AppSentinels.ai
Application Security Re-invented

AppSentinels API Security Platform

Azure APIM Integration

Revision	Date Modified	Author	Comments
1.0	09-Sep-24	Arun	Initial spec for Azure APIM Integration
1.1	19-Oct-24	Arun	Update deployment steps using Powershell script
1.2	16-Mar-25	Arun	Policy fragment for inbound section in transparent mode policy
1.3	24-April-26	sagar	Updated new named value

Contents

AppSentinels sensor for Azure API Management service

This document covers the details of AppSentinels sensor for Azure API Management (APIM) service. Azure APIM uses policies to intercept HTTP requests and responses. It provides mechanisms to capture the data from headers and body of HTTP requests and responses. It also provides constructs to send that data to external endpoint. AppSentinels sensor for Azure APIM leverages this policy-based mechanism to capture required data and send that to AppSentinels Edge Controller. Similarly, AppSentinels uses this capability to enforce blocking policy on threat actor traffic. The policy is defined in XML format and uses the constructs provided by Azure APIM. It can be deployed manually in the Azure portal or can be deployed using a PowerShell script.

Policy modes

AppSentinels sensor policy can be deployed in following modes:

OOB mode or transparent mode

AppSentinels sensor captures the request and response headers and body and forwards the captured data to AppSentinels Edge Controller asynchronously. It does not enforce any blocking policy. The transparent mode policy is defined in *transparent-policy.xml* file.

Inline mode or auth mode

AppSentinels sensor can enforce threat actor policy actions in inline mode. It captures the request and forwards the request data to AppSentinels Edge Controller, which responds with the policy action on the API request. Depending on the action received from Controller, the sensor allows or blocks the request. If the request is allowed, it forwards the response data to the Controller. The auth mode policy is defined in *auth-policy.xml* file.

Configuration

AppSentinels sensor APIM policy configuration is defined under ***APIM Service > APIs > Named values*** configuration in APIM service configuration. Following table describes different parameters. It is important to note that many of these parameters are optional and are initialized with default values. It is mandatory to configure all below variable in ***Named values*** section

Name	Description	Default
apim-name	Name of the Azure APIM service, this is sent as part of sensor logs to controller.	
api-version	API version, this is also sent as part of logs.	1
appsentinels-endpoint	AppSentinels Controller endpoint URL, specified as http://<Controller URL>:<port number>	

Name	Description	Default
authz-api	This defines the URI to be used to send the request data to Controller in auth mode.	auth
response-api	This defines the URI to be used to send the response data to Controller in auth mode.	nginxlog
authz-timeout	This duration defines the timeout in seconds for authz request sent to the AppSentinels Edge Controller.	1
logging-timeout	This duration (in seconds) is used in auth mode as well as in transparent mode. • In auth mode, it defines timeout for sending response data to controller. • In transparent mode, it is used as timeout for sending the log message to AppSentinels Edge Controller.	1
max-payload-size	This defines the maximum size of the request body, or the response body captured in the log. If the size of request/response body exceeds this value, the size of the body included in the log message is truncated to zero bytes.	131072
sensor-host-name	This value is sent as a header in the logs to Controller. This is used for sensor visibility. It should be modified while configuring APIM policy. The default value of this parameter is azure-apim.	azure-apim
sensor-instance-name	This value is sent as a header in the logs to Controller. Like sensor-host-name, this is also used for sensor visibility. By default, the APIM service name is used as the sensor-instance-name, but this can be modified in the Named values configuration based on the requirement.	{APIM SERVICE NAME}
skip-chunked-body	This configuration is used to skip to read the chunked data, default is true, if you want to read the chunked data configure as false	true

The parameters *appsentinels-endpoint* and *apim-name* are mandatory parameters which are needed while deploying the AppSentinels sensor policy using PowerShell script.

AppSentinels APIM policy deployment

AppSentinels APIM policy can be deployed manually or using a PowerShell script. If APIM service already has some existing policy, then it is recommended to manually merge the AppSentinels XML policy with the existing policy. It is important to note that AppSentinels policy can be deployed at global, product, API or operation level and is enforced on the APIs in that scope.

PowerShell script

PowerShell script *deploy_apim_policy.ps1* can be used to deploy AppSentinels APIM policy. Please note that this script should not be used if APIM service has an existing policy configured. Following are the parameters required to run the script:

- Azure subscription id
- Azure resource group name
- Azure APIM service name
- Scope type: Level at which policy should be deployed, possible values are:
  - global – policies applied to all APIs
  - product – policies applied to APIs part of the product
  - api – policies applied to all operation of the given API
  - operation – policies applied to specific operation of an API
- Scope Id: It extends the scope type by naming the specific object which are part of the scope of the policy.
  - If scope type is global, since all APIs are in the scope, empty value ('') is used as scope id.
  - product ID or product name for product scope.
  - API ID or API name for api scope – please note that the powershell script does not accept API display name having 'space', in such a case API name should be used.
  - Use 'API:operation ID' as scope id for operation scope.
- Controller endpoint: AppSentinels controller endpoint URL in *http://<Controller URL>:<Port number>* format.
- Sensor mode: It is used to identify the type of policy to be deployed.
  - *auth* for auth mode or inline mode, and
  - *transparent* for transparent or OOB mode

This script adds the parameters with default values in Named values configuration, adds the required policy fragment and adds the policy XML in API configuration.

Script usage

DEPLOY_APIM_POLICY.PS1 <SUBSCRIPTION-ID> <RESOURCE-GROUP> <APIM-SERVICE-NAME> <SCOPE-TYPE> <SCOPE-ID> <CONTROLLER-ENDPOINT> <SENSOR-MODE>

Please find below an example of deploying AppSentinels APIM *transparent* mode policy for *all operations* of *appsentinels-httpserver-app* API in APIM service *appsentinels-apim-service* which exists in *appsentinels-apim-rg* resource group in subscription *072ddb92-e77f-44c1-*

b5d1-c6cc1cd8b115. The command specifies http://testcontroller.appsentinels.ai:9004 as controller endpoint.

```
./deploy_apim_policy.ps1
        072ddb92-e77f-44c1-b5d1-c6cc1cd8b115 \
        appsentinels-apim-rg \
        appsentinels-apim-service \
        api \
        appsentinels-httpserver-app \
        http://testcontroller.appsentinels.ai:9004 \
        transparent
```

This command adds all the parameters used in the AppSentinels policy in ***APIM Service > APIs > Named values*** configuration and copies the policy XML data in APIM service. This can be verified in APIM service configuration in Azure port.

It is important to note that this command overwrites the *Names values* and policy XML configuration, that's the reason it should not be used if APIM service has an existing policy.

Manual deployment

This method should be used if APIM service has an existing policy. It can also be used if APIM service has no (or default) policy. Please follow these steps to deploy the AppSentinels sensor policy:

- Create the required policy fragments in **APIM service > APIs > Policy fragments**.
    - Auth mode policy
        - Specify the fragment **Name** as *appsentinels-auth-mode-fragment*.
        - Copy the contents of file *auth-policy-fragment.xml* in **XML policy fragment** box.
        - Click **Create**.
    - Transparent mode policy
        - Create a fragment for inbound policy with **Name** as *appsentinels-transparent-mode-fragment-inbound*.
        - Copy the contents of file *transparent-policy-fragment-inbound.xml* in **XML policy fragment** box.
        - Click **Create**.
        - Create a fragment for outbound policy with **Name** as *appsentinels-transparent-mode-fragment*.
        - Copy the contents of file *transparent-policy-fragment.xml* in **XML policy fragment** box.
        - Click **Create**.
- Take a backup of your current policy XML.
- Open your existing policy XML in Azure portal.
- Open the required AppSentinels policy XML - *transparent-policy.xml* for transparent mode policy and *auth-policy.xml* for auth mode policy.
- Copy the contents enclosed by and from the AppSentinels policy XML file to *inbound* section in APIM policy XML in Azure portal.

- Copy the contents enclosed by and from the AppSentinels policy XML file to *backend* section in APIM policy XML in Azure portal.
- Copy the contents enclosed by and from the AppSentinels policy XML file to *outbound* section in APIM policy XML in Azure portal.
- Remove duplicate lines if there are any in inbound, backend or outbound sections.
- Add the named values in ***APIM service > APIs > Named values*** configuration, one by one. Please note that all the parameters are listed in Configuration section in this document.

Please contact AppSentinels support if merged policy does not work.

