radware

AZURE SENTINEL IMPLEMENTATION GUIDE

Contents

Overview

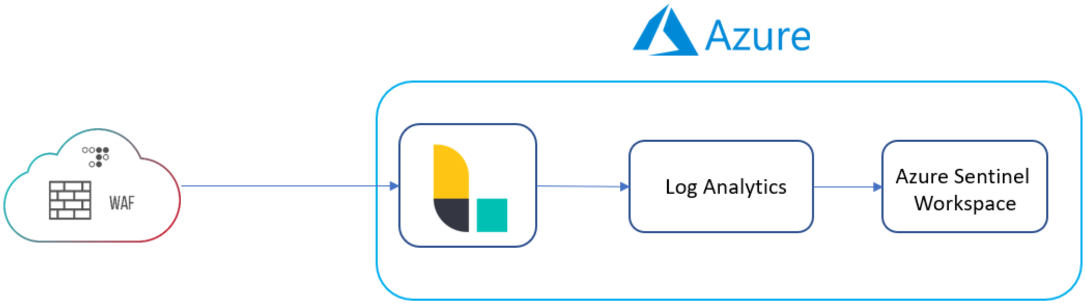
A security event is generated whenever Radware Cloud Services detect an attack when an ongoing attack is still active or when an ongoing attack status has changed. The generated security event includes the information relevant to this specific attack or security breach. Once an event has been created, it is reported to Radware Cloud management (portal), and also optionally distributed to the customer's on-premise/cloud logging system (SIEM).

Each tenant owns a queue for attacks as detected by Cloud WAF. The queues are fully isolated, with permissions only for the tenant that owns the queue. When a tenant is registered for SIEM services, the events that are generated by Radware Cloud Services are sent to their queues.

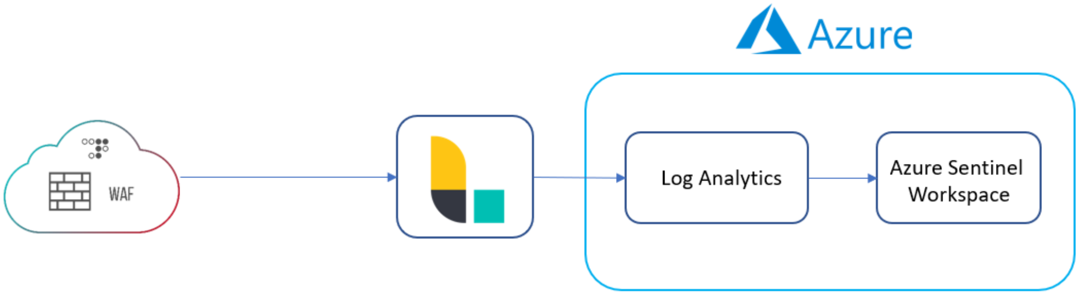
This document describes the deployment of a LogStash system to consume events from Radware CloudWAF and forward them to Azure Sentinel for analytics.

Solution

To connect Radware CloudWAF to Azure Sentinel, a LogStash instance needs to be deployed to consume logs, process, and forward them to a Log Analytics workspace.



Alternatively, the logstash instance can be deployed on premise or in any other location with access to Azure services and Radware Cloud WAF.



Pre-Setup

Logstash

Logstash requires one of these versions:

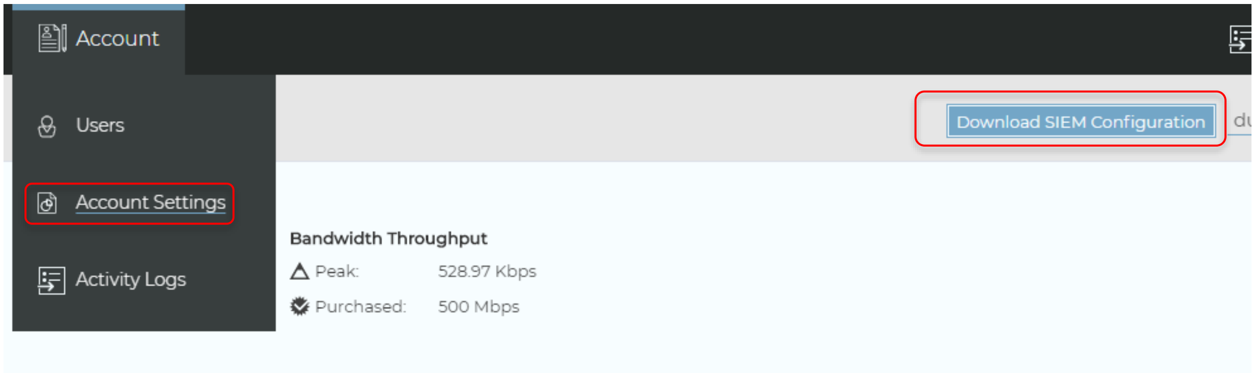
- Java 8
- Java 11
- Java 14

Installation instructions for Logstash can be found here: https://www.elastic.co/guide/en/logstash/current/installing-logstash.html

Configuration File

From the Radware Cloud WAF portal, download the SIEM configuration file.

Navigate to Account -> Account Settings -> Download SIEM Configuration



Bandwidth Throughput

Peak:	528.97 Kbps
Purchased:	500 Mbps

Azure Log Analytics output plugin

The Azure Log Analytics plugin needs to be installed for logstash to forward events to Azure.

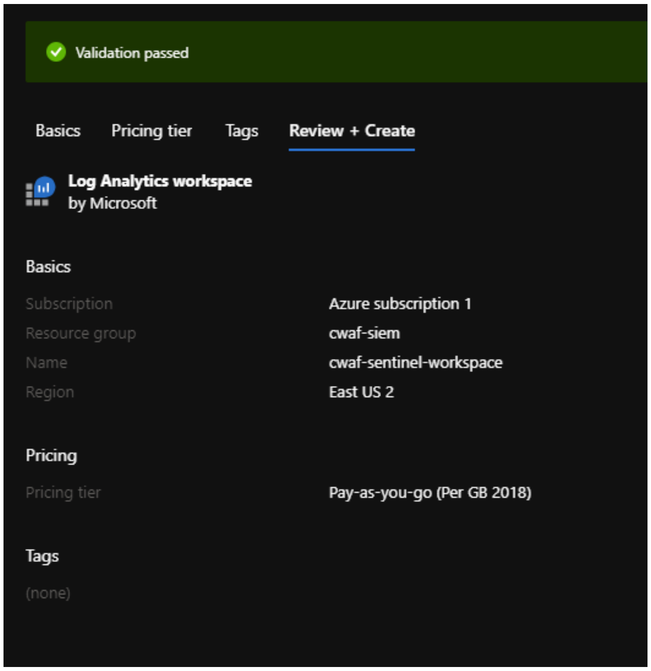
You can install this plugin using the Logstash "plugin" or "logstash-plugin" (for newer versions of Logstash) command:

```
bin/plugin install logstash-output-azure_loganalytics
# or
bin/logstash-plugin install logstash-output-azure_loganalytics  (Newer versions of
Logstash)
```

For more information, refer to https://github.com/yokawasa/logstash-output-azure_loganalytics

Azure Log Analytics workspace

An Azure Log Analytics workspace needs to be created to ingest events forwarded by logstash



Validation passed

Log Analytics workspace
by Microsoft

Basics

Subscription Azure subscription 1
Resource group cwaf-siem
Name cwaf-sentinel-workspace
Region East US 2

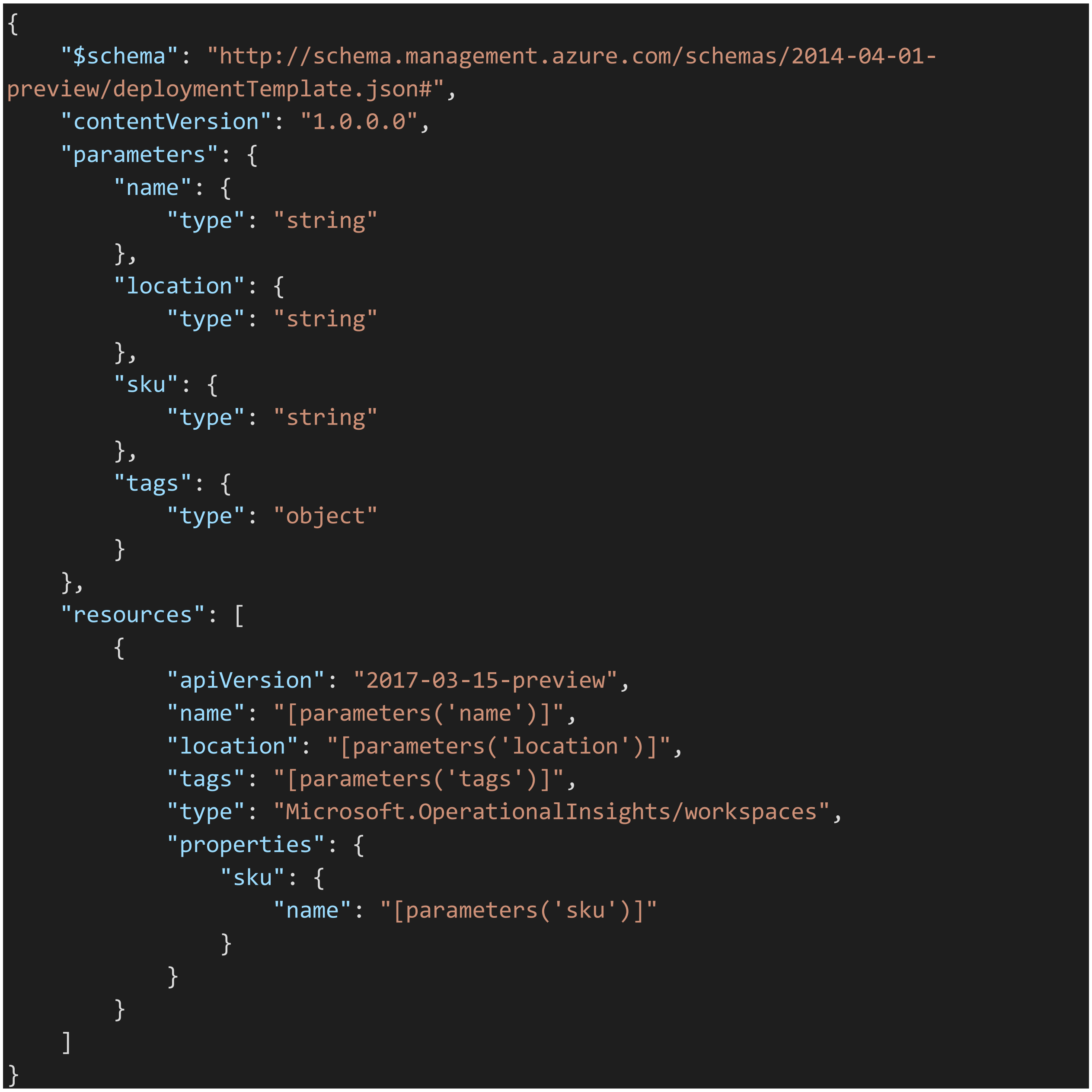
Pricing

Pricing tier Pay-as-you-go (Per GB 2018)

Tags

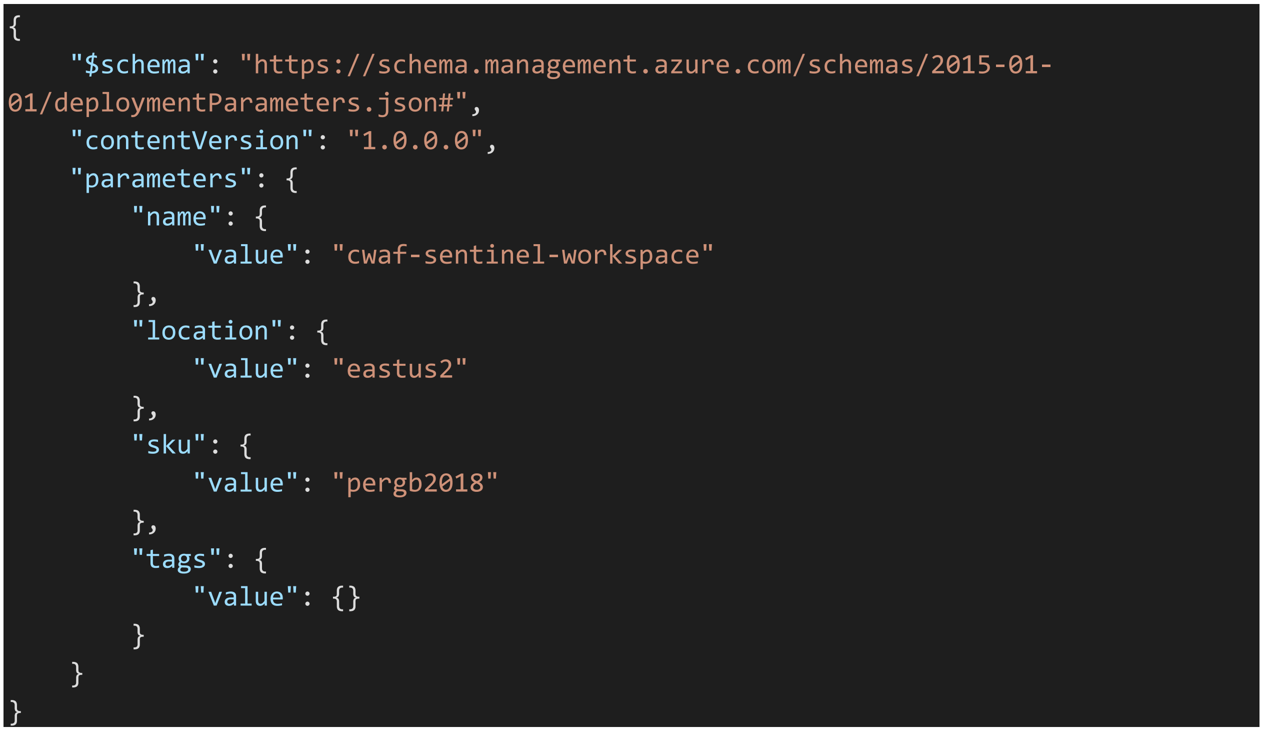
(none)

Azure Templates for Log Analytics Workspace:



```
{
    "$schema": "http://schema.management.azure.com/schemas/2014-04-01-
preview/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "name": {
            "type": "string"
        },
        "location": {
            "type": "string"
        },
        "sku": {
            "type": "string"
        },
        "tags": {
            "type": "object"
        }
    },
    "resources": [
        {
            "apiVersion": "2017-03-15-preview",
            "name": "[parameters('name')]",
            "location": "[parameters('location')]",
            "tags": "[parameters('tags')]",
            "type": "Microsoft.OperationalInsights/workspaces",
            "properties": {
                "sku": {
                    "name": "[parameters('sku')]"
                }
            }
        }
    ]
}
```

Template.json



```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "name": {
            "value": "cwaf-sentinel-workspace"
        },
        "location": {
            "value": "eastus2"
        },
        "sku": {
            "value": "pergb2018"
        },
        "tags": {
            "value": {}
        }
    }
}
```

Parameters.json

Azure Sentinel Workspace

Azure Sentinel uses a Log Analytics workspace for data. When creating an Azure Sentinel Workspace, use the Log Analytics Workspace created previously.

Setup

Logstash

Any WAF message sent from Radware Cloud WAF Services starts with the following prefix: AppWallAttackSysLogMessage.

Each parameter is a key-value pair, where an equal sign (=) separates the key and the value, and the key-value pairs are separated by tab separators or new lines.

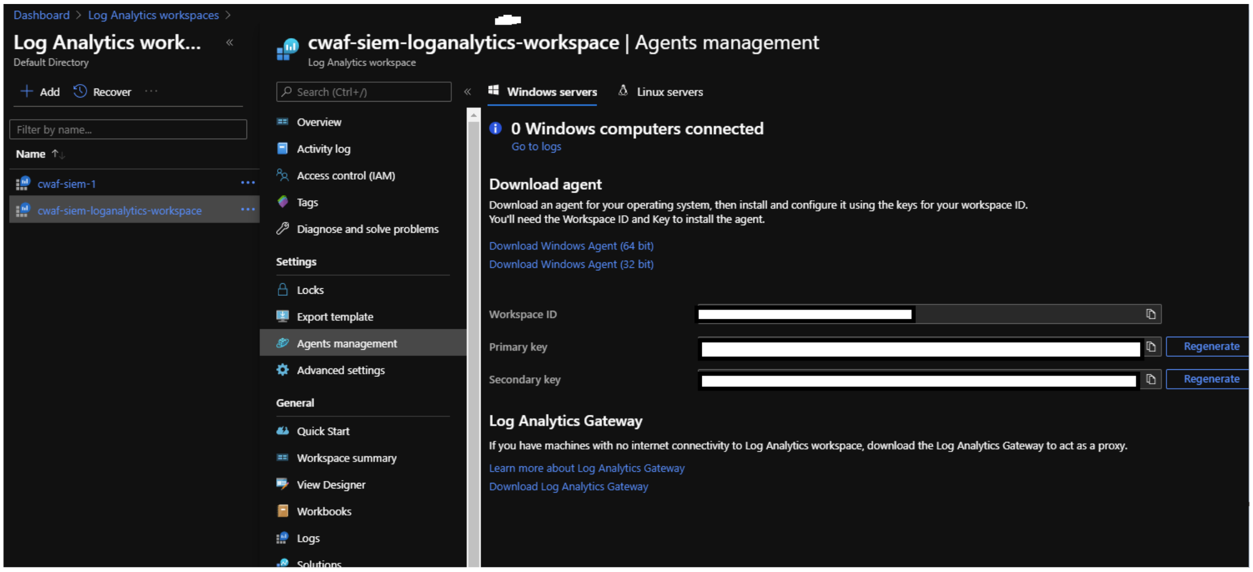
The logstash configuration file downloaded from the portal will be in the following format:

```
input{
        sqs{
        queue=> <AWS queue name, enclosed in “ “ >
   access_key_id=>"KEY "
      region=> <AWS region, enclosed between “ “ >
      secret_access_key=> <secret access key, enclosed in “ “ >
}
}
output{
        udp {
  host=> <IP address of SIEM system, enclosed in “ “ >
      port=> <destination port of SIEM system, enclosed in “ “ >
}
}
```

The ‘input’ block contains information needed by logstash to get events from the radware CWAF queues.
The ‘output’ block contains the information logstash will use to send events to.

To send events to Azure log analytics, output will need to be modified and a filtering mechanism will be added for events to be sent in the appropriate format.
In addition, we would need the workspace ID and key from Azure log analytics.

To get this info, on the azure portal, navigate to
Log Analytics Workspace -> Select the appropriate workspace -> Agents Management



cwaf-siem-loganalytics-workspace | Agents management
Log Analytics workspace

0 Windows computers connected

Download agent

Download an agent for your operating system, then install and configure it using the keys for your workspace ID.
You'll need the Workspace ID and Key to install the agent.

Download Windows Agent (64 bit)
Download Windows Agent (32 bit)

Workspace ID [redacted]

Primary key [redacted]

Secondary key [redacted]

Log Analytics Gateway

If you have machines with no internet connectivity to Log Analytics workspace, download the Log Analytics Gateway to act as a proxy.

Learn more about Log Analytics Gateway
Download Log Analytics Gateway

Using the installed azure log analytics plugin, a sample ‘output’ block would be:

```
output {
   azure_loganalytics {
      customer_id => "<WORKSPACE ID>"
      shared_key => "KEY"
      log_type => "<Alphabetical Name>"
   }
}
```

A filter needs to be added to parse the events’ key-value pairs using:

```
filter {
      kv { }
}
```

Sample logstash config file:

```
input {
   sqs {
      queue => "<QUEUE ID>"
      region => "region"
      secret_access_key => "<QUEUE ACCESS KEY>"
      codec => "plain"
   }
   sqs {
      queue => "<QUEUE ID>"
      access_key_id => "<QUEUE ACCESS KEY>"
      region => "region"
      secret_access_key => "<QUEUE ACCESS KEY>"
      codec => "plain"
   }
}
filter {
      kv { }
      date {
            timezone => "UTC"
            match => ["receivedTimeStamp", "UNIX_MS"]
            target => "@rwtimestamp"
     }
}
output {
   azure_loganalytics {
      customer_id => "<WORKSPACE ID>"
      shared_key => “<KEY>”
      log_type => "RdwrCWAFLogs"
   }
}
```

To run logstash, use

```
sudo /usr/share/logstash/bin/logstash -f <logstash_config_file
```

GeoLocation

Modify the filter block to the following:

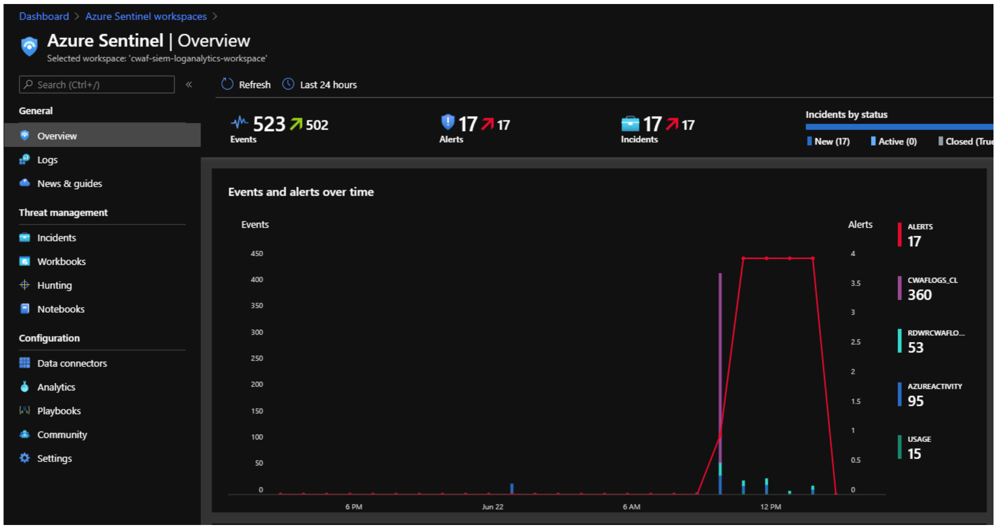
```
filter {
      kv { }
      geoip {
            database => “<Path to local MaxMind DB file>"
            source => "sourceIp"
      }
}
```

Logstash supports Maxmind GeoIlte2 databases. If a database is not specified in the filter, logstash uses a default GeoLite2-City database already part of the deployment.

Azure Sentinel

No specific configuration needs to be done on Azure to ingest data as long as the workspace ID and shared key are correct.

Once logstash starts processing and forwarding events, the Azure Sentinel overview dashboard will show data received.



Azure Sentinel Workbooks

Once data is being consumed by Azure Sentinel, you can visualize and monitor the data using the Azure Sentinel adoption of Azure Monitor Workbooks, which provides versatility in creating custom dashboards.

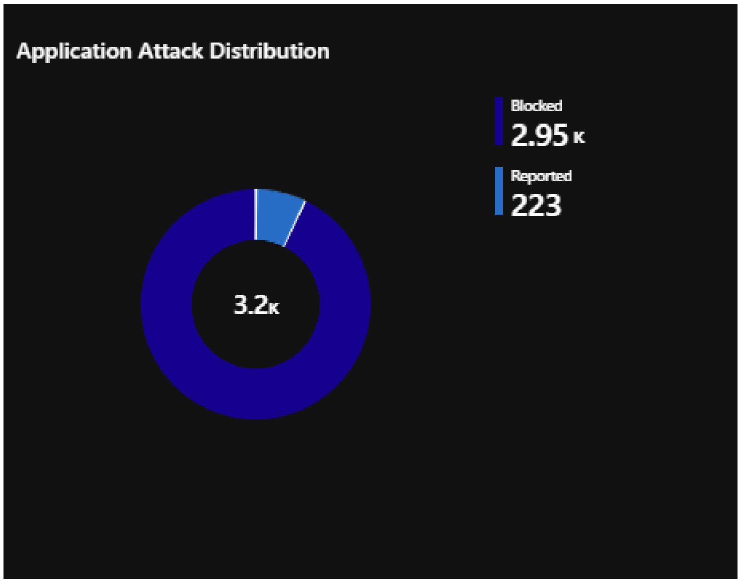
Refer to https://docs.microsoft.com/en-us/azure/sentinel/tutorial-monitor-your-data for instructions on creating custom workbooks.
Once a workbook is created, queries can be written to parse the logs and create visualizations.
Azure queries use the Kusto query language. More info and documentation can be found here: https://docs.microsoft.com/en-us/azure/data-explorer/kusto/concepts/

In the custom workbook, in editing mode, queries and visualizations can be added.

Examples:

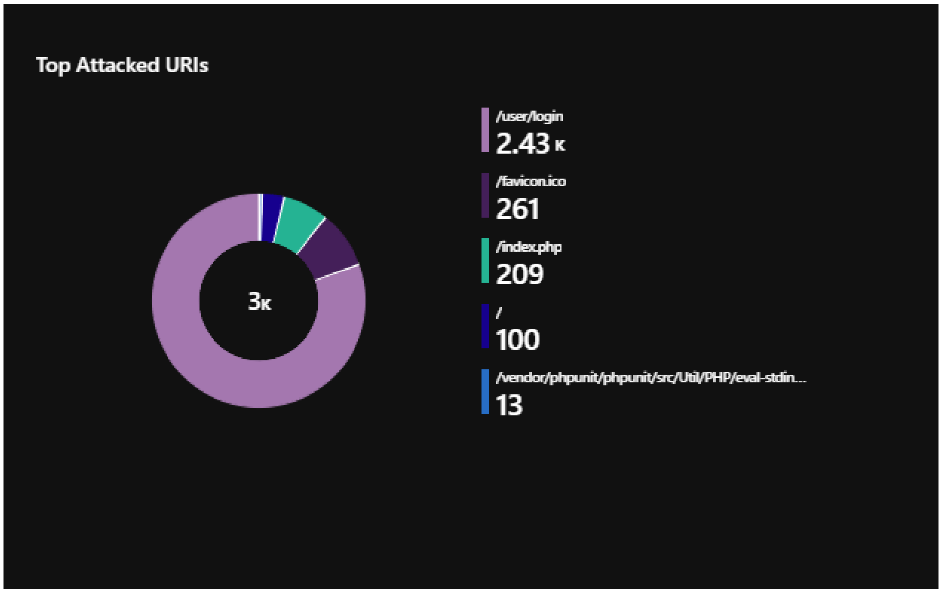
1. Application Attack Distribution
    a. Visualization: Pie Chart
    b. Query:

```
RdwrCWAFLogs_CL
| summarize count() by action_s
```



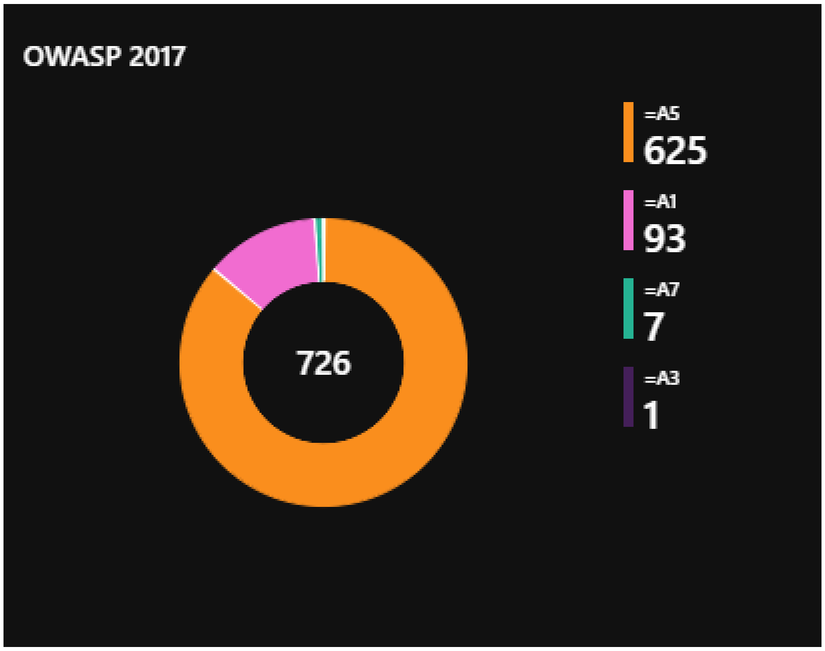
2. Top Attacked URIs
    a. Visualization: Pie Chart
    b. Query:

```
RdwrCWAFLogs_CL
| summarize count() by uri_s
| top-hitters 5 of uri_s by count_
```



3. OWASP 2017 Categorization:
    a. Visualization: Pie Chart
    b. Query:

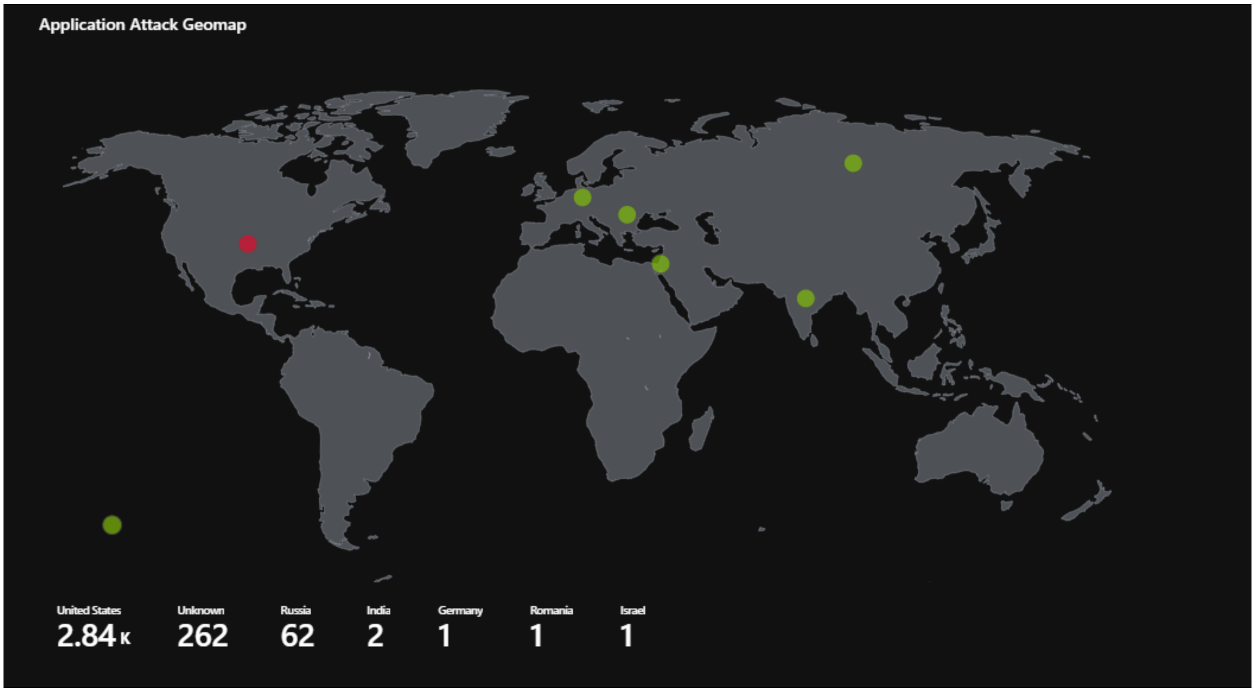
```
union RdwrCWAFLogs_CL
| where enrichmentContainer_s contains "owasp" and enrichmentContainer_s
!contains "owaspCategory2017=null"
| extend enrichFields = split(enrichmentContainer_s, 'owaspCategory2017')
| extend owaspField = tostring(enrichFields[1])
| extend owaspField2 = split(owaspField, ',')
| project owaspCategory = owaspField2[0]
| summarize count() by tostring(owaspCategory)
```



4. Application Attack Geomap:
   Note: Logstash must be configured to send geo information. Refer to the setup section for details.
   a. Visualization: Map
   b. Query:

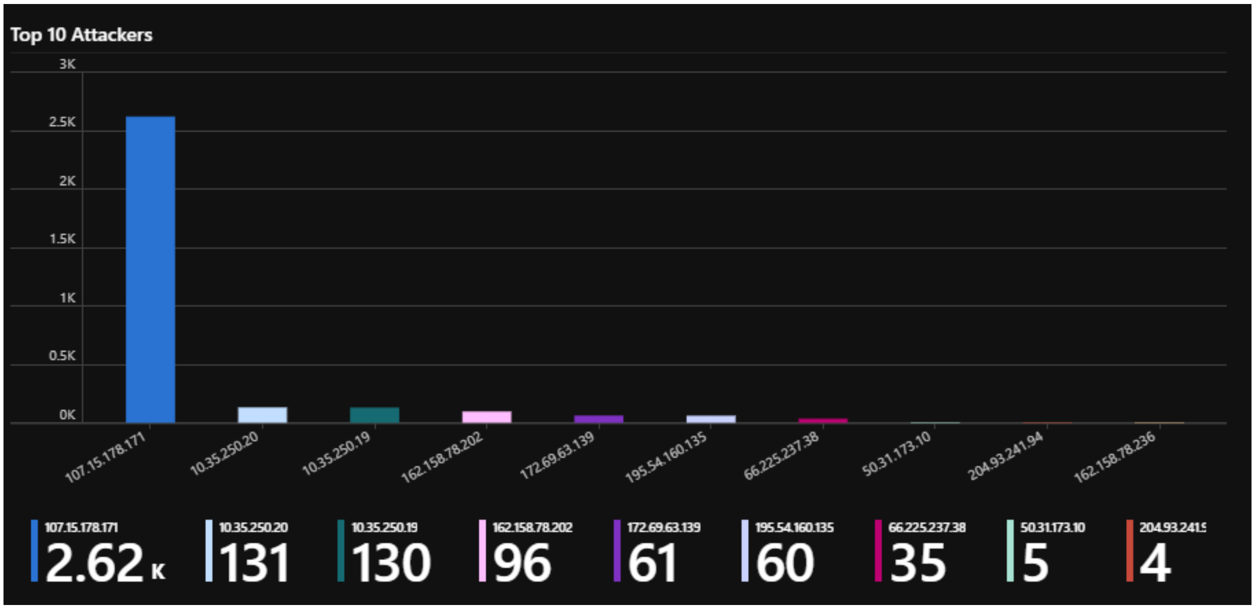
```
RdwrCWAFLogs_CL
| summarize count() by geoip_country_name_s
```

   c. Map Settings:
      - Location Info Using: Country or region
      - Country/region field: geoip_country_name_s
      - Size by: count_



5. Top 10 Attackers:
    a. Visualization: Bar Chart
    b. Query:

```
union RdwrCWAFLogs_CL
| summarize count() by SourceIP
| top-hitters 10 of SourceIP by count_
| order by approximate_sum_count_ desc
```



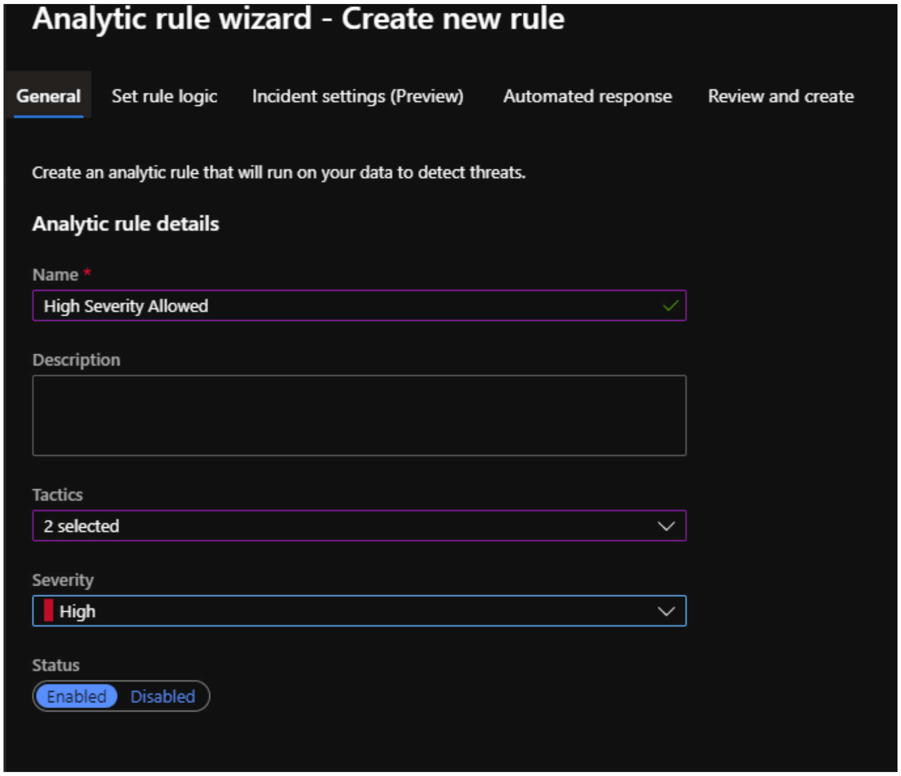
Incidents

After you connected your data sources to Azure Sentinel, you can create custom rules that can search for specific criteria across your environment and generate incidents when the criteria are matched so that you can investigate them.

For example, let's create a Analytics rule to create an incident when:

- Report Only alerts > 100 over the past 24 hours
- Run every 15 minutes and suspend rule for 5 hours on trigger
- Include Host, IP and URI for investigation

1. In the Azure portal under Azure Sentinel, select Analytics.
2. In the top menu bar, select +Create and select Scheduled query rule. This opens the Analytics rule wizard.
3. In the General tab, provide a unique Name, and a Description. In the Tactics field, you can choose from among categories of attacks by which to classify the rule. Set the alert Severity as necessary.



Analytic rule wizard - Create new rule

Create an analytic rule that will run on your data to detect threats.

Analytic rule details

Name * High Severity Allowed

Description

Tactics 2 selected

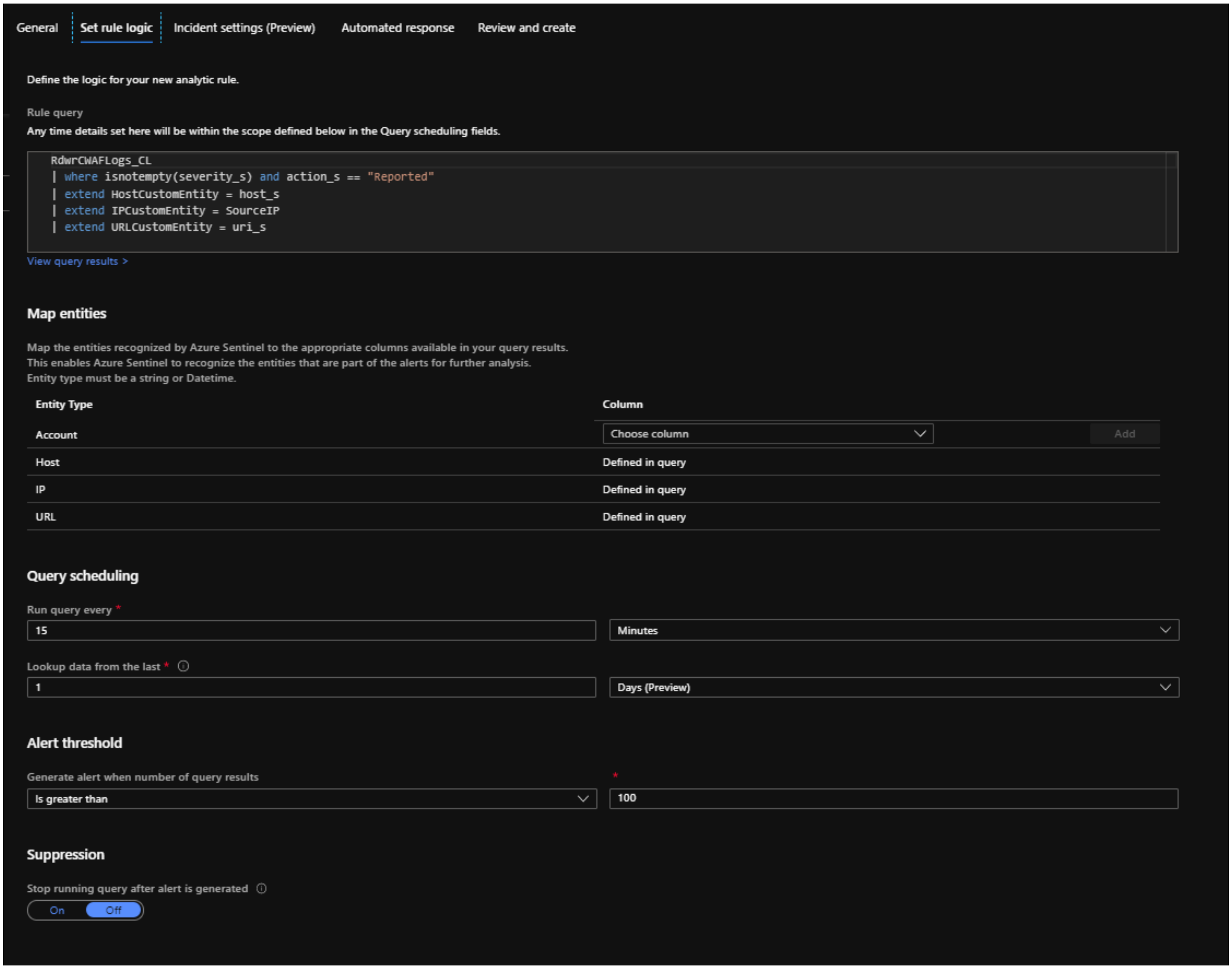
Severity High

Status Enabled Disabled

4. In the Set Rule logic tab, use the following query in the textbox:

```
RdwrCWAFLogs_CL
| where isnotempty(severity_s) and action_s == "Reported"
| extend HostCustomEntity = host_s
| extend IPCustomEntity = SourceIP
| extend URLCustomEntity = uri_s
```

Set other parameters as shown or as per need



Define the logic for your new analytic rule.

Rule query

Any time details set here will be within the scope defined below in the Query scheduling fields.

```
RdwrCWAFLogs_CL
| where isnotempty(severity_s) and action_s == "Reported"
| extend HostCustomEntity = host_s
| extend IPCustomEntity = SourceIP
| extend URLCustomEntity = uri_s
```

Map entities

Map the entities recognized by Azure Sentinel to the appropriate columns available in your query results.
This enables Azure Sentinel to recognize the entities that are part of the alerts for further analysis.
Entity type must be a string or Datetime.

Entity Type	Column
Account	Choose column
Host	Defined in query
IP	Defined in query
URL	Defined in query

Query scheduling

Run query every *: 15 Minutes

Lookup data from the last *: 1 Days (Preview)

Alert threshold

Generate alert when number of query results: Is greater than 100

Suppression

Stop running query after alert is generated: On [✓ Off]

5. In the Incident Settings tab, you can choose whether and how Azure Sentinel turns alerts into actionable incidents. If this tab is left alone, Azure Sentinel will create a single, separate incident from each and every alert. You can choose to have no incidents created, or to group several alerts into a single incident, by changing the settings in this tab.
6. In the **Automated responses** tab, select any playbooks you want to run automatically when an alert is generated by the custom rule. For more information on creating and automating playbooks, see Respond to threats.
7. Select Review and create to review all the settings for your new alert rule and then select Create to initialize your alert rule.
8. After the alert is created, a custom rule is added to the table under Active rules. From this list you can enable, disable, or delete each rule.
9. To view the results of the alert rules you create, go to the Incidents page, where you can triage, investigate incidents, and remediate the threats.

Playbooks

A security playbook is a collection of procedures that can be run from Azure Sentinel in response to an alert. A security playbook can help automate and orchestrate your response, and can be run manually or set to run automatically when specific alerts are triggered. Security playbooks in Azure Sentinel are based on Azure Logic Apps, which means that you get all the power, customizability, and built-in templates of Logic Apps.

An example is documented in the appendices.

APPENDIX

1. **Debugging event parsing:**

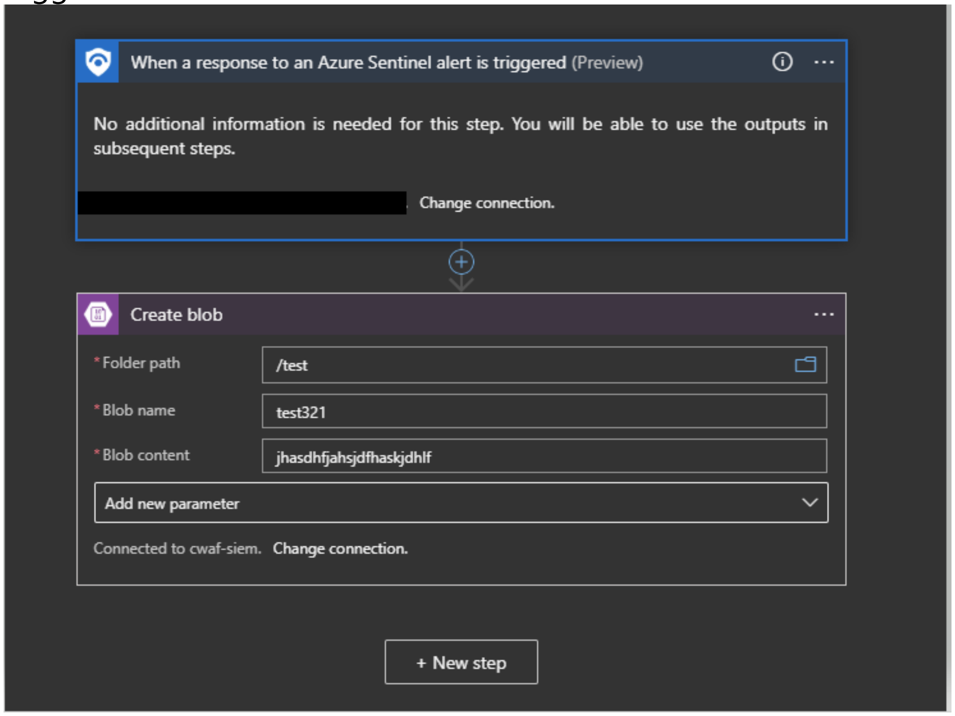
Include stdout to debug to check for any parsing failures in the output block.
For example, set the output block in the logstash config file to:

```
output {
   azure_loganalytics {
      customer_id => "<WORKSPACE ID>"
      shared_key => “<KEY>”
      log_type => "RdwrCWAFLogs"
   }
   stdout { codec => rubydebug }
}
```

This will dump all parsed events to the console

2. **Logic App Example:**

This example creates a blob in a storage account container when a response to an azure sentinel is triggered.



When a response to an Azure Sentinel alert is triggered (Preview)

No additional information is needed for this step. You will be able to use the outputs in subsequent steps.

[redacted] Change connection.

Create blob

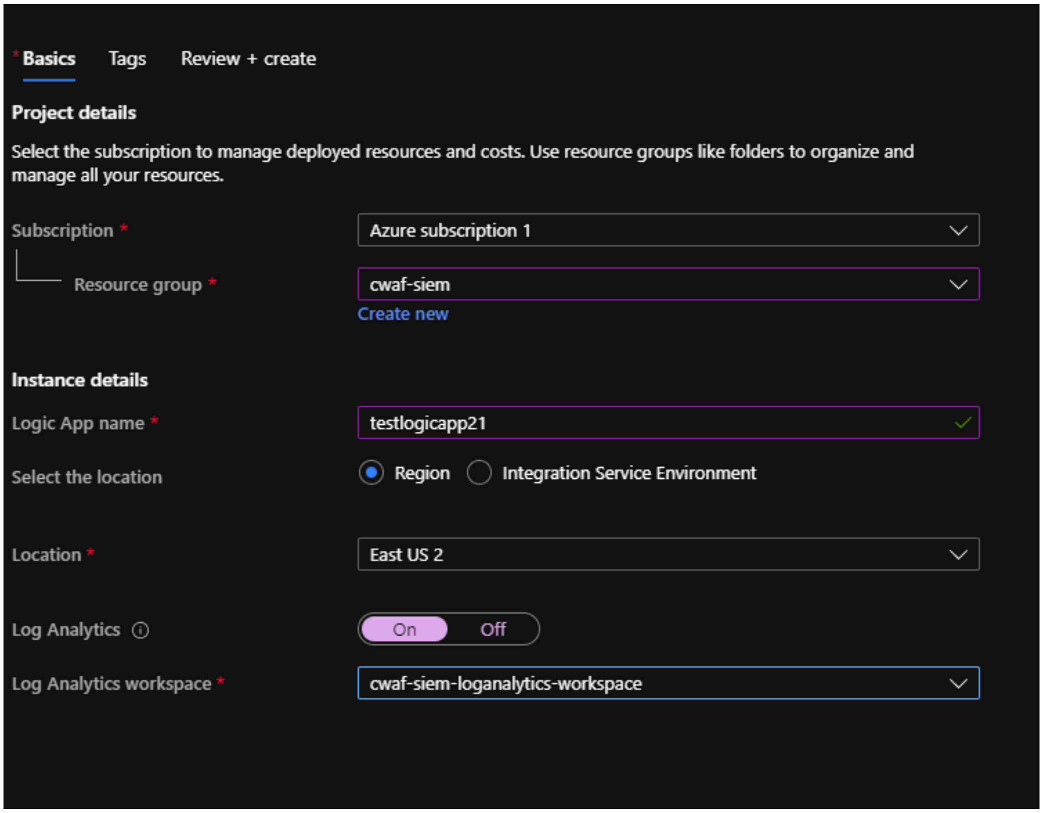
- *Folder path: /test
- *Blob name: test321
- *Blob content: jhasdhfjahsjdfhaskjdhlf

Add new parameter

Connected to cwaf-siem. Change connection.

Steps:

1. In the Azure Sentinel workspace, go to ***Playbooks*** and ***Add Playbook***
2. Set the parameters for a new logic app – ***Resource Group, Name, Location***



Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure subscription 1

Resource group * cwaf-siem

Create new

Instance details

Logic App name * testlogicapp21

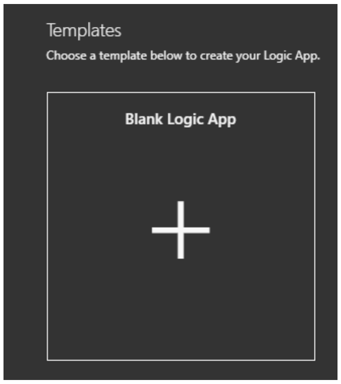
Select the location ● Region ○ Integration Service Environment

Location * East US 2

Log Analytics ⓘ On Off

Log Analytics workspace * cwaf-siem-loganalytics-workspace

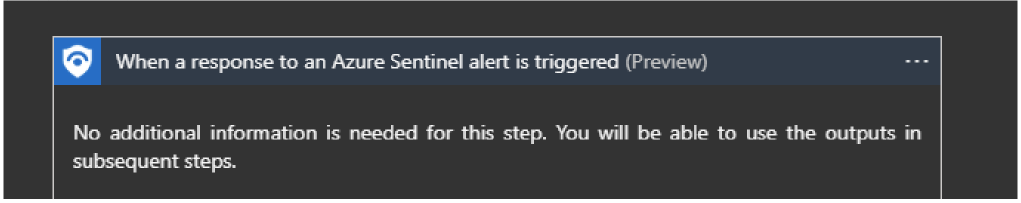
3. Add tags as required and click ***Create***
4. Once created, go to the Resource and click on ***Blank Logic App*** in the Logic Apps Designer



Templates

Choose a template below to create your Logic App.

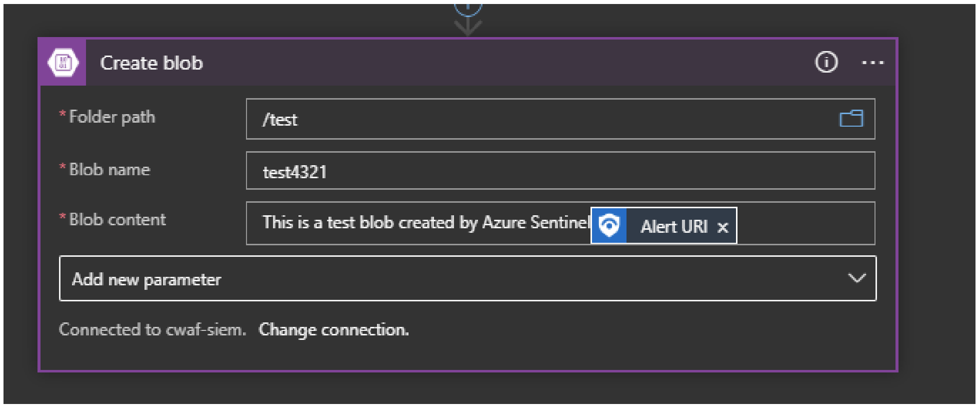
5. Search and Select ***Azure Sentinel***
6. As a trigger, select ***When a response to an Azure Sentinel alert is triggered***



When a response to an Azure Sentinel alert is triggered (Preview)

No additional information is needed for this step. You will be able to use the outputs in subsequent steps.

7. Click on **+** ***New Step*** and search and select ***Azure Blob Storage***
8. Set action to ***Create blob***



Create blob

*Folder path: /test

*Blob name: test4321

*Blob content: This is a test blob created by Azure Sentinel Alert URI ×

Add new parameter

Connected to cwaf-siem. Change connection.

9. Click on Save
10. This app is now available as a playbook in Azure Sentinel Analytics Rules

Sample Log Processing:

Sample message:

```
"- [AppWallAttackSysLogMessage devType=\"null\" destinationPort=\"54001\"
request=\"GET /favicon.ico HTTP/1.1\r\nHost:HOST_FOR_HC\r\nX-RDWR-HC: health
check\r\n\r\n\" sourcePort=\"64054\" role=\"public\" awVersion=\"null\"
transId=\"2567684990\" paramName=\"null\" title=\"Forbidden Request\"
appPath=\"/favicon.ico\" directory=\"/\" destinationIp=\"10.35.101.156\"
paramType=\"null\" protocol=\"HTTP\" refineCRC=\"null\" security=\"true\"
violationType=\"Folder Access Violation\" host=\"HOST_FOR_HC\" action=\"Blocked\"
details=\"PathBlocking Security Filter intercepted a malicious request. Users are
forbidden to access the requested path.\n\nNo Src Page: might be manual hacking
attempt !\nAuthenticated as Public\n\" receivedTimeStamp=\"1594870692471\"
ruleId=\"null\" paramValue=\"null\" severity=\"High\" violationCategory=\"Access
Control\" webApp=\"DemoSite-Secure\" targetModule=\"PathBlocking\"
method=\"GET\" module=\"PathBlocking\" refine=\"null\"
enrichmentContainer=\"{owaspCategory=A7, geoLocation={countryCode=--},
owaspCategory2017=A5, contractId=63bce674-e83d-4fae-909d-84b309ba0cd9,
applicationId=6452a09b-4d58-42fd-af72-593091bca6eb, tenant=75292c67-8443-4714-
babe-851b29de7cab}\" uri=\"/favicon.ico\" passive=\"false\" vhost=\"&lt;any host&gt;\"
sourceIp=\"10.35.250.20\" appWallTimeStamp=\"1594870518662\" externalIp=\"null\"
user=\"public\" tunnel=\"ChicagoDemoSiteSecure\"]"
```

Processed message into relevant fields:

```
{
             "paramName" => "null",
                "ruleId" => "null",
                   "uri" => "/favicon.ico",
                "tunnel" => "\"ChicagoDemoSiteSecure\"]",
            "sourcePort" => "64054",
                "action" => "Blocked",
            "paramValue" => "null",
                "method" => "GET",
                "webApp" => "DemoSite-Secure",
                "passive" => "false",
                  "geoip" => {},
                   "host" => "HOST_FOR_HC",
    "enrichmentContainer" => "owaspCategory=A7 geoLocation=countryCode=-- owaspCategory2017=A5 contractId=63bce674-e83d-4fae-909d-84b309ba0cd9 applicationId=6452a09b-4d58-42fd-af72-593091bca6eb tenant=75292c67-8443-4714-babe-851b29de7cab",
                 "module" => "PathBlocking",
      "violationCategory" => "Access Control",
           "targetModule" => "PathBlocking",
               "@version" => "1",
                  "vhost" => "&lt;any host&gt;",
                 "refine" => "null",
             "@timestamp" => 2020-07-21T19:24:39.881Z,
                "transId" => "2567684990",
               "protocol" => "HTTP",
               "sourceIp" => "10.35.250.20",
       "appWallTimeStamp" => "1594870518662",
              "refineCRC" => "null",
              "awVersion" => "null",
      "receivedTimeStamp" => "1594870692471",
                "appPath" => "/favicon.ico",
              "directory" => "/",
                  "title" => "Forbidden Request",
             "externalIp" => "null",
                "request" => "GET /favicon.ico HTTP/1.1\r\nHost:HOST_FOR_HC\r\nX-RDWR-HC: health check\r\n\r\n",
        "destinationPort" => "54001",
               "security" => "true",
                "devType" => "null",
          "destinationIp" => "10.35.101.156",
               "severity" => "High",
           "@rwtimestamp" => 2020-07-16T03:38:12.471Z,
                "details" => "PathBlocking Security Filter intercepted a malicious request. Users are forbidden to access the requested path.\n\nNo Src Page: might be manual hacking attempt !\nAuthenticated as Public\n",
                   "user" => "public",
           "role" => "public",
      "violationType" => "Folder Access Violation",
        "paramType" => "null",
           "tags" => []
}
```