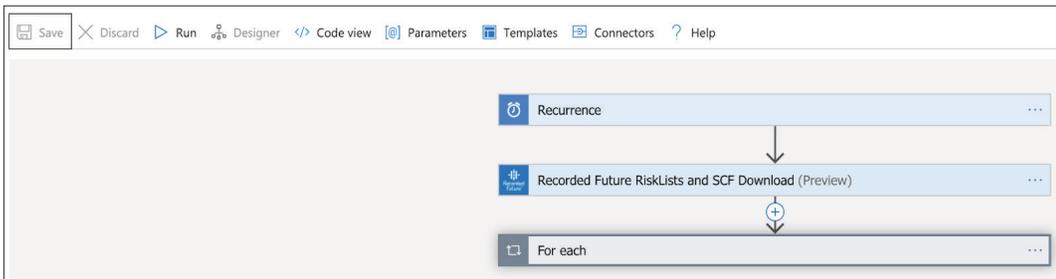
```
"domainName": "{ Name X ",
" body('Parse_JSON')['Name']"
```

- “networkIPv4” / “networkDestinationIPv4” - populate this value with the “Name” from Recorded Future data via the Dynamic Content only if you are importing IP IOCs (for other you can leave it empty)
- “url”- populate this value with the “Name” from Recorded Future data via the Dynamic Content only if you are importing URL IOCs (for other you can leave it empty)
- “fileHashValue” - populate this value with the “Name” from Recorded Future data via the Dynamic Content only if you are importing HASH IOCs (for other you can leave it empty)
- “expirationDateTime” - Expiration date/time depends on the interval of importing the IOCs (Step 6). If the interval of importing is set to 1 hour, then the Expiration time should be in 1 hour from the moment of the creation. So we set the value to addHours(utcNow(),1) using the Expressions section from “Dynamic Content”

```
"expirationDateTime": "fx addHours(...) X ",
: addHours(utcNow(),1)
```

- “targetProduct” - input “Azure Sentinel” if the use case is detection via Microsoft Sentinel or “Microsoft Defender ATP” if the use case is blocking/alerting via Microsoft Defender ATP
- “threatType” - choose one of the followings depending on what type of Recorded Future data set you have chosen to import (step 9): Botnet, C2, CryptoMining, Darknet, DDoS, MaliciousUrl, Malware, Phishing, Proxy, PUA, WatchList
- “tIpLevel” - choose one of the following: unknown, white, green, amber, red

## 22. Save the Logic App



## APPENDIX

### A1 - JSON Parsing Schemas

#### A1.1 - Recorded Future RiskLists (Detection) - Full List

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "EvidenceDetails": {
        "type": "object",
        "properties": {
          "EvidenceDetails": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "Rule": {
                  "type": "string"
                },
                "EvidenceString": {
                  "type": "string"
                },
                "CriticalityLabel": {
                  "type": "string"
                },
                "Timestamp": {
                  "type": "integer"
                },
                "Criticality": {
                  "type": "integer"
                }
              }
            },
            "required": [
              "Rule",
              "EvidenceString",
              "CriticalityLabel",
              "Timestamp",
              "Criticality"
            ]
          }
        }
      }
    }
  },
}
```

```

    "Name": {
      "type": "string"
    },
    "Risk": {
      "type": "integer"
    },
    "riskString": {
      "type": "string"
    }
  },
  "required": [
    "EvidenceDetails",
    "Name",
    "Risk",
    "riskString"
  ]
}
}

```

#### A1.2 - Recorded Future RiskLists (Detection) - Only one indicator

```

{
  "type": "object",
  "properties": {
    "EvidenceDetails": {
      "type": "object",
      "properties": {
        "EvidenceDetails": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "Rule": {
                "type": "string"
              },
              "EvidenceString": {
                "type": "string"
              },
              "CriticalityLabel": {
                "type": "string"
              },
              "Timestamp": {
                "type": "integer"
              }
            }
          }
        }
      }
    }
  }
}

```



```
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "port": {
          "type": "integer"
        },
        "protocol": {
          "type": "string"
        }
      },
      "required": [
        "port",
        "protocol"
      ]
    },
    "malware": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "last_seen": {
      "type": "string"
    }
  },
  "required": [
    "ip",
    "ports",
    "malware",
    "last_seen"
  ]
},
"timestamp": {
  "type": "string"
}
}
```

### A1.3 - Recorded Future Security Control Feed - Weaponized Domains

```

{
  "type": "object",
  "properties": {
    "count": {
      "type": "integer"
    },
    "results": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "domain": {
            "type": "string"
          },
          "last_seen": {
            "type": "string"
          },
          "service_provider": {
            "type": "string"
          },
          "detection_strings": {
            "type": "object",
            "properties": {
              "phishing site": {
                "type": "boolean"
              },
              "spam site": {
                "type": "boolean"
              },
              "mining site": {
                "type": "boolean"
              },
              "malicious site": {
                "type": "boolean"
              },
              "suspicious site": {
                "type": "boolean"
              },
              "malware site": {
                "type": "boolean"
              }
            }
          }
        }
      }
    }
  }
}

```





```
},
"score": {
  "type": "integer"
},
"evidenceDetails": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "mitigationString": {},
      "timestamp": {
        "type": "string"
      },
      "criticalityLabel": {
        "type": "string"
      },
      "evidenceString": {
        "type": "string"
      },
      "rule": {
        "type": "string"
      },
      "criticality": {
        "type": "integer"
      }
    }
  },
  "required": [
    "mitigationString",
    "timestamp",
    "criticalityLabel",
    "evidenceString",
    "rule",
    "criticality"
  ]
},
"riskString": {
  "type": "string"
},
"rules": {
  "type": "integer"
},
}
```



```
"type": "object",
"properties": {
  "level": {
    "type": "integer"
  },
  "rule": {
    "type": "object",
    "properties": {
      "count": {
        "type": "integer"
      },
      "mostCritical": {
        "type": "string"
      },
      "maxCount": {
        "type": "integer"
      },
      "evidence": {
        "type": "object",
        "properties": {
          "relatedNote": {
            "type": "object",
            "properties": {
              "count": {
                "type": "integer"
              },
              "timestamp": {
                "type": "string"
              },
              "description": {
                "type": "string"
              },
              "rule": {
                "type": "string"
              },
              "level": {
                "type": "integer"
              }
            }
          },
          "linkedToCyberExploit": {
            "type": "object",
```

```
    "properties": {
      "count": {
        "type": "integer"
      },
      "timestamp": {
        "type": "string"
      },
      "description": {
        "type": "string"
      },
      "rule": {
        "type": "string"
      },
      "level": {
        "type": "integer"
      }
    }
  },
  "recentMalwareActivity": {
    "type": "object",
    "properties": {
      "count": {
        "type": "integer"
      },
      "timestamp": {
        "type": "string"
      },
      "description": {
        "type": "string"
      },
      "rule": {
        "type": "string"
      },
      "level": {
        "type": "integer"
      }
    }
  },
  "nistHigh": {
    "type": "object",
    "properties": {
      "count": {
```

```
        "type": "integer"
      },
      "timestamp": {
        "type": "string"
      },
      "description": {
        "type": "string"
      },
      "rule": {
        "type": "string"
      },
      "level": {
        "type": "integer"
      }
    }
  },
  "cyberSignalHigh": {
    "type": "object",
    "properties": {
      "count": {
        "type": "integer"
      },
      "timestamp": {
        "type": "string"
      },
      "description": {
        "type": "string"
      },
      "rule": {
        "type": "string"
      },
      "level": {
        "type": "integer"
      }
    }
  },
  "linkedToRecentCyberExploit": {
    "type": "object",
    "properties": {
      "count": {
        "type": "integer"
      },
    },
  },
}
```

```
    "timestamp": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "rule": {
      "type": "string"
    },
    "level": {
      "type": "integer"
    }
  }
},
"pocUnverified": {
  "type": "object",
  "properties": {
    "count": {
      "type": "integer"
    },
    "timestamp": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "rule": {
      "type": "string"
    },
    "level": {
      "type": "integer"
    }
  }
}
},
"summary": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "count": {
```

```

        "type": "integer"
      },
      "level": {
        "type": "integer"
      }
    },
    "required": [
      "count",
      "level"
    ]
  }
}
},
"context": {
  "type": "object",
  "properties": {
    "malware": {
      "type": "object",
      "properties": {
        "rule": {
          "type": "object",
          "properties": {
            "count": {
              "type": "integer"
            },
            "maxCount": {
              "type": "integer"
            }
          }
        },
        "score": {
          "type": "integer"
        }
      }
    }
  },
  "public": {
    "type": "object",
    "properties": {
      "rule": {
        "type": "object",
        "properties": {

```

```

        "maxCount": {
          "type": "integer"
        }
      },
      "summary": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "count": {
              "type": "integer"
            },
            "level": {
              "type": "integer"
            }
          },
          "required": [
            "count",
            "level"
          ]
        }
      },
      "mostCriticalRule": {
        "type": "string"
      },
      "score": {
        "type": "integer"
      }
    }
  },
  "score": {
    "type": "integer"
  }
}
},
"required": [
  "entity",
  "risk"
]

```

```

        ]
      }
    }
  },
  "counts": {
    "type": "object",
    "properties": {
      "returned": {
        "type": "integer"
      },
      "total": {
        "type": "integer"
      }
    }
  }
}
}
}

```

#### **A1.7 - Recorded Future Alert Notifications Search**

```

{
  "type": "object",
  "properties": {
    "data": {
      "type": "object",
      "properties": {
        "results": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "review": {
                "type": "object",
                "properties": {
                  "assignee": {},
                  "noteAuthor": {},
                  "note": {},
                  "status": {
                    "type": "string"
                  },
                },
              },
              "noteDate": {}
            }
          },
        },
      },
    },
  },
}

```

```
    "url": {
      "type": "string"
    },
    "rule": {
      "type": "object",
      "properties": {
        "url": {
          "type": "string"
        },
        "name": {
          "type": "string"
        },
        "id": {
          "type": "string"
        }
      }
    },
    "triggered": {
      "type": "string"
    },
    "id": {
      "type": "string"
    },
    "title": {
      "type": "string"
    },
    "type": {
      "type": "string"
    }
  },
  "required": [
    "review",
    "url",
    "rule",
    "triggered",
    "id",
    "title",
    "type"
  ]
}
}
},
"counts": {
```



```
"fileType": "String",  
"id": "String (identifier)",  
"ingestedDateTime": "String (timestamp)",  
"isActive": true,  
"killChain": ["String"],  
"knownFalsePositives": "String",  
"lastReportedDateTime": "String (timestamp)",  
"malwareFamilyNames": ["String"],  
"networkCidrBlock": "String",  
"networkDestinationAsn": 1024,  
"networkDestinationCidrBlock": "String",  
"networkDestinationIPv4": "String",  
"networkDestinationIPv6": "String",  
"networkDestinationPort": 1024,  
"networkIPv4": "String",  
"networkIPv6": "String",  
"networkPort": 1024,  
"networkProtocol": 1024,  
"networkSourceAsn": 1024,  
"networkSourceCidrBlock": "String",  
"networkSourceIPv4": "String",  
"networkSourceIPv6": "String",  
"networkSourcePort": 1024,  
"passiveOnly": true,  
"severity": 1024,  
"tags": ["String"],  
"targetProduct": "String",  
"threatType": "String",  
"tlpLevel": "string",  
"url": "String",  
"userAgent": "String"  
}
```

## REFERENCES

Microsoft Azure tiIndicator resource type - <https://docs.microsoft.com/en-us/graph/api/resources/tiindicator?view=graph-rest-beta>

## About Recorded Future

Recorded Future delivers security intelligence to amplify the effectiveness of security and IT teams by informing decisions in real time with contextual, actionable intelligence. By analyzing data from open, dark, and proprietary sources, Recorded Future offers a singular, integration-ready view of threat information, risks to digital brand, vulnerabilities, third-party risk, geopolitical risk, and more.