**Forcepoint NGFW**

**Logic Apps Connector integration spec for Azure Sentinel**

# Introduction

Forcepoint Next Generation Firewall (NGFW) connects and protects people and the data they use throughout the enterprise network – all with the greatest efficiency, availability, and security. Forcepoint NGFWs can selectively and automatically whitelist or blacklist network traffic originating from particular applications on PCs, laptops, servers, file shares, and other endpoint devices based on highly granular endpoint contextual data. The Forcepoint NGFW solution consists of one or more Forcepoint NGFW Engines and the Forcepoint NGFW Security Management Center (SMC). The SMC is the management component of the Forcepoint NGFW solution.

Features:

* Forcepoint integration can be used to create a block list category for URL and IP addresses.
* Whitelisting / blacklisting by client application and version for device lockdown
* Application layer exfiltration protection
* Built-in IPsec and SSL VPN

# Authorization**:**

* API Key <http://www.websense.com/content/support/library/ngfw/v610/rfrnce/ngfw_6100_ug_smc-api_a_en-us.pdf>

# Pre-Requisite:

* Create a new **Security Policy Rule** or share an existing security policy rule.

# Sentinel scenarios through Force-Point integration

1. Events from various data sources are collected by Sentinel’s data connectors, particularly the Force-point data connector.
2. Sentinel Analytics rules are enabled, creating alerts which indicates possible attacks that are affiliated with external Ips and URLs. Incidents are created on these alerts, containing entities such as IP and URLs of the possible attacker.
3. Playbook templates are enabled and attached to these analytic rules.

# Playbook Templates:

## Scenario 1: Enrichment – IP Address

Sentinel Playbook is triggered when a new Incident is created over an analytic rule that indicates a IP with all the entities supported by Sentinel and enrich the incident with information of that IP.

Action flow:

1.  **When an Azure Sentinel Incident is triggered**
2.  **Entities – Get IP**
3.  **Get ‘Security Policy Rule’ of this IP**
4.  **Verify if IP is available/not available.**
5.  **Add a comment to the incident** with summary of the information gathered.

## Scenario 2: Response – IP Address

Sentinel Playbook is triggered when a new Incident is created over an analytic rule that indicates a IP with all the entities supported by Sentinel and provide options to SOC with different actions.

Action flow:

1.  **When an Azure Sentinel Incident is triggered**
2.  **Entities – Get IP**
3.  **Get ‘Security Policy Rule’ of this IP**
4. Icon

   Description automatically generated **Post a choice of options as the flow bot** (Teams adaptive card) on Teams channel about the new incident, containing:
   1. Informative summary of the incident, ( incident description, Incident Severity, alert product names, Incident URL dynamic fields)
   2. Incident configuration:
      1. Incident Severity choices (Low, Medium, High)
      2. Incident Status options (Close the incident with choices of classification reasons)
   3. Options choices for SOC:
      1. Close incident
      2. Update severity
5. **If IP was found** in the policy rule



* + 1. Icon

       Description automatically generated **In Teams adaptive card,** provide additional Options choices for SOC:
       - 1. Delete IP from Security Policy or option to skip this step.
    2.  **Delete IP** from the Security Policy by SOC choice.
    3.  **Change incident status** by user choice.
    4.  **Change incident severity** by user choice.
    5.  **Add a comment to the incident** with summary of action been taken.

1. **If IP is not found** in the policy rule
   * 1. Add IP to the Security Policy



* + 1.  **Change incident status** to close.
    2.  **Add a comment to the incident** with summary of information and action taken.

## Scenario 3: Enrichment – URL

Sentinel Playbook is triggered when a new Incident is created over an analytic rule that indicates a URL with all the entities supported by Sentinel and enrich the incident with information of that URL.

Action flow:

1.  **When an Azure Sentinel Incident is triggered**
2.  **Entities – Get URL**
3.  **Get ‘Security Policy Rule’ of this URL**
4.  **Verify if URL is available/not available**
5.  **Add a comment to the incident** with summary of the information gathered.

## Scenario 4: Response – URL

Sentinel Playbook is triggered when a new Incident is created over an analytic rule that indicates a URL with all the entities supported by Sentinel and provide options to SOC with different actions.

Action flow:

1.  **When an Azure Sentinel Incident is triggered**
2.  **Entities – Get URL**
3.  **Get ‘Security Policy Rule’ of this URL.**
4. Icon

   Description automatically generated **Post a choice of options as the flow bot** (Teams adaptive card) on Teams channel about the new incident, containing:
   1. Informative summary of the incident, ( incident description, Incident Severity, alert product names, Incident URL dynamic fields)
   2. Incident configuration:
      1. Incident Severity choices (Low, Medium, High)
      2. Incident Status options (Close the incident with choices of classification reasons)
   3. Options choices for SOC:
      1. Close incident
      2. Update severity
5. ** If URL was found** in the policy rule
   * 1. Icon

        Description automatically generated **In Teams adaptive card,** provide additional Options choices for SOC:
        + 1. Delete URL from Security Policy or option to skip this step.
     2.  **Delete URL** from the Security Policy by SOC choice.
     3.  **Change incident status** by user choice.
     4.  **Change incident severity** by user choice.
     5.  **Add a comment to the incident** with summary of action been taken.
6. **If URL is not found** in the policy rule
   * 1.  Add URL to the Security Policy by SOC choice.
     2.  **Change incident status** to close.
     3.  **Add a comment to the incident** with summary of action been taken.

# Scenario 5 – Block user and IP address

Sentinel Playbook is triggered when a new Incident is created over an analytic rule that indicates a IP with all the entities supported by Sentinel and enrich the incident with information of that IP.

Action flow:

1.  **When an Azure Sentinel Incident is triggered**
2.  **Entities – Get User Name**
3. **Get IP address based on User Name**



1.  **Get ‘Security Policy Rule’ of this IP**
2.  **Verify is IP is available**
3. **If IP is not found** in the policy rule
4.  Add IP to the Security Policy
5.  **Change incident status** to Close.
6.  **Add a comment to the incident** with summary of action been taken.

# Connector spec

Actions:

Based on them these are the actions that seems to make it work:

|  |  |  |
| --- | --- | --- |
| Title | Description | API reference |
| Create an Ip element object | Creates a new element object with the relevant IP address list.  Input:  Name: string  Comment (string – optional) | Create Ip name to add ip address.  Post:  <http://locahost:8082/version/elements/ip_list> |
| Create an Ip Address list based on element object | Creates a new Ip address list with in the element name.  Input:  ip: array | Creates a new Ip address list  Post:  [http://localhost:8082/v01/elements/ip\_list/{elementkey}/ip\_address\_list](http://localhost:8082/v01/elements/ip_list/%7belementkey%7d/ip_address_list) |
| Create an Uri element object | Creates a new element object with the relevant IP address list.  Input:  Name: string,  url\_entry:array  Comment (string – optional) | Creates a new Uri list  Post:  <http://localhost:8082/version/elements/url_list_application> |
| Create an host element object | Creates a new element object with the relevant host.  Input:  Name: string,  Address: Ip address  Secondary: Ip address(optional)  Comment (string – optional) | Create an host element object  Post:  <http://localhost:8082/version/elements/host> |
| Unlock the policy rule | Unlock the policy if policy is blocked by superuser: | Unlock the policy  Post:  [http://localhost:8082/version/elements/{policy\_type}/{element\_key}/force\_unlock](http://localhost:8082/version/elements/%7bpolicy_type%7d/%7belement_key%7d/force_unlock) |
| Reference an Address Object in a Security Policy Rule | Add a reference address object to existing Security Policy Rule.  Input:  Allow: string,  Service:tcp service, url element key,  Source: ip address list, host  Destination: ip address list, host | Reference an Element Object in the rule  Post:  [http://localhost:8082/version/elements/{policy\_type}/{element\_key}/fw\_ipv4\_access\_rule](http://localhost:8082/version/elements/%7bpolicy_type%7d/%7belement_key%7d/fw_ipv4_access_rule) |
| Get Security Policy rule | Get Security policy to modify or delete actions | Get:  [http://localhost:8082/version/elements/{policy\_type}/{element\_key}/search\_rule?filter=123&expose\_etag=true](http://localhost:8082/version/elements/%7bpolicy_type%7d/%7belement_key%7d/search_rule?filter=123&expose_etag=true) |
| Un-Reference an Element Object in a Security Policy Rule | Remove a reference of an Address Object from a Security Policy.  Input  Etag: string | Reference an Element Object in the rule  Delete:  [http://localhost:8082/version/elements/{policy\_type}/{element\_key}/fw\_ipv4\_access\_rule](http://localhost:8082/version/elements/%7bpolicy_type%7d/%7belement_key%7d/fw_ipv4_access_rule) |
| Upload/Update the Policy rule | After we add / remove the IP/URL into policy rule then we must upload policy rule again. Then only rule will applicable.  Input:  Filter: string(optional) | Post:  <http://forcepointsmc.eastus.cloudapp.azure.com:8082/6.9/elements/fw_policy/18/upload?filter=name> of the policy |
| Get Element Objects (IP) | Retrieve a specific address objects by name.  Input:  Address object identifiers | Get Element (IP) Objects  <http://localhost:8082/version/elements/host?filter=ip> |
| Get an Address(Url) Object | Retrieve a specific address objects by name.  Input:  url object identifiers | Get Element(Uri) Objects  <http://forcepointsmc.eastus.cloudapp.azure.com:8082/6.9/elements/url_list_application?filter=domain>  http://localhost/version/elements/url\_list\_application/{element\_key} |
| Delete Element Object(IP) | Delete Ip address  Input:  Etag:sting | Delete IP Address  [http://localhost:8082/version/elements/host/{eletment\_key}](http://localhost:8082/version/elements/host/%7beletment_key%7d) |
| Delete Element Object(Url) | Delete Url address  Input:  Etag:sting | Delete Url Address  [http://localhost:8082/version/elements/ url\_list\_application /{eletment\_key}](http://localhost:8082/version/elements/ur_li/%7beletment_key%7d) |
| Delete Element Object(host) | Delete Host address  Input:  Etag:sting | Delete Host Address  [http://localhost:8082/version/elements/host/{eletment\_key}](http://localhost:8082/version/elements/host/%7beletment_key%7d) |
| Login | Login into session  Input:  Authentication:apikey | Login in  Post:  <http://localhost:8082/version/login> |
| Logout | Logout from session | Logout  <http://localhost:8082/version/logout> |
| Get the Ip Address by username | we will get IP address based on usename  to add into security policy rule  Input  Ntlm-identity:Domain\\Username | Get  <https://localhost:5000/api/uid/v1.0/user/ntlm-identity/DEMO%5Cuser4> |