Zscaler Internet Access (ZIA) Functions for IBM SOAR

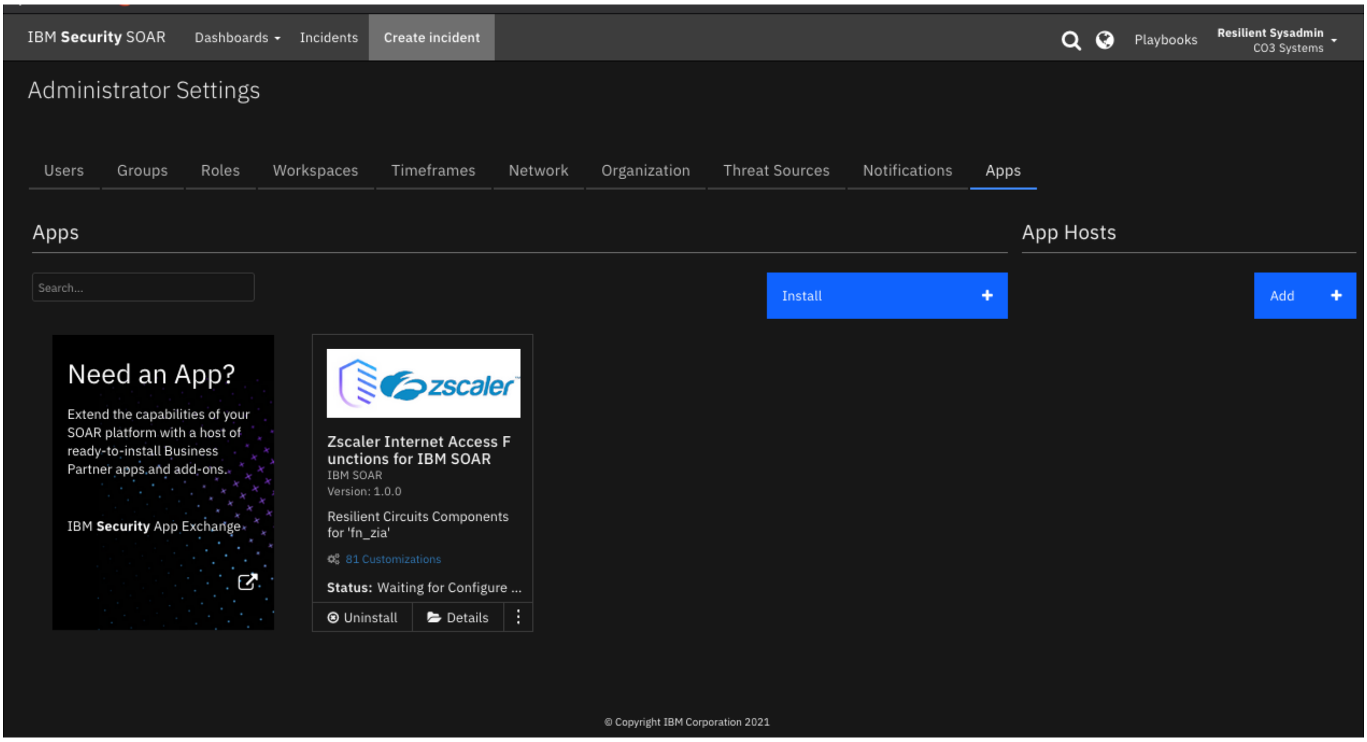
Table of Contents

Release Notes

Version	Date	Notes
1.0.0	06/2021	Initial Release
1.0.1	06/2022	Rate limiting support, function retry support, improved request error handling, container image refreshed to python 3.9

Overview

Zscaler Internet Access Integration for IBM SOAR.



Zscaler Internet Access(ZIA) is a secure internet and web gateway delivered as a service from the cloud. Zia provides a secure connection between users and the internet, inspecting every byte of traffic inline across multiple security techniques.

Key Features

The Zscaler Internet Access Integration provides the following functionality:

- Functions to get the global blocklist and allowlist.
- Functions to add URLs, URIs, DNS hostnames and IP addresses to the main blocklist and allowlist.
- Functions to remove URLs, URIs, DNS hostnames and IP addresses from the main blocklist and allowlist.
- Functions to get URL catergories and create custom URL categories.
- Functions to add URLs, URIs, DNS hostnames and IP addresses to custom URL categories.
- A function to query the ZIA sandbox for a hash value.
- A function to lookup the category of a URL.

Note: See the following for ZIA URL definitions https://help.zscaler.com/zia/url-format-guidelines.

Requirements

This app supports the IBM SOAR Platform and the IBM Cloud Pak for Security.

IBM SOAR platform

The IBM SOAR platform supports two app deployment mechanisms, App Host and integration server.

If deploying to an IBM SOAR platform with an App Host, the requirements are:

- IBM SOAR platform >= `39.0.6328`.
- The app is in a container-based format (available from the AppExchange as a `zip` file).

If deploying to an IBM SOAR platform with an integration server, the requirements are:

- IBM SOAR platform >= `39.0.6328`.
- The app is in the older integration format (available from the AppExchange as a `zip` file which contains a `tar.gz` file).
- Integration server is running `resilient_circuits>=35.0.0`.
- If using an API key account, make sure the account provides the following minimum permissions:

Name	Permissions
Org Data	Read
Function	Read

The following IBM SOAR platform guides provide additional information:

- *App Host Deployment Guide*: provides installation, configuration, and troubleshooting information, including proxy server settings.
- *Integration Server Guide*: provides installation, configuration, and troubleshooting information, including proxy server settings.
- *System Administrator Guide*: provides the procedure to install, configure and deploy apps.

The above guides are available on the IBM Documentation at ibm.biz/resilient-docs. On this web page, select your IBM SOAR platform version. On the follow-on page, you can find the *App Host Deployment Guide* or *Integration Server Guide* by expanding **Resilient Apps** in the Table of Contents pane. The System Administrator Guide is available by expanding **System Administrator**.

Cloud Pak for Security

If you are deploying to IBM Cloud Pak for Security, the requirements are:

- IBM Cloud Pak for Security >= 1.4.
- Cloud Pak is configured with an App Host.
- The app is in a container-based format (available from the AppExchange as a `zip` file).

The following Cloud Pak guides provide additional information:

- *App Host Deployment Guide*: provides installation, configuration, and troubleshooting information, including proxy server settings. From the Table of Contents, select Case Management and Orchestration & Automation > **Orchestration and Automation Apps**.
- *System Administrator Guide*: provides information to install, configure, and deploy apps. From the IBM Cloud Pak for Security Knowledge Center table of contents, select Case Management and Orchestration & Automation > **System administrator**.

These guides are available on the IBM Knowledge Center at ibm.biz/cp4s-docs. From this web page, select your IBM Cloud Pak for Security version. From the version-specific Knowledge Center page, select Case Management and Orchestration & Automation.

Proxy Server

The app **does** support a proxy server.

Python Environment

Python 3.6 is supported. Additional package dependencies may exist for each of these packages:

- resilient-circuits>=41.0.0
- resilient-lib>=41.0.0

Endpoint Developed With

This app has been implemented using:

Product Name	Product Version	API URL	API Version
Zscaler Internet Access	6.1	<zia_cloud_name>/api/v1	v1

Prerequisites

- The app user must have a ZIA API subscription.
- The user organization's API key must be enabled.

NOTE: API authentication is based on a combination of the API key and ZIA admin credentials (i.e., username and password).

Configuration

- Retrieve the Base URL and API key/token.

To locate the base URL and key/token:

- Log in to the ZIA Admin Portal using the users admin credentials.
- Go to Administration > Cloud Service API Key Management.

NOTE: In order to view the Cloud Service API Key Management page, the admin must be assigned an admin role that includes the Authentication Configuration functional scope.

See: ZIA API getting started

Permissions

- REST API access must be enabled for the account that IBM SOAR is communicating with as specified in the App configration file.

Installation

Install

- To install or uninstall an App or Integration on the *IBM SOAR platform*, see the documentation at ibm.biz/resilient-docs.
- To install or uninstall an App on *IBM Cloud Pak for Security*, see the documentation at ibm.biz/cp4s-docs and follow the instructions above to navigate to Orchestration and Automation.

App Configuration

The following table provides the settings you need to configure the app. These settings are made in the app.config file. See the documentation discussed in the Requirements section for the procedure.

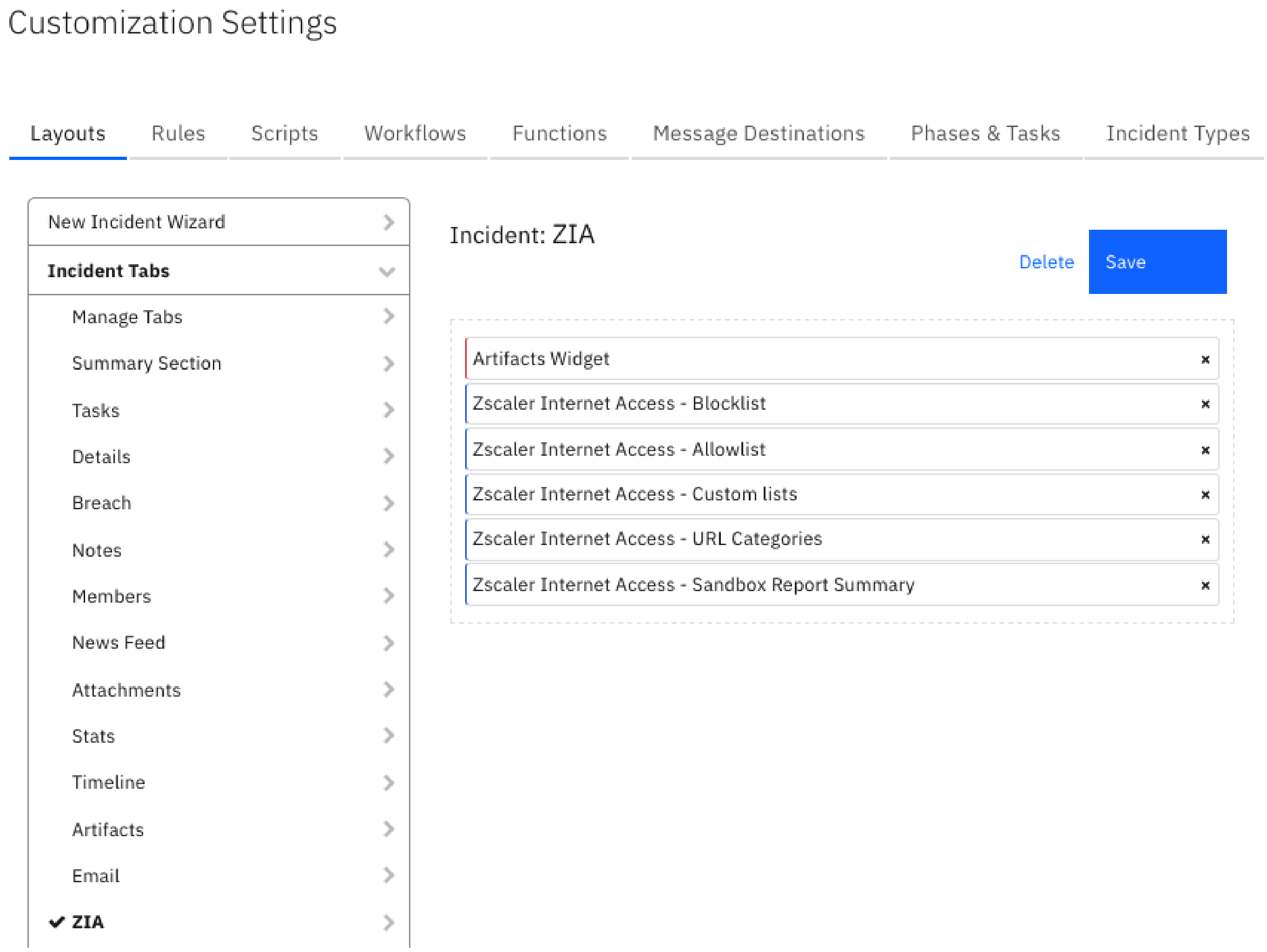
Config	Required	Example	Description
zia_api_base_url	Yes	`https://zsapi.zscaler.net/api/v1`	*ZIA API base URL. Dependent on particular ZIA cloud instance being accessed by user.*
zia_username	Yes	`user@1234567.zscalerbeta.net`	*Username used for programmatic (API) access to ZIA.*
zia_password	Yes	`password`	*Password used for programmatic (API) access to ZIA.*
zia_api_key	Yes	`abCDeFGHij0K`	*Api Key used for programmatic (API) access to ZIA.*
http_proxy	No	`http://proxy:80` or `proxy:80`	*Optional setting for an http proxy if required.*
https_proxy	No	`https://proxy:443` or `proxy:443`	*Optional setting for an https proxy if required.*

Custom Layouts

- Import the Data Tables and Custom Fields like the screenshot below: Create a new ZIA incident tab in Layouts as follows:

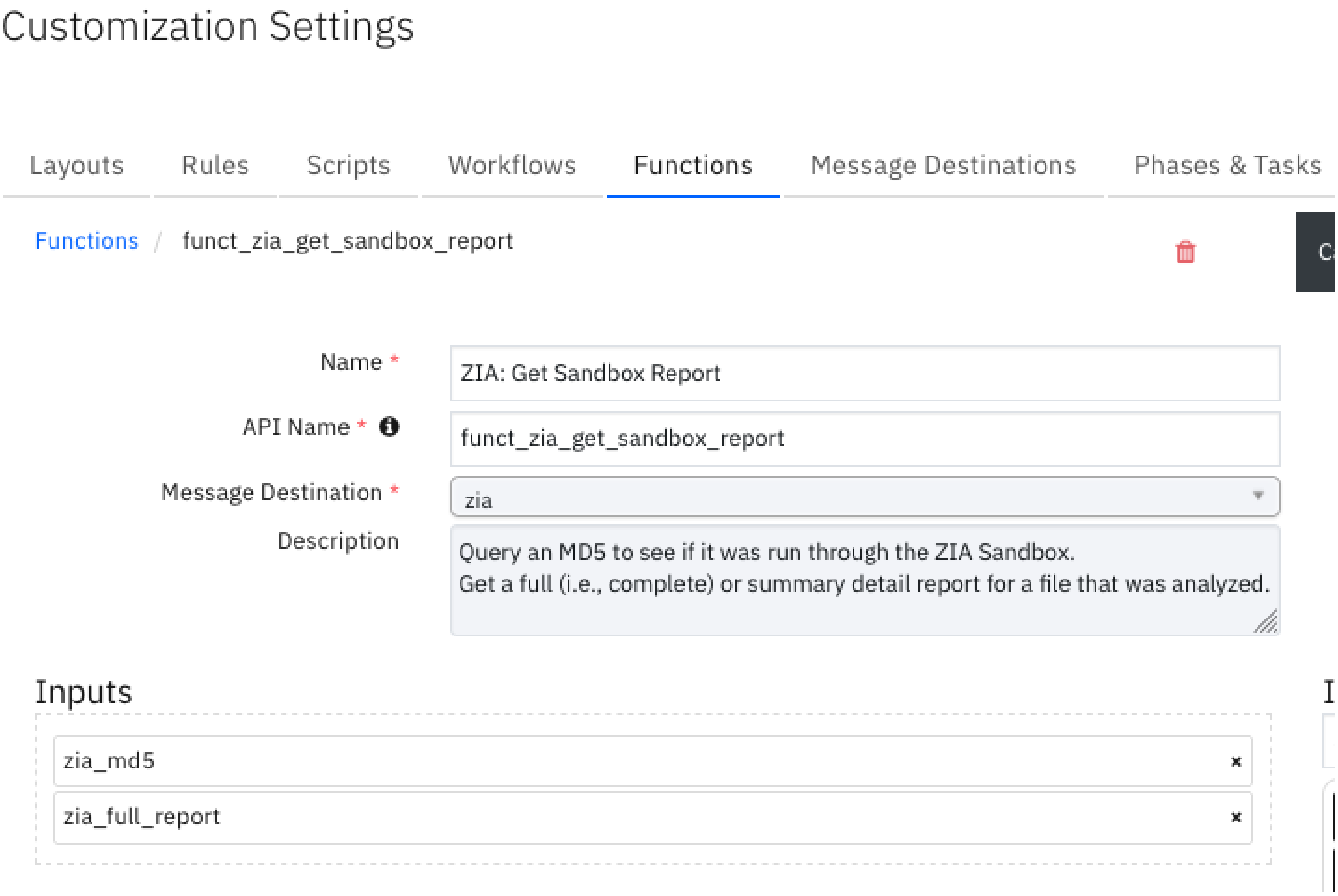
1. Navigate to the ‘Customization Settings’ and select the Layouts tab.
2. Click on ‘Incident Tabs’.
3. Add a new incident tab named ‘ZIA’.
4. Drag and drop the ZIA data tables under the new tab.
5. Click Save.

The following screenshot shows the data tables added to the ZIA tab:



Function - ZIA: Get Sandbox Report

Query an MD5 to see if it was run through the ZIA Sandbox. Get a full (i.e., complete) or summary detail report for a file that was analyzed.



Customization Settings

Functions / funct_zia_get_sandbox_report

Name * ZIA: Get Sandbox Report

API Name * funct_zia_get_sandbox_report

Message Destination * zia

Description Query an MD5 to see if it was run through the ZIA Sandbox.
Get a full (i.e., complete) or summary detail report for a file that was analyzed.

Inputs

zia_md5

zia_full_report

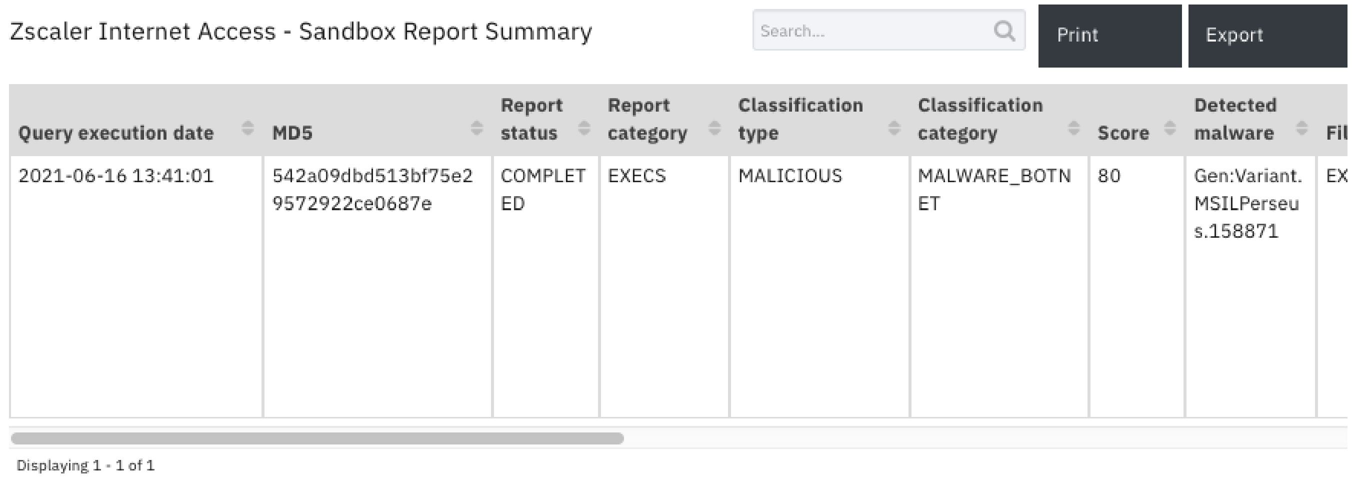
The function provides the following functionality.

- Query the ZIA sandbox for MD5 values where the associated file has been run through the sandbox.
- If the report type is `summary`, the data table `Zscaler Internet Access - Sandbox Report Summary` is updated.
- A note is added to the SOAR incident with the status of the query.
- If the report is a `full` report, the raw JSON is included in the note, the data table will not be updated.

An example workflow that uses this IBM SOAR function is `ZIA: Get Sandbox Report`.

The workflow is initiated by the manual artifact rule `ZIA: Get Sandbox Report`.

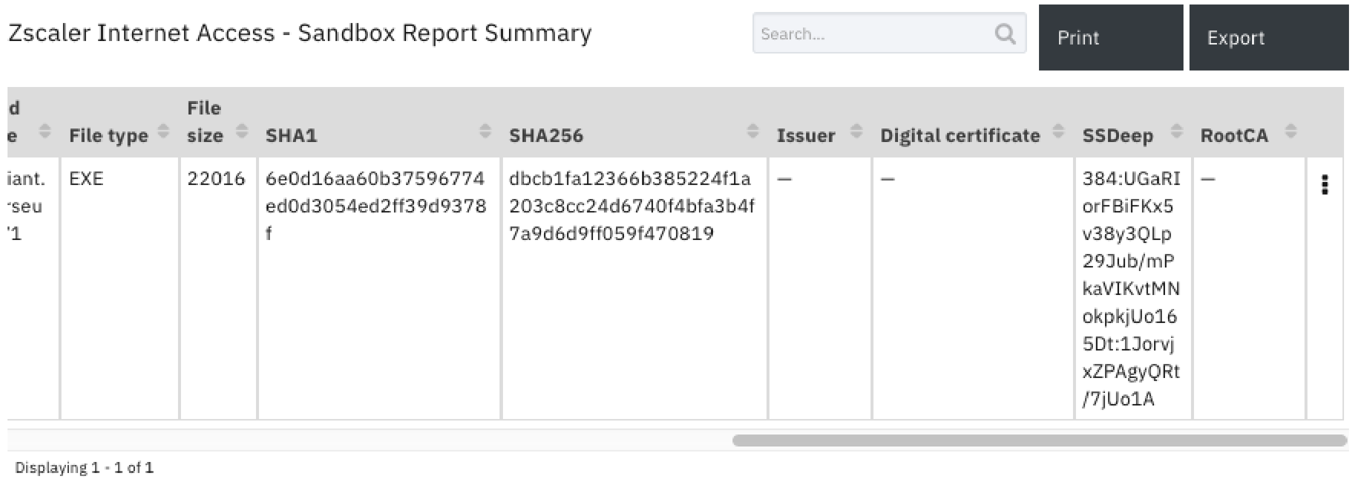
The following screenshot shows an example of the data table updated by the function:



Zscaler Internet Access - Sandbox Report Summary

Query execution date	MD5	Report status	Report category	Classification type	Classification category	Score	Detected malware	Fil
2021-06-16 13:41:01	542a09dbd513bf75e29572922ce0687e	COMPLETED	EXECS	MALICIOUS	MALWARE_BOTNET	80	Gen:Variant.MSILPerseus.158871	EX

Displaying 1 - 1 of 1

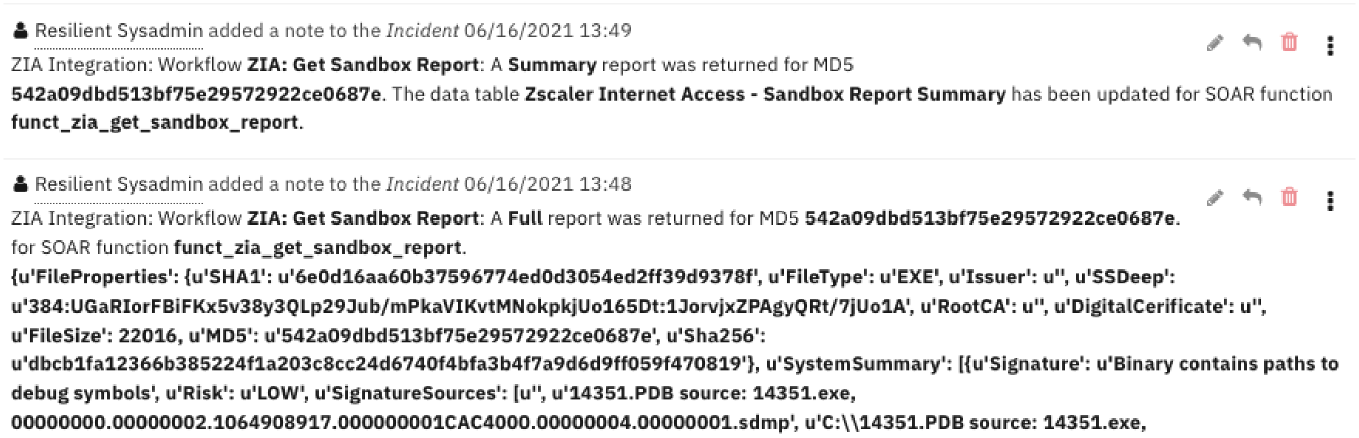


Zscaler Internet Access - Sandbox Report Summary

d e	File type	File size	SHA1	SHA256	Issuer	Digital certificate	SSDeep	RootCA	
iant. 'seu '1	EXE	22016	6e0d16aa60b37596774ed0d3054ed2ff39d9378f	dbcb1fa12366b385224f1a203c8cc24d6740f4bfa3b4f7a9d6d9ff059f470819	–	–	384:UGaRIorFBiFKx5v38y3QLp29Jub/mPkaVIKvtMNokpkjUo165Dt:1JorvjxZPAgyQRt/7jUo1A	–	⋮

Displaying 1 - 1 of 1

The following screenshot shows an example of notes added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 13:49

ZIA Integration: Workflow **ZIA: Get Sandbox Report**: A **Summary** report was returned for MD5 **542a09dbd513bf75e29572922ce0687e**. The data table **Zscaler Internet Access - Sandbox Report Summary** has been updated for SOAR function **funct_zia_get_sandbox_report**.

Resilient Sysadmin added a note to the *Incident* 06/16/2021 13:48

ZIA Integration: Workflow **ZIA: Get Sandbox Report**: A **Full** report was returned for MD5 **542a09dbd513bf75e29572922ce0687e**.
for SOAR function **funct_zia_get_sandbox_report**.
{u'FileProperties': {u'SHA1': u'6e0d16aa60b37596774ed0d3054ed2ff39d9378f', u'FileType': u'EXE', u'Issuer': u'', u'SSDeep': u'384:UGaRIorFBiFKx5v38y3QLp29Jub/mPkaVIKvtMNokpkjUo165Dt:1JorvjxZPAgyQRt/7jUo1A', u'RootCA': u'', u'DigitalCerificate': u'', u'FileSize': 22016, u'MD5': u'542a09dbd513bf75e29572922ce0687e', u'Sha256': u'dbcb1fa12366b385224f1a203c8cc24d6740f4bfa3b4f7a9d6d9ff059f470819'}, u'SystemSummary': [{u'Signature': u'Binary contains paths to debug symbols', u'Risk': u'LOW', u'SignatureSources': [u'', u'14351.PDB source: 14351.exe, 00000000.00000002.1064908917.000000001CAC4000.00000004.00000001.sdmp', u'C:\\14351.PDB source: 14351.exe,

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_full_report`	`boolean`	Yes	`-`	Boolean True if a full report required.
`zia_md5`	`text`	Yes	`-`	MD5 hash value.

▶ Outputs:

```
results = {
    'content': {
        "Summary": {
            "Summary": {
                "Status": "COMPLETED",
                "Category": "EXECS",
                "FileType": "DLL",
                "StartTime": 1522111841,
                "Duration": 481690
            },
            "Classification": {
                "Type": "MALICIOUS",
                "Category": "MALWARE_BOTNET",
                "Score": 82,
                "DetectedMalware": "Win32/TrojanDownloader.Banload.TNJ
trojan"
            },
            "FileProperties": {
                "FileType": "DLL",
                "FileSize": 2358272,
                "MD5": "b3b13c2fe5710507612106cb11ceced3",
                "SHA1": "6f30404f8b30812758acc06455bc95348c86f9f2",
                "Sha256":
"c77ab4c60b73c8f8135d54162813ab7c63432058f17ff00754d5fd547c22db76",
                "Issuer": "",
                "DigitalCerificate": "",
                "SSDeep":
"49152:mQU0HSp/RcGuBLe/PESBbFVZ86MfBWPvGZxnBGVV3NcKRLFcTOJP:mQUn6LsPQp6vko
iKt",
                "RootCA": ""
            }
        }
    },
    "inputs": {
        "zia_full_report": False,
        "zia_md5": "542a09dbd513bf75e29572922ce0687e"
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "raw": '{"Summary": {"Summary": {"Status": "COMPLETED", "Category":
"EXECS", "FileType": "EXE", "StartTime": 1620229320, "Duration": 658451},
"Classification": {"Type": "MALICIOUS", "Category": "MALWARE_BOTNET",
"Score": 80, "DetectedMalware": "Gen:Variant.MSILPerseus.158871"},
"FileProperties": {"FileType": "EXE", "FileSize": 22016, "MD5":
"542a09dbd513bf75e29572922ce0687e", "SHA1":
"6e0d16aa60b37596774ed0d3054ed2ff39d9378f", "Sha256":
"dbcb1fa12366b385224f1a203c8cc24d6740f4bfa3b4f7a9d6d9ff059f470819",
"Issuer": "", "DigitalCerificate": "", "SSDeep":
"384:UGaRIorFBiFKx5v38y3QLp29Jub/mPkaVIKvtMNokpkjUo165Dt:1JorvjxZPAgyQRt/7
jUo1A", "RootCA": ""}}}',
    "version": "1.0",
    "success": True,
    "reason": None
}
```

▶ Example Pre-Process Script:

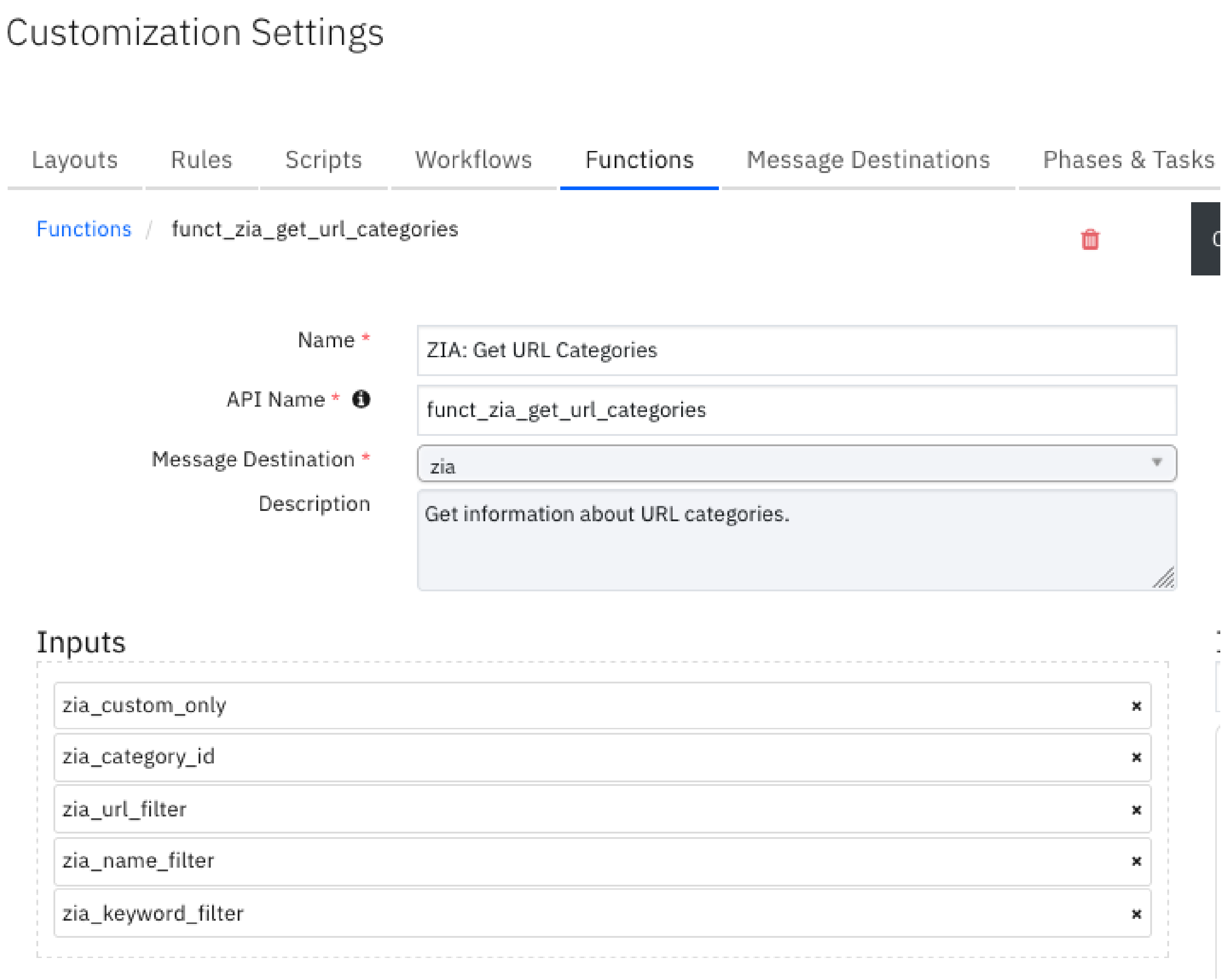
```
inputs.zia_md5 =  artifact.value
if rule.properties.zia_report_type.lower() == "full":
    inputs.zia_full_report = True
else:
    inputs.zia_full_report = False
```

▶ Example Post-Process Script:

```
##  ZIA - wf_zia_get_sandbox_report post processing script ##

#  Globals
FN_NAME = "funct_zia_get_sandbox_report"
WF_NAME = "ZIA: Get Sandbox Report"
CONTENT = results.content
INPUTS = results.inputs
QUERY_EXECUTION_DATE = results["metrics"]["timestamp"]

SUMM_HEADERS = ["Summary", "Classification", "FileProperties"]
DATA_TBL_FIELDS = {
    "Summary": ["Status", "report_Category"],
    "Classification": ["Type", "Category", "Score", "DetectedMalware"],
    "FileProperties":  ["FileType", "FileSize", "MD5", "SHA1", "Sha256",
"Issuer", "DigitalCerificate",
                        "SSDeep", "RootCA"]
}

#Processing
def main():
    note_text = u''
    bad_summary = u''
    bad_report_status = False
    unknown_md5_str = "md5 is unknown or analysis has yet not been
completed"
    md5 = INPUTS.get("zia_md5")
    report_type = "Full" if INPUTS.get("zia_full_report") else "Summary"
    if CONTENT:
        full = CONTENT.get("Full Details")
        summary = CONTENT.get("Summary")
        for r in list(filter(None, [full, summary])):
            if  not isinstance(r, dict) and r.find(unknown_md5_str) > -1:
                bad_summary = r
                bad_report_status = True
        if bad_report_status:
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: A <b>{1}</b> report was not returned for MD5 <b>{2}</b> " \
                        u"for SOAR function <b>{3}</b>." \
                .format(WF_NAME, report_type, md5, FN_NAME)
            note_text += "<br><b>{0}</b>".format(bad_summary)
        elif summary:
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: A <b>{1}</b> report was returned for MD5 <b>{2}</b>. " \
                        u"The data table <b>{3}</b> has been updated for SOAR function <b>{4}</b>."\
                .format(WF_NAME, report_type, md5, "Zscaler Internet Access - Sandbox Report Summary", FN_NAME)

            newrow = incident.addRow("zia_sandbox_report_summary")
            newrow.query_execution_date = QUERY_EXECUTION_DATE

            for header in SUMM_HEADERS:
                section = summary.get(header)
                for field in DATA_TBL_FIELDS[header]:
                    try:
                        f = field.split('_')[1]
                    except IndexError:
                        f = field
                    newrow[field]  = "{}".format(section[f])
        elif full:
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: A <b>{1}</b> report was returned for MD5 <b>{2}</b>. " \
                        u"for SOAR function <b>{3}</b>."\
                .format(WF_NAME, report_type, md5, FN_NAME)
            note_text += "<br><b>{0}</b>".format(full)
        else:
            note_text += u"ZIA Integration: Workflow <b>{0}</b>: There was an unknown report type returned for MD5 <b>{1}</b>" \
                         u" for SOAR function <b>{2}</b>.".format(WF_NAME, md5, FN_NAME)
    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: There were <b>no</b> results returned for MD5 <b>{1}</b>" \
                     u" for SOAR function <b>{2}</b>."\
            .format(WF_NAME, md5, FN_NAME)

    incident.addNote(helper.createRichText(note_text))

main()
```

Function - ZIA: Get URL Categories

Get information about URL categories.



Name * ZIA: Get URL Categories

API Name * funct_zia_get_url_categories

Message Destination * zia

Description Get information about URL categories.

Inputs

zia_custom_only ×

zia_category_id ×

zia_url_filter ×

zia_name_filter ×

zia_keyword_filter ×

The function provides the following functionality.

- Query ZIA for information on URL categories.
- The result can be filtered by category name and/or by URL values using regular expressions.
- A data table is updated (The data table depends on the workflow).
- A note is added to the IBM SOAR incident with the status of the query.
- If the number of categories or URL count are > 50 the raw JSON is included in the note, the data table will not be updated.

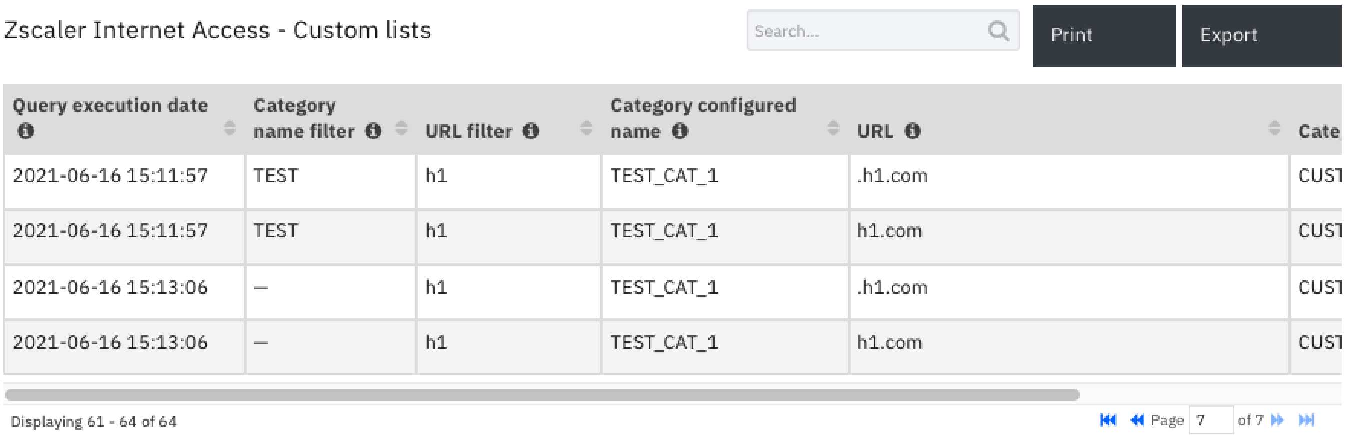
Note: The category name field is only valid for custom categories.

An example workflow that uses this IBM SOAR function is `ZIA: Get Customlist`.

The workflow is initiated by the manual incident rule `ZIA: Get Customlist`.

The data table `Zscaler Internet Access - Custom lists` is updated.

The following screenshot shows an example of data table updated by the function:

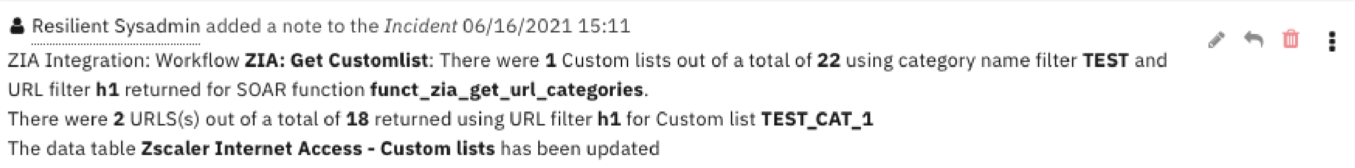


Zscaler Internet Access - Custom lists

Query execution date	Category name filter	URL filter	Category configured name	URL	Cate
2021-06-16 15:11:57	TEST	h1	TEST_CAT_1	.h1.com	CUST
2021-06-16 15:11:57	TEST	h1	TEST_CAT_1	h1.com	CUST
2021-06-16 15:13:06	—	h1	TEST_CAT_1	.h1.com	CUST
2021-06-16 15:13:06	—	h1	TEST_CAT_1	h1.com	CUST

Displaying 61 - 64 of 64

The following screenshot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 15:11

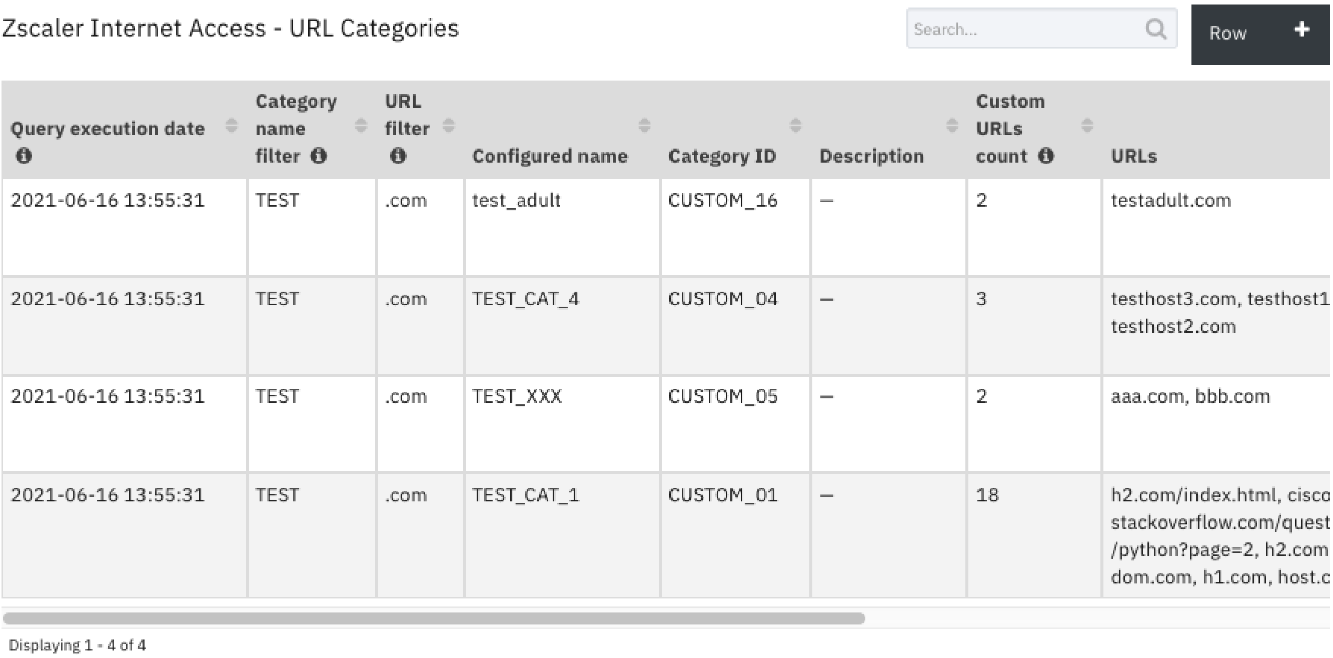
ZIA Integration: Workflow **ZIA: Get Customlist**: There were **1** Custom lists out of a total of **22** using category name filter **TEST** and URL filter **h1** returned for SOAR function **funct_zia_get_url_categories**.
There were **2** URLS(s) out of a total of **18** returned using URL filter **h1** for Custom list **TEST_CAT_1**
The data table **Zscaler Internet Access - Custom lists** has been updated

Another example workflow that uses this IBM SOAR function is `ZIA: Get URL Categories`.

The workflow is initiated by the manual incident rule `ZIA: Get URL Categories`.

The data table `Zscaler Internet Access - URL Categories` is updated.

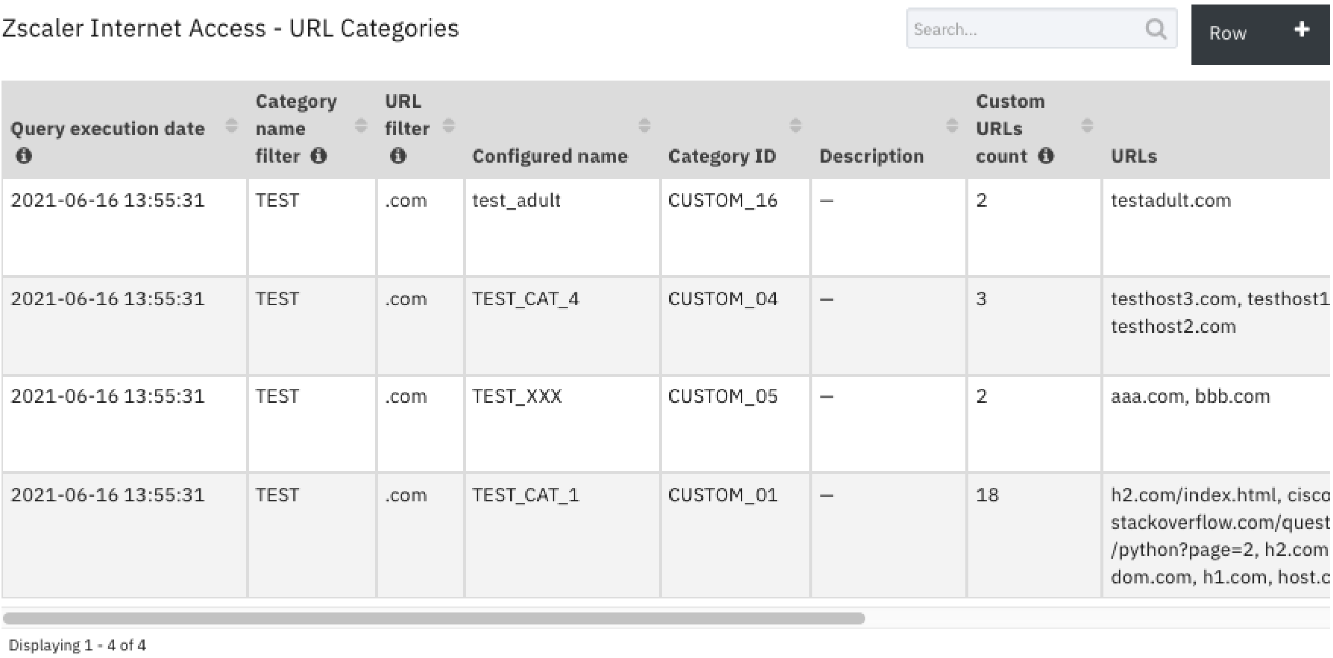
The following screenshot shows an example of data table updated by the function:



Zscaler Internet Access - URL Categories

Query execution date	Category name filter	URL filter	Configured name	Category ID	Description	Custom URLs count	URLs
2021-06-16 13:55:31	TEST	.com	test_adult	CUSTOM_16	–	2	testadult.com
2021-06-16 13:55:31	TEST	.com	TEST_CAT_4	CUSTOM_04	–	3	testhost3.com, testhost1 testhost2.com
2021-06-16 13:55:31	TEST	.com	TEST_XXX	CUSTOM_05	–	2	aaa.com, bbb.com
2021-06-16 13:55:31	TEST	.com	TEST_CAT_1	CUSTOM_01	–	18	h2.com/index.html, cisco stackoverflow.com/quest /python?page=2, h2.com dom.com, h1.com, host.c

Displaying 1 - 4 of 4

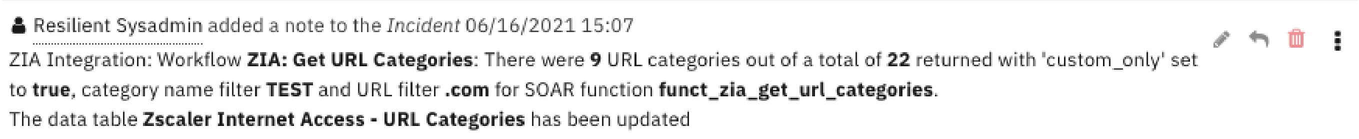

Search...
Row +

Zscaler Internet Access - URL Categories

Query execution date	Category name filter	URL filter	Configured name	Category ID	Description	Custom URLs count	URLs
2021-06-16 13:55:31	TEST	.com	test_adult	CUSTOM_16	–	2	testadult.com
2021-06-16 13:55:31	TEST	.com	TEST_CAT_4	CUSTOM_04	–	3	testhost3.com, testhost1 testhost2.com
2021-06-16 13:55:31	TEST	.com	TEST_XXX	CUSTOM_05	–	2	aaa.com, bbb.com
2021-06-16 13:55:31	TEST	.com	TEST_CAT_1	CUSTOM_01	–	18	h2.com/index.html, cisco stackoverflow.com/quest /python?page=2, h2.com dom.com, h1.com, host.c

Displaying 1 - 4 of 4

The following screenshot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 15:07

ZIA Integration: Workflow **ZIA: Get URL Categories**: There were **9** URL categories out of a total of **22** returned with 'custom_only' set to **true**, category name filter **TEST** and URL filter **.com** for SOAR function **funct_zia_get_url_categories**.

The data table **Zscaler Internet Access - URL Categories** has been updated

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_category_id`	`text`	No	`-`	Category ID of a URL category.
`zia_custom_only`	`text`	No	`-`	Parameter to get only Custom Categories. Options 'true' or 'false'.

Name	Type	Required	Example	Tooltip
`zia_name_filter`	`text`	No	`<REGEX>|<STRING>`	Filter by category name. Can be a string or regular expression. e.g. 'Custom category', '^CAT.*' or ^(List 1
`zia_url_filter`	`text`	No	`<REGEX>|<STRING>`	Filter by URL. Can be a string or regular expression. e.g. host.ibm.com, '^host.*' or ^(?:[0-9]{1,3}.){3}[0-9]{1,3}$. Leaving the filter blank will return the full list.

▶ Outputs:

```
results = {
    "content": {
        "categories": [
            {
                "id": "CUSTOM_01",
                "configuredName": "TEST_CAT_1",
                "urls": [
                    "testhost.com"
                ],
                "dbCategorizedUrls": [],
                "customCategory": True,
                "editable": True,
                "description": "CUSTOM_01_DESC",
                "type": "URL_CATEGORY",
                "val": 129,
                "customUrlsCount": 1,
                "urlsRetainingParentCategoryCount": 0,
                "url_counts": {
                    "total": 1,
                    "filtered": 1
                }
            }
        ],
        "category_counts": {
            "total": 5,
            "filtered": 1
        }
    },
    "inputs": {
        "zia_custom_only": "true",
        "zia_url_filter": "testhost",
        "zia_name_filter": "TEST",
        "zia_category_id": "CUSTOM_01"
    },
    "raw": '{"categories": [{"id": "CUSTOM_01", "configuredName":
"TEST_CAT_1", "urls": ["testhost.com"], "dbCategorizedUrls": [],
"customCategory": true, "editable": true, "description": "CUSTOM_01_DESC",
"type": "URL_CATEGORY", "val": 129, "customUrlsCount": 1,
"urlsRetainingParentCategoryCount": 0, "url_counts": {"total": 1,
"filtered": 1}}], "category_counts": {"total": 5, "filtered": 1}}',
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "version": "1.0",
    "success": True,
    "reason": None
}
```

▶ Example Pre-Process Script:

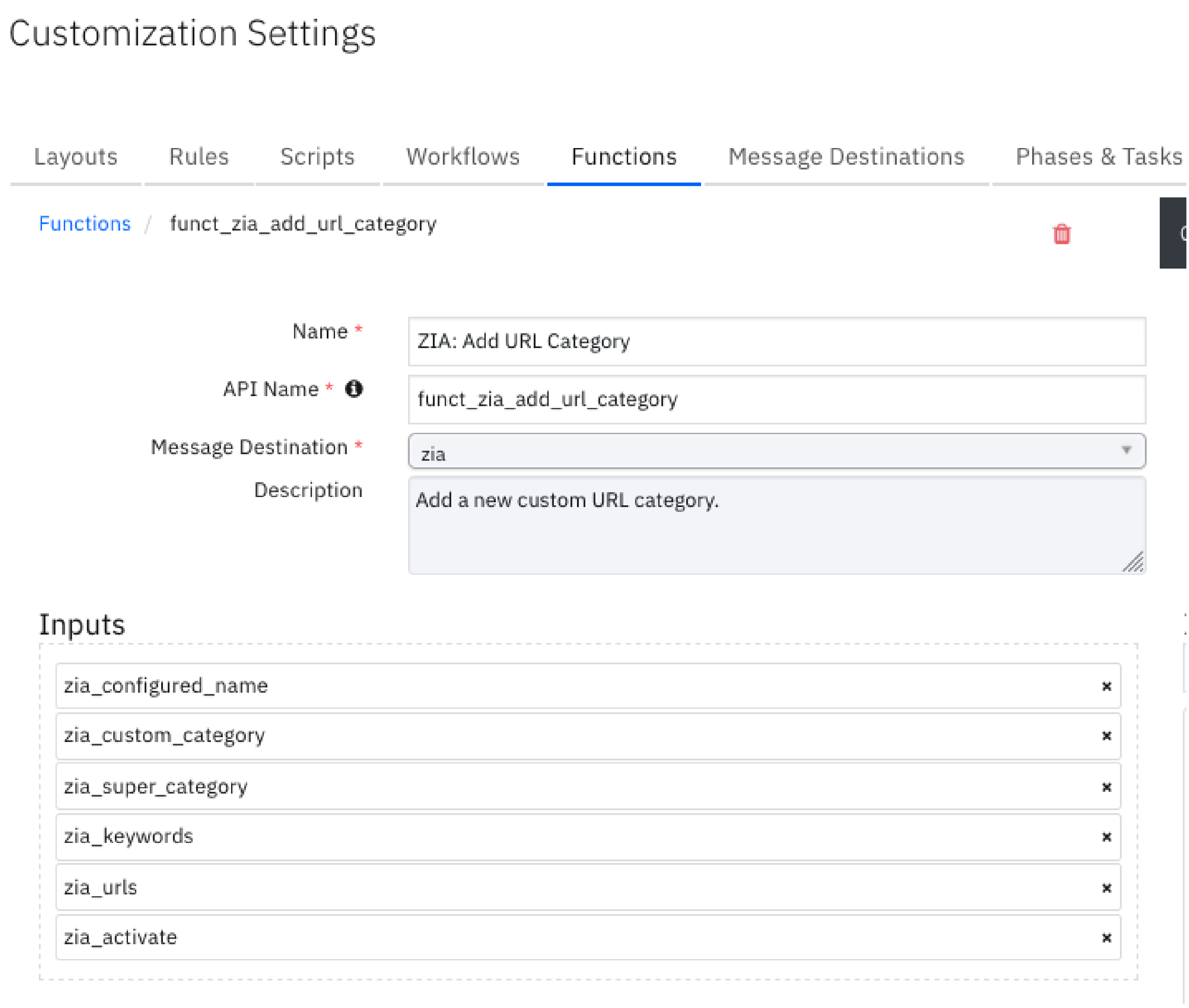
```
##  ZIA - wf_zia_get_customlist pre processing script ##
inputs.zia_custom_only = "true"
import re

URL_FILTER = rule.properties.zia_url_filter
NAME_FILTER = rule.properties.zia_name_filter

def is_regex(regex_str):
    """"Test if sting is a correctly formed regular expression.

    :param regex_str: Regular expression string.
    :return: Boolean.
    """
    try:
        re.compile(regex_str)
        return True
    except (re.error, TypeError):
        return False


def main():
    # Test filters to ensure they are valid regular expressions.
    for query_filter in [URL_FILTER, NAME_FILTER]:
        if query_filter and not is_regex(query_filter):
            raise ValueError("The filter '{}' is not a valid regular
expression.".format(unicode(repr(query_filter))))

    inputs.zia_url_filter = URL_FILTER
    inputs.zia_name_filter = NAME_FILTER

if __name__ == "__main__":
    main()
```

▶ Example Post-Process Script:

```
##  ZIA - wf_zia_get_customlist post processing script ##

#  Globals
FN_NAME = "funct_zia_get_url_categories"
WF_NAME = "ZIA: Add Artifact To Customlist"
CONTENT = results.content
INPUTS = results.inputs
note_text = ''


# Processing
def main():
    catname_exists = False
    note_text = u''
    name_filter = INPUTS.get("zia_name_filter")
    if CONTENT:
        cats = CONTENT.get("categories")
        if any(name_filter == c["configuredName"] for c in cats):
            catname_exists = True
    if catname_exists:
        workflow.addProperty("catname_exists", {})
    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: Workflow <b>{0}</b>: The category nmae  <b>{1}</b> was not found " \
                     u"for SOAR function <b>{2}</b>." \
            .format(WF_NAME, name_filter, FN_NAME)
        incident.addNote(helper.createRichText(note_text))

main()
```

Function - ZIA: Add URL Category

Add a new custom URL category.



Name * ZIA: Add URL Category

API Name * funct_zia_add_url_category

Message Destination * zia

Description Add a new custom URL category.

Inputs

zia_configured_name ×

zia_custom_category ×

zia_super_category ×

zia_keywords ×

zia_urls ×

zia_activate ×

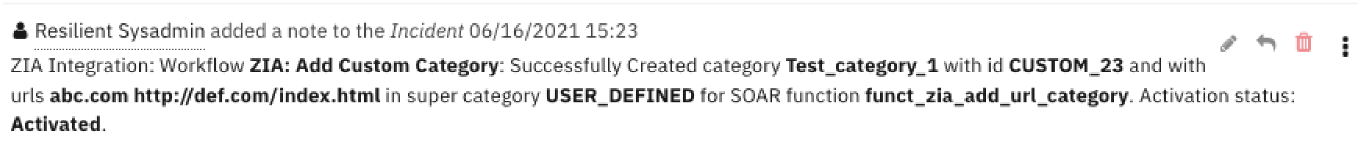
The function provides the following functionality.

- An action to add a new custom URL category to the ZIA environment.
- A list of URLs must be included in the action with at least one URL in the list.
- A super category must be specified.
- Set `zia_custom_category` to `true` to indicate this is a custom category.
- There is an option to activate the configuration change.
- A note is added to the IBM SOAR incident with the status of the action.

An example workflow that uses this IBM SOAR function is `ZIA: Add Custom Category`.

The workflow is initiated by the manual incident rule `ZIA: Add Custom Category`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 15:23

ZIA Integration: Workflow **ZIA: Add Custom Category**: Successfully Created category **Test_category_1** with id **CUSTOM_23** and with urls **abc.com http://def.com/index.html** in super category **USER_DEFINED** for SOAR function **funct_zia_add_url_category**. Activation status: **Activated**.

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_activate`	`boolean`	Yes	`-`	Activate the configuration on ZIA.
`zia_configured_name`	`text`	No	`-`	Configured name of a custom URL category.
`zia_custom_category`	`text`	No	`-`	Indicates whether category is custom type. Valid values are "true" or "false"
`zia_keywords`	`text`	No	`-`	Keyword string.
`zia_super_category`	`text`	Yes	`-`	Super category of a custom URL category.
`zia_urls`	`text`	Yes	`-`	Entries of type DNS hostname, IP Address, URL or URI. Wildcard is a '.' Entries will be parsed to extract a format suitable for ZIA to add to a list. URL guidelines https://help.zscaler.com/zia/url-format-guidelines.

▶ Outputs:

```
results = {
    "content": {
        "response": {
            "id": "CUSTOM_01",
            "configuredName": "TEST_CAT_1",
            "superCategory": "USER_DEFINED",
            "keywords": [],
            "keywordsRetainingParentCategory": [],
            "urls": [
                "192.168.1.1",
                "testhost.com"
            ],
            "dbCategorizedUrls": [],
            "customCategory": True,
            "editable": True,
            "type": "URL_CATEGORY",
            "val": 130,
            "customUrlsCount": 2,
            "urlsRetainingParentCategoryCount": 0
        },
        "activation": {
            "status": "Activated"
        }
    },
    "inputs": {
        "zia_custom_category": "true",
        "zia_configured_name": "TEST_CAT_1",
        "zia_urls": "testhost.com 192.168.1.1",
        "zia_super_category": "USER_DEFINED",
        "zia_keywords": "test1"
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "raw": '{"id": "CUSTOM_01", "configuredName": "TEST_CAT_1", "superCategory": "USER_DEFINED", "keywords": [], "keywordsRetainingParentCategory": [], "urls": ["192.168.1.1", "testhost.com"], "dbCategorizedUrls": [], "customCategory": true, "editable": true, "type": "URL_CATEGORY", "val": 130, "customUrlsCount": 2, "urlsRetainingParentCategoryCount": 0}',
    "version": "1.0",
    "success": True,
    "reason": None
}
```

▶ Example Pre-Process Script:

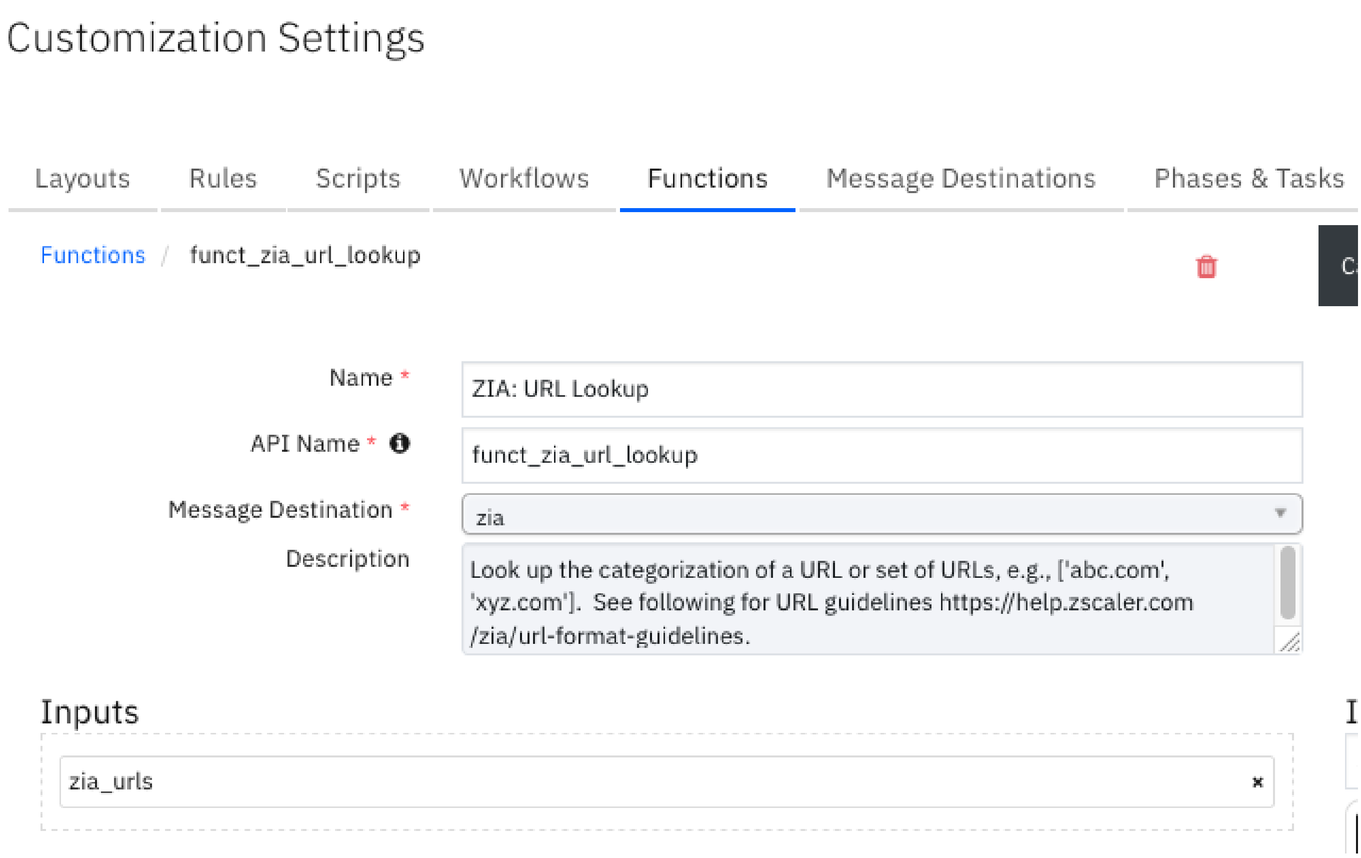
```
inputs.zia_configured_name = rule.properties.zia_configured_name_input
inputs.zia_custom_category = "true"
inputs.zia_super_category = rule.properties.zia_super_category
inputs.zia_urls = rule.properties.zia_urls.content
inputs.zia_activate = rule.properties.zia_activate
```

▶ Example Post-Process Script:

```
##  ZIA - wf_zia_add_url_category post processing script ##
import re
#  Globals
FN_NAME = "funct_zia_add_url_category"
WF_NAME = "ZIA: Add Custom Category"
CONTENT = results.content
INPUTS = results.inputs

# Processing
def main():
    note_text = u''

    urls = INPUTS.get("zia_urls")
    configured_name = INPUTS.get("zia_configured_name")

    if CONTENT:
        response = CONTENT.get("response")
        activation = CONTENT.get("activation")
        id = response.get("id")
        super_cat = response.get("superCategory")
        # In order to test all urls have been successfully added, convert
string of urls
        # to a list and convert urls to the format used by ZIA. e.g.
https://user:password@domain.com:port/index.html ->
        # domain.com:port/index.html
        list_urls = [re.sub(r'^.*\/\/(.*@)*(.*)', r'\2', u) for u in
re.split("\s+|,", urls)]
        category_list = response.get("urls")
        if all(a in category_list for a in list_urls):
            note_text = u"ZIA Integration: Workflow <b>{0}</b>:
Successfully Created category <b>{1}</b> with id "\
                        u"<b>{2}</b> and with urls <b>{3}</b> in super
category <b>{4}</b> for SOAR function <b>{5}</b>."\
            .format(WF_NAME, configured_name, id, urls, super_cat,
FN_NAME)
            note_text += u" Activation status: <b>{0}
</b>.".format(activation["status"])

        else:
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Category
<b>{1}</b> creation not successfull " \
                        u"with URLs <b>{2}</b>  for SOAR function <b>{3}
</b>."\
                .format(WF_NAME, configured_name, urls, FN_NAME)
    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: There was
<b>no</b> result returned while attempting " \
                     u"to create category <b>{1}</b> with URLS <b>{2}</b>
for SOAR function <b>{3}</b>."\
            .format(WF_NAME, configured_name, urls, FN_NAME)

    incident.addNote(helper.createRichText(note_text))

main()
```

Function - ZIA: URL Lookup

Look up the categorization of a URL or set of URLs, e.g., ['abc.com', 'xyz.com']. See the following for URL guidelines https://help.zscaler.com/zia/url-format-guidelines.



Customization Settings

Name * ZIA: URL Lookup

API Name * funct_zia_url_lookup

Message Destination * zia

Description Look up the categorization of a URL or set of URLs, e.g., ['abc.com', 'xyz.com']. See following for URL guidelines https://help.zscaler.com/zia/url-format-guidelines.

Inputs

zia_urls

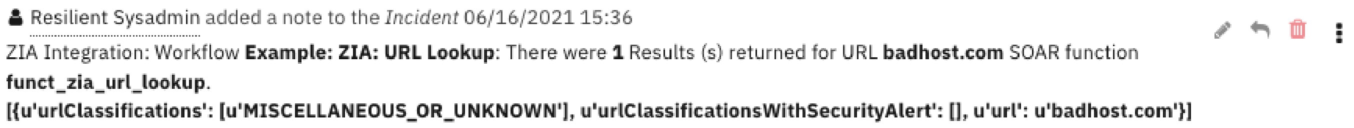
The function provides the following functionality.

- A query which looks up the categorization of a URL in the ZIA environment.
- A note is added to the IBM SOAR incident with the status of the query including the query result as raw JSON data.

An example workflow that uses this IBM SOAR function is `ZIA: URL Lookup`.

The workflow is initiated by the manual artifact rule `ZIA: URL Lookup`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 15:36

ZIA Integration: Workflow **Example: ZIA: URL Lookup**: There were **1** Results (s) returned for URL **badhost.com** SOAR function **funct_zia_url_lookup**.

[{u'urlClassifications': [u'MISCELLANEOUS_OR_UNKNOWN'], u'urlClassificationsWithSecurityAlert': [], u'url': u'badhost.com'}]

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_urls`	`text`	Yes	`-`	Entries of type DNS hostname, IP Address, URL or URI. Wildcard is a period (.). Entries will be parsed to extract a format suitable for ZIA to add to a list. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.

▶ Outputs:

```
results = {
    "content": [
        {
            "url": "host.com",
            "urlClassifications": ["PROFESSIONAL_SERVICES"],
            "urlClassificationsWithSecurityAlert": []
        }
    ],
    "inputs": {
        "zia_urls": "host.com"
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "raw": '[{"url": "host.com", "urlClassifications": ["PROFESSIONAL_SERVICES"], "urlClassificationsWithSecurityAlert": []}]',
    "version": "1.0",
    "success": True,
    "reason": None
}
```

▶ Example Pre-Process Script:

```
inputs.zia_urls = artifact.value
```

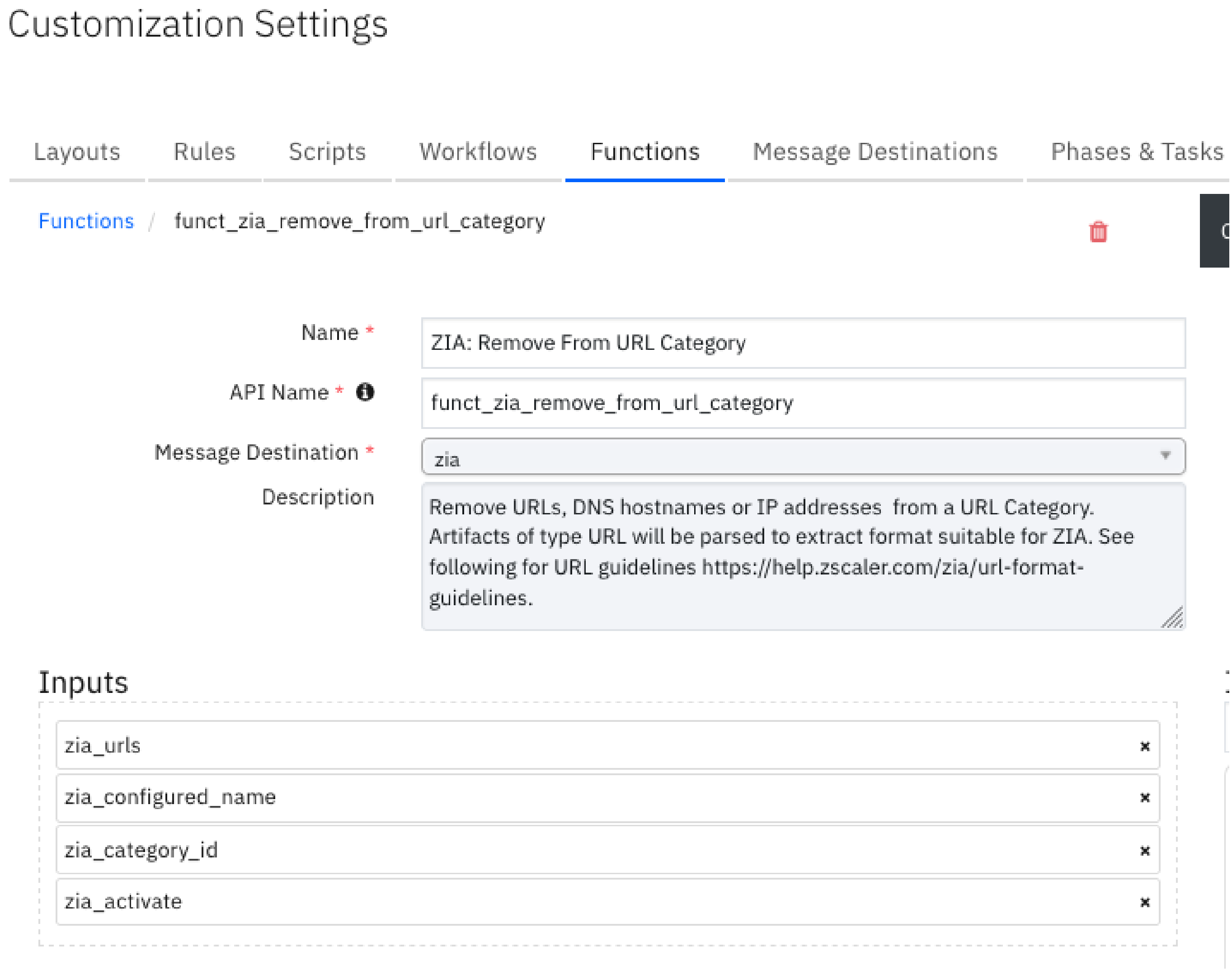
▶ Example Post-Process Script:

```
##  ZIA - wf_zia_url_lookup post processing script ##

#  Globals
FN_NAME = "funct_zia_url_lookup"
WF_NAME = "Example: ZIA: URL Lookup"
CONTENT = results.content
INPUTS = results.inputs
QUERY_EXECUTION_DATE = results["metrics"]["timestamp"]

# Processing
def main():
    note_text = u''
    urls = INPUTS.get("zia_urls")
    if CONTENT:
        note_text = u"ZIA Integration: Workflow <b>{0}</b>: There were <b>{1}</b> Results (s) returned for " \
                        u"URL <b>{2}</b> SOAR function <b>{3}
```

```
</b>.".format(WF_NAME, len(CONTENT), urls, FN_NAME)
        note_text += u"<br><b>{}</b>".format(CONTENT)

    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: There were
<b>no</b> results returned " \
                     u"for SOAR function <b>{1}</b>."\
            .format(WF_NAME, FN_NAME)

    incident.addNote(helper.createRichText(note_text))

main()
```

Function - ZIA: Remove From URL Category

Remove URLs, DNS hostnames or IP addresses from a URL Category. Artifacts of type URL are parsed to extract the format suitable for ZIA. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.



Customization Settings

Name *	ZIA: Remove From URL Category
API Name *	funct_zia_remove_from_url_category
Message Destination *	zia
Description	Remove URLs, DNS hostnames or IP addresses from a URL Category. Artifacts of type URL will be parsed to extract format suitable for ZIA. See following for URL guidelines https://help.zscaler.com/zia/url-format-guidelines.

Inputs

- zia_urls
- zia_configured_name
- zia_category_id
- zia_activate

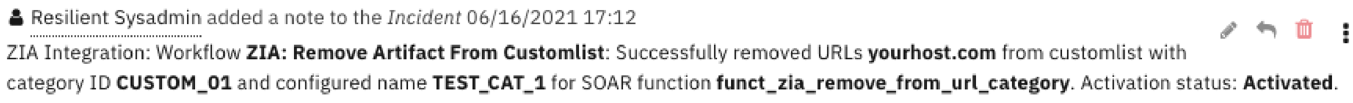
The function provides the following functionality.

- An action to remove a URL or list of URLs from a custom URL category in the ZIA environment.
- A URL or list of URLs must be included in the action.
- A custom category name and category ID must be included in the action.
- In the example workflows the category name is set from a user defined drop-down list.
- In the example workflows the category ID is obtained by querying ZIA by category name.
- There is an option to activate the configuration change.
- A note is added to the IBM SOAR incident with the status of the action.

An example workflow that uses this IBM SOAR function is `ZIA: Remove Artifact From Customlist`.

The workflow is initiated by the manual artifact rule `ZIA: Remove Artifact From Customlist`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



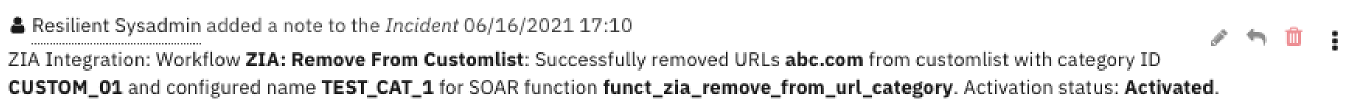
Resilient Sysadmin added a note to the *Incident* 06/16/2021 17:12

ZIA Integration: Workflow **ZIA: Remove Artifact From Customlist**: Successfully removed URLs **yourhost.com** from customlist with category ID **CUSTOM_01** and configured name **TEST_CAT_1** for SOAR function **funct_zia_remove_from_url_category**. Activation status: **Activated**.

Another example workflow that uses this IBM SOAR function is `ZIA: Remove From Customlist`.

The workflow is initiated by the manual data table rule `ZIA: Remove From Customlist` for data table `Zscaler Internet Access - Custom lists`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 17:10

ZIA Integration: Workflow **ZIA: Remove From Customlist**: Successfully removed URLs **abc.com** from customlist with category ID **CUSTOM_01** and configured name **TEST_CAT_1** for SOAR function **funct_zia_remove_from_url_category**. Activation status: **Activated**.

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_activate`	`boolean`	Yes	`-`	Activate the configuration on ZIA.
`zia_category_id`	`text`	No	`-`	Category ID of a URL category.
`zia_configured_name`	`text`	No	`-`	Configured name of a custom URL category.
`zia_urls`	`text`	Yes	`-`	Entries of type DNS hostname, IP Address, URL or URI. Wildcard is a period (.). Entries are parsed to extract a format suitable for ZIA to add to a list. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.

▶ Outputs:

```
results = {
    "content": {
        "response": {
            "id": "CUSTOM_01",
            "configuredName": "TEST_CAT_1",
            "keywordsRetainingParentCategory": [],
            "urls": [
                "1.1.1.1",
                "testhost.com"
            ],
            "dbCategorizedUrls": [],
            "customCategory": True,
            "editable": True,
            "description": "CUSTOM_01_DESC",
            "type": "URL_CATEGORY",
            "val": 128, "customUrlsCount": 2,
            "urlsRetainingParentCategoryCount": 0
        },
        "activation": {
            "status": "Activated"
        }
    },
    "inputs": {
        "zia_configured_name": "TEST_CAT_1",
        "zia_urls": "192.168.1.1",
        "zia_category_id": "CUSTOM_01"
    },
    "raw": '{"id": "CUSTOM_01", "configuredName": "TEST_CAT_1",
"keywordsRetainingParentCategory": [], "urls": ["1.1.1.1",
"testhost.com"], "dbCategorizedUrls": [], "customCategory": true,
"editable": true, "description": "CUSTOM_01_DESC", "type": "URL_CATEGORY",
"val": 128, "customUrlsCount": 2, "urlsRetainingParentCategoryCount": 0}',
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "version": "1.0",
    "success": True,
    "reason": None
}
```

▶ Example Pre-Process Script:

```
content = workflow.properties.get_categories_results.content
configured_name = rule.properties.zia_configured_name
cats = content.get("categories")
inputs.zia_category_id = [c["id"] for c in cats if configured_name ==
c["configuredName"]][0]
inputs.zia_urls = artifact.value
inputs.zia_configured_name = rule.properties.zia_configured_name
inputs.zia_activate = rule.properties.zia_activate
```

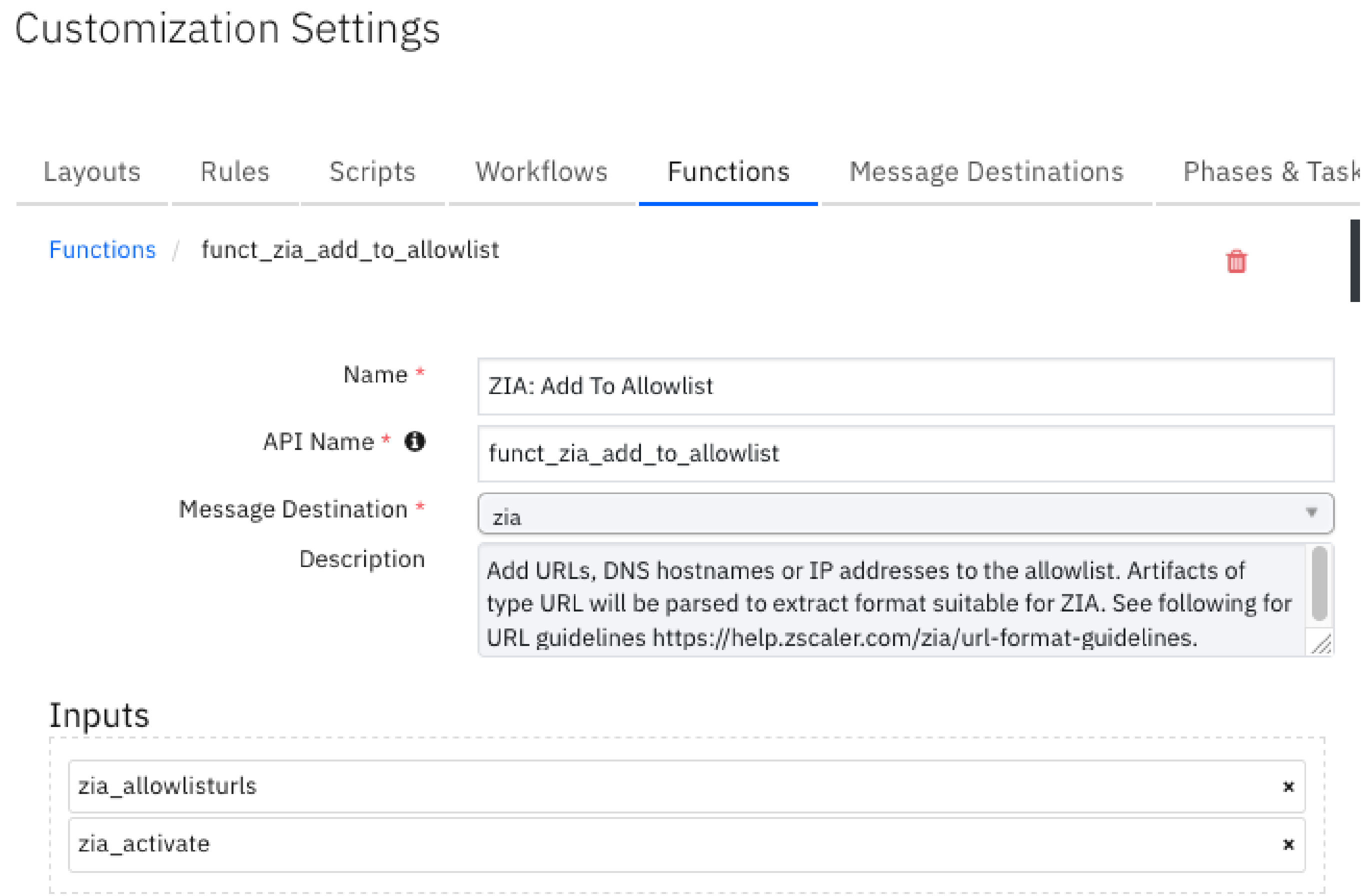
▶ Example Post-Process Script:

```
##  ZIA - wf_zia_remove_artifact_from_customlist post processing script ##
import re
#  Globals
FN_NAME = "funct_zia_remove_from_url_category"
WF_NAME = "ZIA: Remove Artifact From Customlist"
CONTENT = results.content
INPUTS = results.inputs

def main():
    note_text = u''
    urls = INPUTS.get("zia_urls")
    category_id = INPUTS.get("zia_category_id")
    configured_name = INPUTS.get("zia_configured_name")
    if CONTENT:
        response = CONTENT.get("response")
        activation = CONTENT.get("activation")
        # In order to test all urls have been successfully added, convert
string of urls
        # to a list and convert urls to the format used by ZIA. e.g.
https://user:password@domain.com:port/index.html ->
        # domain.com:port/index.html
        customlist_urls = [re.sub(r'^.*\/\/(.*@)*(.*)', r'\2', u) for u in
re.split("\s+|,", urls)]
        updated_customlist = response.get("urls")
        if not any(a in updated_customlist for a in customlist_urls):
            note_text = u"ZIA Integration: Workflow <b>{0}</b>:
Successfully removed URLs <b>{1}</b> from customlist "\
                        u"with category ID <b>{2}</b> and configured name
<b>{3}</b> for SOAR function <b>{4}</b>."\
            .format(WF_NAME, urls, category_id, configured_name, FN_NAME)
            note_text += u" Activation status: <b>{0}
</b>.".format(activation["status"])
        else:
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Not all
URLs were removed while attempting "\
                        u"to remove URLs <b>{1}</b> from customlist with
category ID <b>{2}</b> and configured name <b>{3}</b> "\
                        u"for SOAR function <b>{4}</b>.".format(WF_NAME,
urls, category_id, configured_name, FN_NAME)
    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: There was
<b>no</b> result returned while attempting " \
                     u"to remove URLs <b>{1}</b> from customlist of
category ID <b>{2}</b> for SOAR function <b>{3}</b>."\
            .format(WF_NAME, urls, category_id, FN_NAME)

    incident.addNote(helper.createRichText(note_text))
```

```
main()
```

Function - ZIA: Add To Allowlist

Add URLs, DNS hostnames or IP addresses to the allowlist. Artifacts of type URL are parsed to extract a format suitable for ZIA. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.



Name * ZIA: Add To Allowlist

API Name * funct_zia_add_to_allowlist

Message Destination * zia

Description Add URLs, DNS hostnames or IP addresses to the allowlist. Artifacts of type URL will be parsed to extract format suitable for ZIA. See following for URL guidelines https://help.zscaler.com/zia/url-format-guidelines.

Inputs

zia_allowlisturls

zia_activate

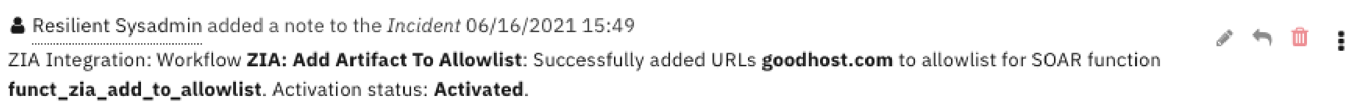
The function provides the following functionality.

- An action to add a URL or list of URLs to the main allow list.
- A URL or list of URLs must be included in the action.
- There is an option to activate the configuration change.
- A note is added to the IBM SOAR incident with the status of the action.

An example workflow that uses this IBM SOAR function is `Add Artifact To Allowlist`.

The workflow is initiated by the manual artifact rule `Add Artifact To Allowlist`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



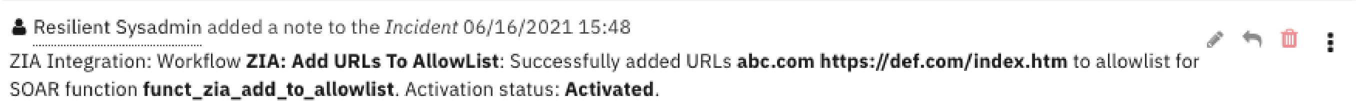
Resilient Sysadmin added a note to the *Incident* 06/16/2021 15:49

ZIA Integration: Workflow **ZIA: Add Artifact To Allowlist**: Successfully added URLs **goodhost.com** to allowlist for SOAR function **funct_zia_add_to_allowlist**. Activation status: **Activated**.

Another example workflow that uses this IBM SOAR function is `Add URLs To AllowList`.

The workflow is initiated by the manual incident rule `Add URLs To AllowList`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 15:48

ZIA Integration: Workflow **ZIA: Add URLs To AllowList**: Successfully added URLs **abc.com https://def.com/index.htm** to allowlist for SOAR function **funct_zia_add_to_allowlist**. Activation status: **Activated**.

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_activate`	`boolean`	Yes	`-`	Activate the configuration on ZIA.
`zia_allowlisturls`	`text`	No	`-`	The URLs, URIs, DNS hostnames or IP addresses to the allowlist. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.

▶ Outputs:

```
results = {
    "content": {
        "response": {
            "whitelistUrls": [
                "1.1.1.1",
                "goodhost.com"
            ]
        },
        "activation": {
            "status": "Activated"
        }
    },
    "inputs": {
        "zia_allowlisturls": "goodhost.com"
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "raw": '{"whitelistUrls": ["1.1.1.1", "goodhost.com"]}',
    "version": "1.0",
    "success": True,
    "reason": None
}
```

▶ Example Pre-Process Script:

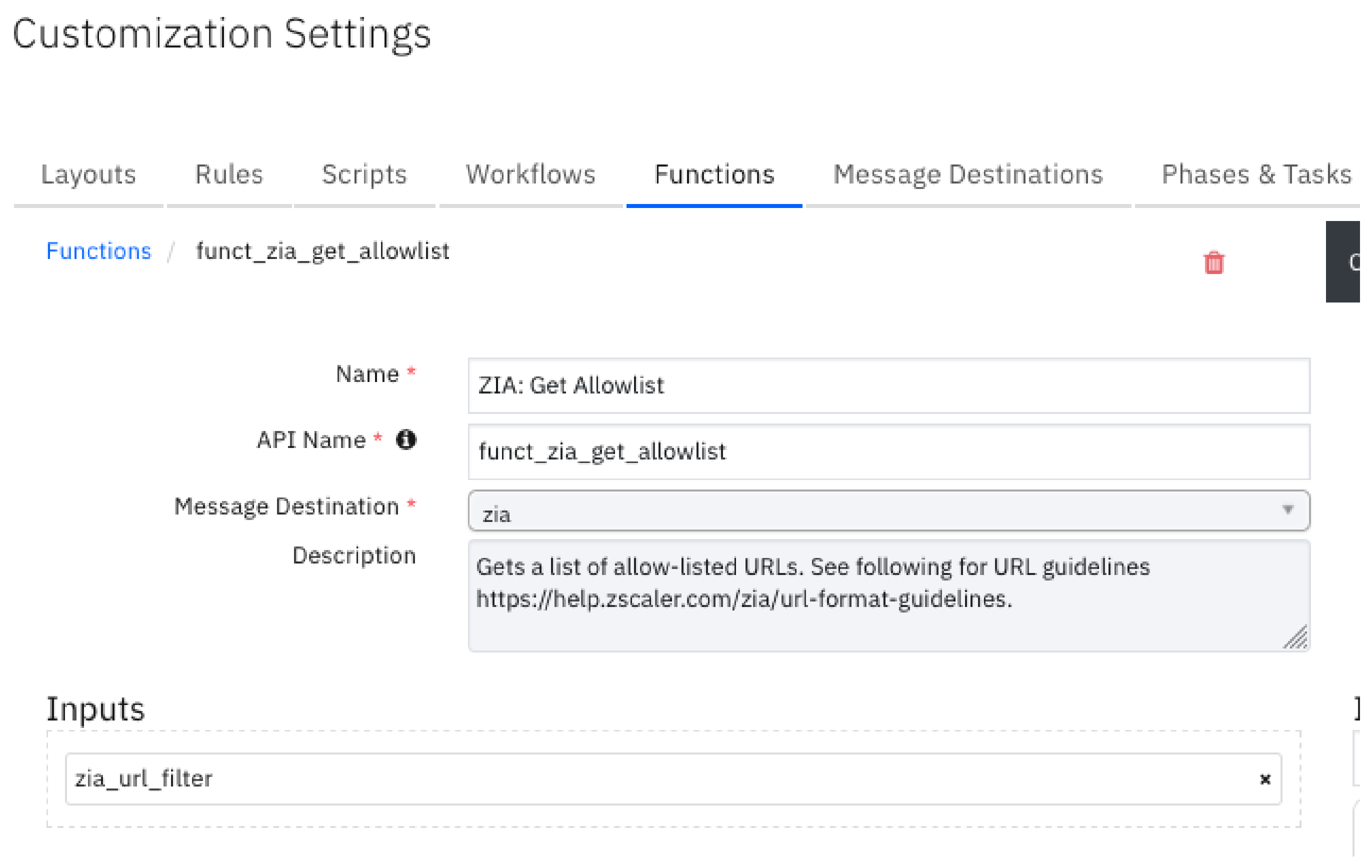
```
inputs.zia_allowlisturls = artifact.value
inputs.zia_activate = rule.properties.zia_activate
```

▶ Example Post-Process Script:

```
##  ZIA - zia_add_artifact_to_allowlist post processing script ##
import re
#  Globals
FN_NAME = "funct_zia_add_to_allowlist"
WF_NAME = "ZIA: Add Artifact To Allowlist"
CONTENT = results.content
INPUTS = results.inputs

# Processing
def main():
    note_text = u''
    urls = INPUTS.get("zia_allowlisturls")
    if CONTENT:
        response = CONTENT.get("response")
        activation = CONTENT.get("activation")
        # In order to test all urls have been successfully added, convert string of urls
        # to a list and convert urls to the format used by ZIA. e.g. https://user:password@domain.com:port/index.html ->
        # domain.com:port/index.html
        allowlist_urls = [re.sub(r'^.*\/\/(.*@)*(.*)', r'\2', u) for u in re.split("\s+|,", urls)]
        updated_allowlist = response.get("whitelistUrls")
        if all(a in updated_allowlist for a in allowlist_urls):
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Successfully added URLs <b>{1}</b> to allowlist " \
                        u"for SOAR function <b>{2}</b>.".format(WF_NAME, urls, FN_NAME)
            note_text += u" Activation status: <b>{0}</b>.".format(activation["status"])

        else:
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Not all URLss added while attempting " \
                        u"to add URLs <b>{2}</b> to allowlist for SOAR function <b>{2}</b>."\
                .format(WF_NAME, urls, FN_NAME)
    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: There was <b>no</b> result returned while attempting " \
                     u"to add URLs <b>{1}</b> to allowlist for SOAR function <b>{2}</b>."\
            .format(WF_NAME, urls, FN_NAME)
    incident.addNote(helper.createRichText(note_text))

main()
```

Function - ZIA: Get Allowlist

Gets a list of allow-listed URLs. See the following URL guidelines: https://help.zscaler.com/zia/url-format-guidelines.



Name * ZIA: Get Allowlist

API Name * funct_zia_get_allowlist

Message Destination * zia

Description Gets a list of allow-listed URLs. See following for URL guidelines https://help.zscaler.com/zia/url-format-guidelines.

Inputs

zia_url_filter

The function provides the following functionality.

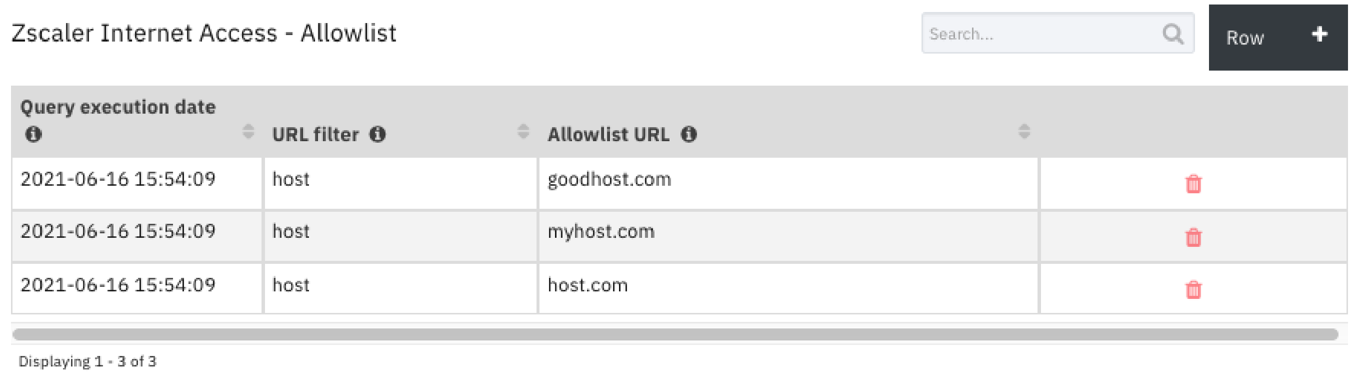
- Query ZIA for information on the main allow list.
- The result can be filtered by URL values using regular expressions.
- A data table is updated.
- A note is added to the IBM SOAR incident with the status of the query.
- If the URL count is > 50, the raw JSON is included in the note and the data table is not updated.

An example workflow that uses this IBM SOAR function is `Get Allowlist`.

The workflow is initiated by the manual incident rule `Get Allowlist`.

The data table `Zscaler Internet Access - Allowlist` is updated.

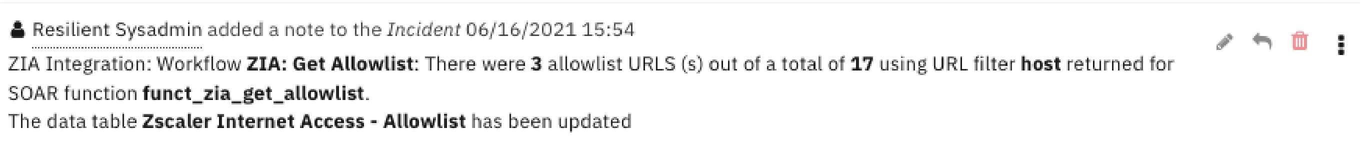
The following screenshot shows an example of data table updated by the function:



Query execution date	URL filter	Allowlist URL	
2021-06-16 15:54:09	host	goodhost.com	
2021-06-16 15:54:09	host	myhost.com	
2021-06-16 15:54:09	host	host.com	

Displaying 1 - 3 of 3

The following screenshot shows an example of a note added to an IBM SOAR incident:


Resilient Sysadmin added a note to the *Incident* 06/16/2021 15:54

ZIA Integration: Workflow **ZIA: Get Allowlist**: There were **3** allowlist URLS (s) out of a total of **17** using URL filter **host** returned for SOAR function **funct_zia_get_allowlist**.
The data table **Zscaler Internet Access - Allowlist** has been updated

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_url_filter`	`text`	No	`<REGEX>|<STRING>`	Filter by URL. Can be a string or regular expression. e.g. host.ibm.com, ^host.*' or ^(?:[0-9]{1,3}.){3}[0-9]{1,3}$. Leaving the filter blank will return the full list.

▶ Outputs:

```
results = {
    "content": {
        "whitelistUrls": [
            "1.1.1.1",
            "goodhost.com"
        ],
        "url_counts": {
            "total": 3,
            "filtered": 3
        }
    },
    "inputs": {
        "zia_url_filter": ".*"
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "raw": '{"whitelistUrls": ["1.1.1.1", "goodhost.com"]}',
    "version": "1.0",
    "success": True,
    "reason": None,
}
```

▶ Example Pre-Process Script:

```
##  ZIA - wf_zia_get_allowlist post processing script ##
import re

URL_FILTER = rule.properties.zia_url_filter

def is_regex(regex_str):
    """"Test if sting is a correctly formed regular expression.

    :param regex_str: Regular expression string.
    :return: Boolean.
    """
    try:
        re.compile(regex_str)
        return True
    except (re.error, TypeError):
        return False


def main():
    # Test filter to ensure it is a valid regular expressions.
    if URL_FILTER and not is_regex(URL_FILTER):
        raise ValueError("The query filter '{}' is not a valid regular
expression.".format(unicode(URL_FILTER)))

    inputs.zia_url_filter = rule.properties.zia_url_filter

if __name__ == "__main__":
    main()
```

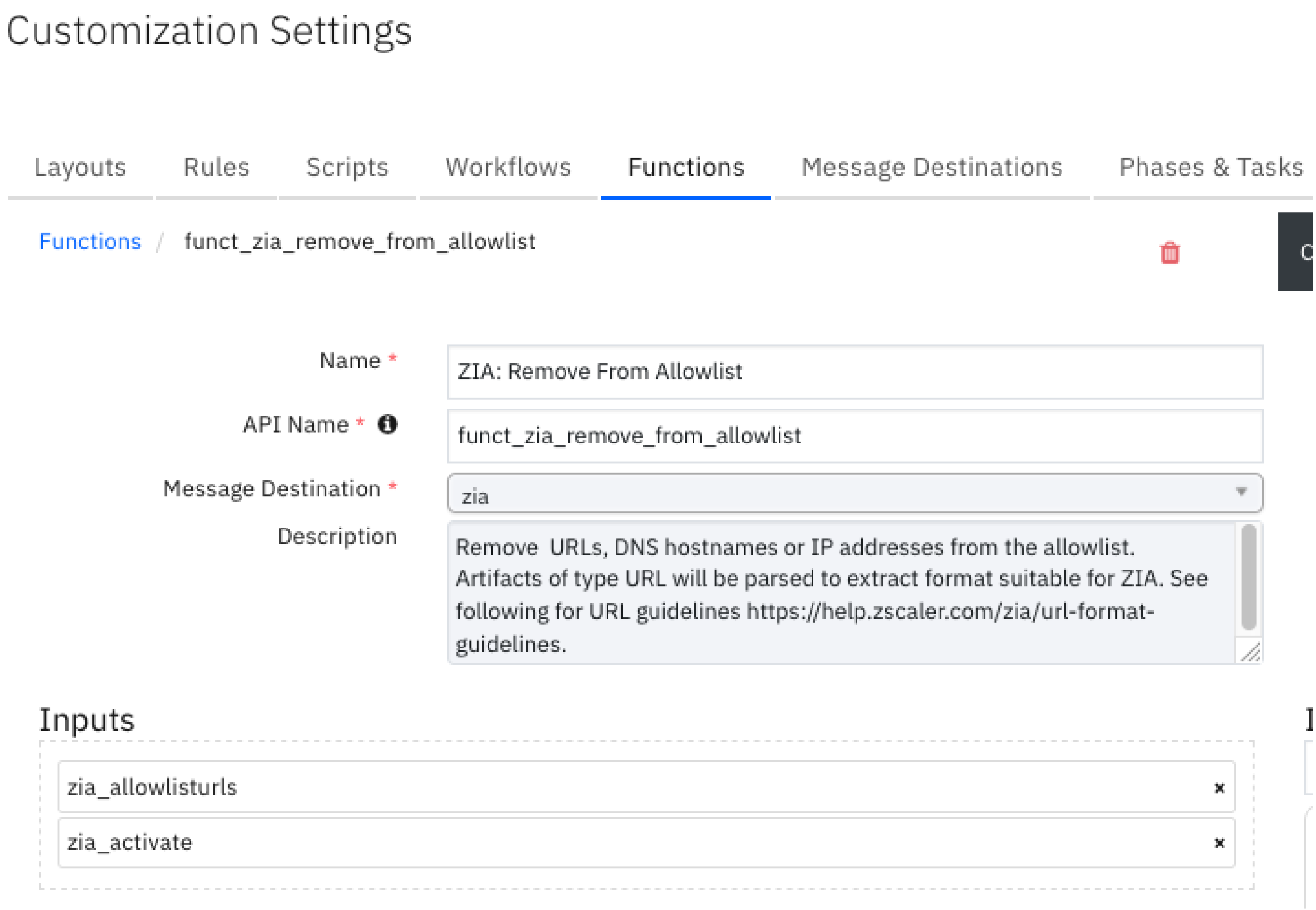
▶ Example Post-Process Script:

```
##  ZIA - wf_zia_get_allowlist post processing script ##

#  Globals

FN_NAME = "funct_zia_get_allowlist"
WF_NAME = "ZIA: Get Allowlist"
# Processing
CONTENT = results.content
INPUTS = results.inputs
QUERY_EXECUTION_DATE = results["metrics"]["timestamp"]
note_text = ''
```

```python
def main():
    note_text = u''
    url_filter = INPUTS.get("zia_url_filter")
    if CONTENT:
        allowlist_urls = CONTENT.whitelistUrls
        url_counts = CONTENT.url_counts
        note_text = u"ZIA Integration: Workflow <b>{0}</b>: There were <b>{1}</b> allowlist URLS (s) out of a total of "\
                    u"<b>{2}</b> using URL filter <b>{3}</b> returned for SOAR function <b>{4}</b>."\
        .format(WF_NAME, url_counts["filtered"], url_counts["total"], url_filter, FN_NAME)
        if allowlist_urls:
            if url_counts["filtered"] <= 50:
                note_text += u"<br>The data table <b>{0}</b> has been updated".format("Zscaler Internet Access - Allowlist")
                for url in allowlist_urls:
                    newrow = incident.addRow("zia_allowlist")
                    newrow.query_execution_date = QUERY_EXECUTION_DATE
                    newrow.allowlist_url = url
                    newrow.query_filter = url_filter
            else:
                note_text += "<br>Allow list URLS: <b>{0}</b>".format(", ".join(allowlist_urls))
    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: There were <b>no</b> results using URL filter <b>{1}</b> "\
                     u"returned for SOAR function <b>{2}</b>."\
            .format(WF_NAME, url_filter, FN_NAME)

    incident.addNote(helper.createRichText(note_text))

main()
```

Function - ZIA: Remove From Allowlist

Remove URLs, DNS hostnames or IP addresses from the allowlist. Artifacts of type URL are parsed to extract a format suitable for ZIA. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.



Customization Settings

Functions / funct_zia_remove_from_allowlist

Name * ZIA: Remove From Allowlist

API Name * funct_zia_remove_from_allowlist

Message Destination * zia

Description Remove URLs, DNS hostnames or IP addresses from the allowlist. Artifacts of type URL will be parsed to extract format suitable for ZIA. See following for URL guidelines https://help.zscaler.com/zia/url-format-guidelines.

Inputs

zia_allowlisturls

zia_activate

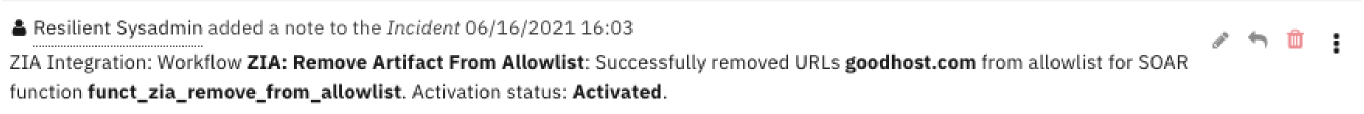
The function provides the following functionality.

- An action to remove a URL or list of URLs from the main allow list.
- A URL or list of URLs must be included in the action.
- There is an option to activate the configuration change.
- A note is added to the IBM SOAR incident with the status of the action.

An example workflow that uses this IBM SOAR function is `ZIA: Remove Artifact From Allowlist`.

The workflow is initiated by the manual artifact rule `ZIA: Remove Artifact From Allowlist`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



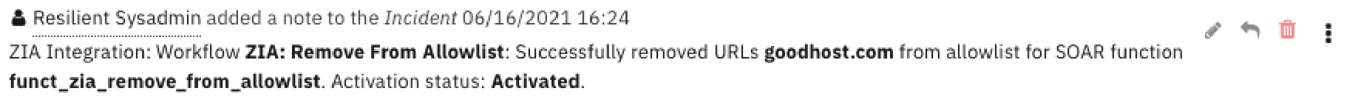
Resilient Sysadmin added a note to the *Incident* 06/16/2021 16:03

ZIA Integration: Workflow **ZIA: Remove Artifact From Allowlist**: Successfully removed URLs **goodhost.com** from allowlist for SOAR function **funct_zia_remove_from_allowlist**. Activation status: **Activated**.

Another example workflow that uses this IBM SOAR function is `ZIA: Remove From Allowlist`.

The workflow is initiated by the manual data table rule `ZIA: Remove From Allowlist` for data table `Zscaler Internet Access - Allowlist`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 16:24

ZIA Integration: Workflow **ZIA: Remove From Allowlist**: Successfully removed URLs **goodhost.com** from allowlist for SOAR function **funct_zia_remove_from_allowlist**. Activation status: **Activated**.

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_activate`	`boolean`	Yes	`-`	Activate the configuration on ZIA.
`zia_allowlisturls`	`text`	No	`-`	The URLs, URIs, DNS hostnames or IP addresses to the allowlist. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.

▶ Outputs:

```
results = {
    "content": {
        "response": {
            "whitelistUrls": [
                "1.1.1.1",
                "goodhost.com"
            ]
        },
        "activation": {
            "status": "Activated"
        }
    },
    "inputs": {
        "zia_allowlisturls": "badhost.com"
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "raw": '{"whitelistUrls": ["1.1.1.1", "goodhost.com"]}',
    "version": "1.0",
    "success": True,
    "reason": None
}
```

▶ Example Pre-Process Script:

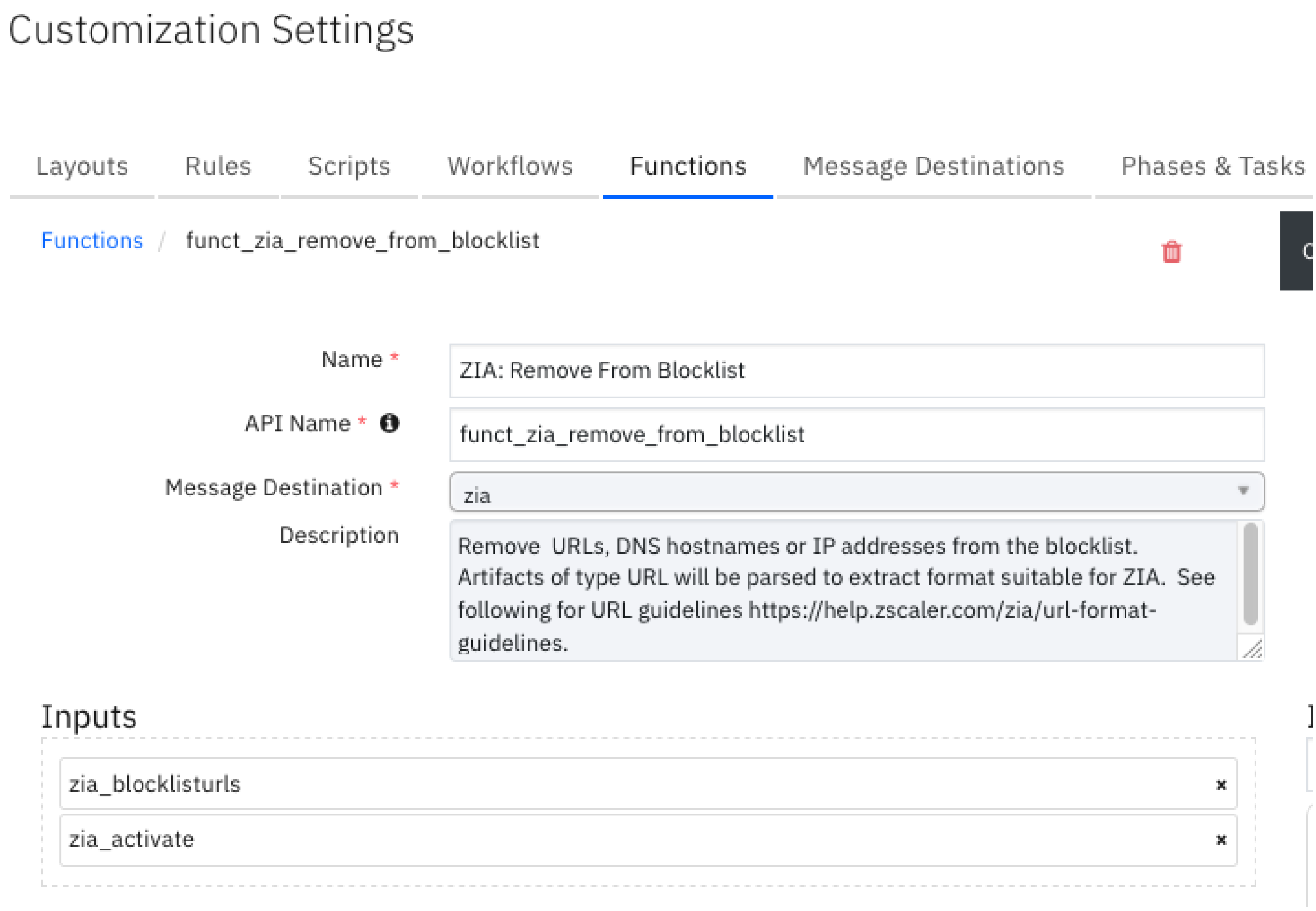
```
inputs.zia_allowlisturls = artifact.value
inputs.zia_activate = rule.properties.zia_activate
```

▶ Example Post-Process Script:

```python
##  ZIA - wf_zia_remove_artifact_from_allowlist post processing script ##
import re
#  Globals
FN_NAME = "funct_zia_remove_from_allowlist"
WF_NAME = "ZIA: Remove Artifact From Allowlist"
CONTENT = results.content
INPUTS = results.inputs

# Processing
def main():
    note_text = u''
    urls = INPUTS.get("zia_allowlisturls")
    if CONTENT:
        response = CONTENT.get("response")
        activation = CONTENT.get("activation")
        # In order to test all urls have been successfully added, convert string of urls
        # to a list and convert urls to the format used by ZIA. e.g. https://user:password@domain.com:port/index.html ->
        # domain.com:port/index.html
        allowlist_urls = [re.sub(r'^.*\/\/(.*@)*(.*)', r'\2', u) for u in re.split("\s+|,", urls)]
        updated_allowlist_urls = response.get("whitelistUrls")
        if not any(a in updated_allowlist_urls for a in allowlist_urls):
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Successfully removed URLs <b>{1}</b> from allowlist " \
                        u"for SOAR function <b>{2}</b>.".format(WF_NAME, urls, FN_NAME)
            note_text += u" Activation status: <b>{0}</b>.".format(activation["status"])
        else:
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Not all urls removed while attempting " \
                        u"to remove URLs <b>{1}</b> from allowlist by SOAR function <b>{2}</b>."\
                .format(WF_NAME, urls, FN_NAME)
    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: There was <b>no</b> result returned while attempting " \
                     u"to remove URLs <b>{1}</b> from allowlist for SOAR function <b>{2}</b>."\
            .format(WF_NAME, urls, FN_NAME)

    incident.addNote(helper.createRichText(note_text))

main()
```

Function - ZIA: Remove From Blocklist

Remove URLs, DNS hostnames or IP addresses from the blocklist. Artifacts of type URL are parsed to extract a format suitable for ZIA. See the following for URL guidelines: https://help.zscaler.com/zia/url-format-guidelines.



Name * ZIA: Remove From Blocklist

API Name * funct_zia_remove_from_blocklist

Message Destination * zia

Description Remove URLs, DNS hostnames or IP addresses from the blocklist. Artifacts of type URL will be parsed to extract format suitable for ZIA. See following for URL guidelines https://help.zscaler.com/zia/url-format-guidelines.

Inputs

zia_blocklisturls

zia_activate

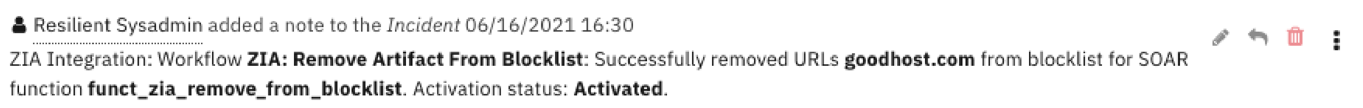
The function provides the following functionality.

- An action to remove a URL or list of URLs from the main block list.
- A URL or list of URLs must be included in the action.
- There is an option to activate the configuration change.
- A note is added to the IBM SOAR incident with the status of the action.

An example workflow that uses this IBM SOAR function is `Remove Artifact From Blocklist`.

The workflow is initiated by the manual artifact rule `Remove Artifact From Blocklist`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



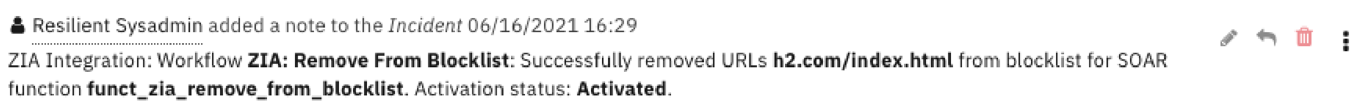
Resilient Sysadmin added a note to the *Incident* 06/16/2021 16:30

ZIA Integration: Workflow **ZIA: Remove Artifact From Blocklist**: Successfully removed URLs **goodhost.com** from blocklist for SOAR function **funct_zia_remove_from_blocklist**. Activation status: **Activated**.

Another example workflow that uses this IBM SOAR function is `Remove From Blocklist`.

The workflow is initiated by the manual data table rule `Remove From Blocklist` for data table `Zscaler Internet Access - Blocklist`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 16:29

ZIA Integration: Workflow **ZIA: Remove From Blocklist**: Successfully removed URLs **h2.com/index.html** from blocklist for SOAR function **funct_zia_remove_from_blocklist**. Activation status: **Activated**.

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_activate`	`boolean`	Yes	`-`	Activate the configuration on ZIA.
`zia_blocklisturls`	`text`	No	`-`	The URLs, URIs, DNS hostnames or IP addresses to add to the blocklist. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.

▶ Outputs:

```
results = {
    "content": {
        "response": {
            "status": "OK"
        },
        "activation": {
            "status": "Activated"
        }
    },
    "inputs": {
        "zia_blocklisturls": "1.1.1.1"
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "raw": "{'status': 'OK'}",
    "version": "1.0",
    "success": True,
    "reason": None
}
```

▶ Example Pre-Process Script:

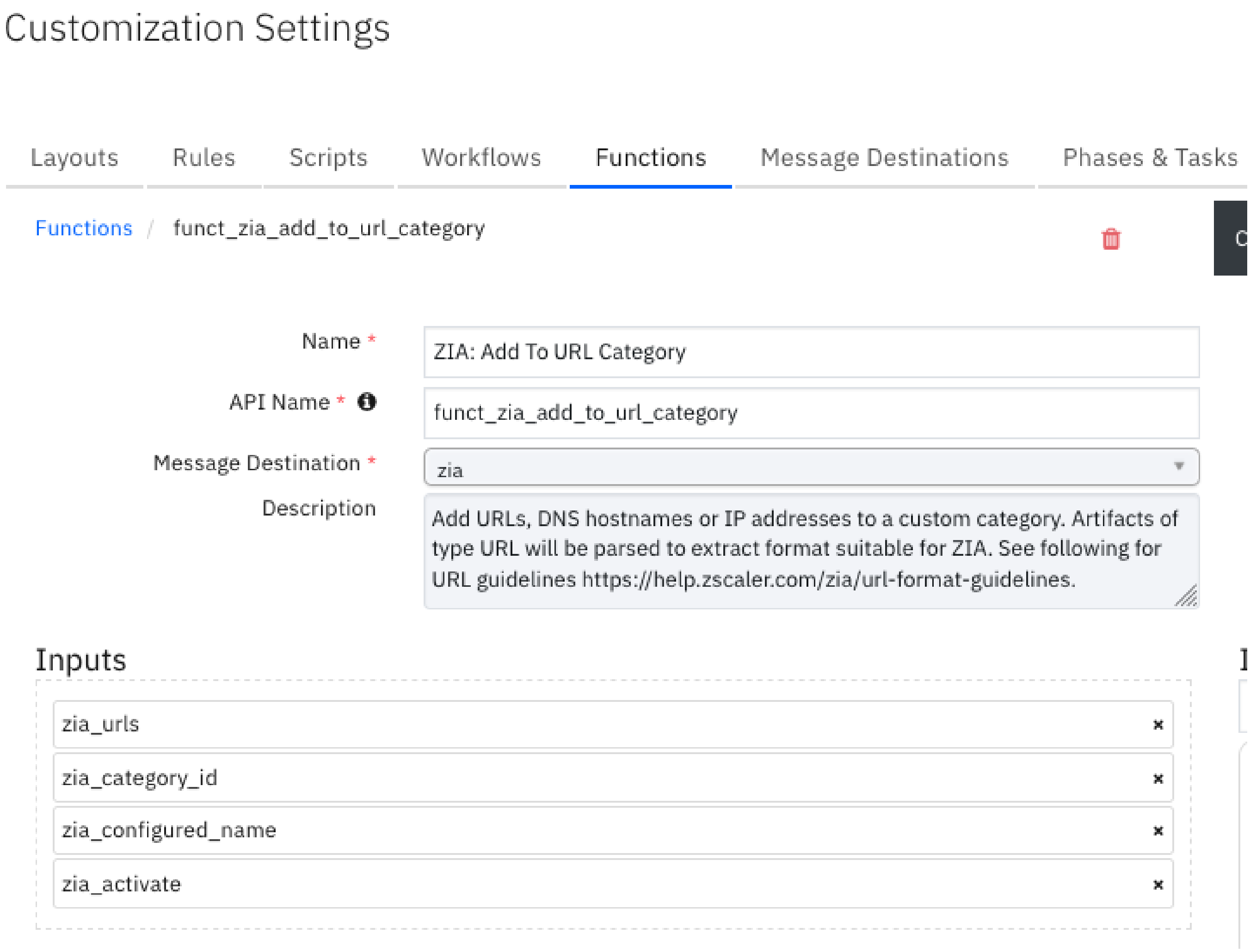
```
inputs.zia_blocklisturls = artifact.value
inputs.zia_activate = rule.properties.zia_activate
```

▶ Example Post-Process Script:

```
##  ZIA - wf_zia_remove_artifact_from_blocklist post processing script ##

#  Globals
FN_NAME = "funct_zia_remove_from_blocklist"
WF_NAME = "ZIA: Remove Artifact From Blocklist"
CONTENT = results.content
INPUTS = results.inputs

# Processing
def main():
    note_text = u''
    urls = INPUTS.get("zia_blocklisturls")
    if CONTENT:
        response = CONTENT.get("response")
        activation = CONTENT.get("activation")
        status = response.get("status")
        if status == "OK":
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Successfully removed URLs <b>{1}</b> from blocklist " \
                        u"for SOAR function <b>{2}</b>.".format(WF_NAME, urls, FN_NAME)
            note_text += u" Activation status: <b>{0}</b>.".format(activation["status"])
        else:
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Unexpected status <b>{1}</b> returned while attempting " \
                        u"to remove URLs <b>{2}</b> from blocklist by SOAR function <b>{3}</b>."\
                .format(WF_NAME, status, urls, FN_NAME)
    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: There was <b>no</b> result returned while attempting " \
                     u"to remove URLs <b>{1}</b> from blocklist for SOAR function <b>{2}</b>."\
            .format(WF_NAME, urls, FN_NAME)

    incident.addNote(helper.createRichText(note_text))

main()
```

Function - ZIA: Add To URL Category

Add URLs, DNS hostnames or IP addresses to a custom category. Artifacts of type URL are parsed to extract a format suitable for ZIA. See the following for URL guidelines: https://help.zscaler.com/zia/url-format-guidelines.



Customization Settings

Name * ZIA: Add To URL Category

API Name * funct_zia_add_to_url_category

Message Destination * zia

Description Add URLs, DNS hostnames or IP addresses to a custom category. Artifacts of type URL will be parsed to extract format suitable for ZIA. See following for URL guidelines https://help.zscaler.com/zia/url-format-guidelines.

Inputs

zia_urls

zia_category_id

zia_configured_name

zia_activate

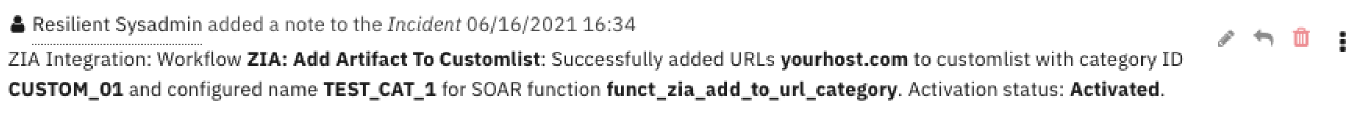
The function provides the following functionality.

- An action to add a URL or list of URLs to a custom URL category in the ZIA environment.
- A URL or list of URLs must be included in the action.
- A custom category name and category ID must be included in the action.
- In the example workflows, the category name is set from a user defined drop-down list.
- In the example workflows, the category ID is obtained by querying ZIA by category name.
- There is an option to activate the configuration change.
- A note is added to the IBM SOAR incident with the status of the action.

An example workflow that uses this IBM SOAR function is `ZIA: Add Artifact To Customlist`.

The workflow is initiated by the manual artifact rule `ZIA: Add Artifact To Customlist`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



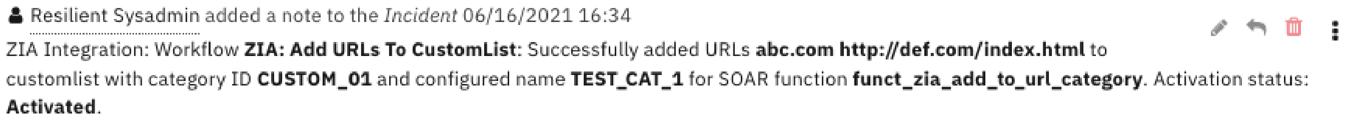
Resilient Sysadmin added a note to the *Incident* 06/16/2021 16:34

ZIA Integration: Workflow **ZIA: Add Artifact To Customlist**: Successfully added URLs **yourhost.com** to customlist with category ID **CUSTOM_01** and configured name **TEST_CAT_1** for SOAR function **funct_zia_add_to_url_category**. Activation status: **Activated**.

Another example workflow that uses this IBM SOAR function is `ZIA: Add URLs To CustomList`.

The workflow is initiated by the manual incident rule `ZIA: Add URLs To CustomList`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 16:34

ZIA Integration: Workflow **ZIA: Add URLs To CustomList**: Successfully added URLs **abc.com http://def.com/index.html** to customlist with category ID **CUSTOM_01** and configured name **TEST_CAT_1** for SOAR function **funct_zia_add_to_url_category**. Activation status: **Activated**.

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_activate`	`boolean`	Yes	`-`	Activate the configuration on ZIA.
`zia_category_id`	`text`	Yes	`-`	Category ID of a URL category.
`zia_configured_name`	`text`	Yes	`-`	Configured name of a custom URL category.
`zia_urls`	`text`	Yes	`-`	Entries of type DNS hostname, IP Address, URL or URI. Wildcard is a period (.). Entries are parsed to extract a format suitable for ZIA to add to a list. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.

▶ Outputs:

```
results = {
    "content": {
        "response": {
            "id": "CUSTOM_01",
            "configuredName": "TEST_CAT_1",
            "keywordsRetainingParentCategory": [],
            "urls": [
                "192.168.1.1",
                "testhost.com"
            ],
            "dbCategorizedUrls": [],
            "customCategory": True,
            "editable": True,
            "description": "CUSTOM_01_DESC",
            "type": "URL_CATEGORY",
            "val": 128,
            "customUrlsCount": 2,
            "urlsRetainingParentCategoryCount": 0
        },
        "activation": {
            "status": "Activated"
        }
    },
    "inputs": {
        "zia_configured_name": "TEST_CAT_1",
        "zia_urls": "192.168.1.1",
        "zia_category_id": "CUSTOM_01"
    },
    "raw": '{"id": "CUSTOM_01", "configuredName": "TEST_CAT_1",
"keywordsRetainingParentCategory": [], "urls": ["192.168.1.1",
"testhost.com"], "dbCategorizedUrls": [], "customCategory": true,
"editable": true, "description": "CUSTOM_01_DESC", "type": "URL_CATEGORY",
"val": 128, "customUrlsCount": 2, "urlsRetainingParentCategoryCount": 0}',
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "version": "1.0",
    "success": True,
    "reason": None
}
```

▶ Example Pre-Process Script:

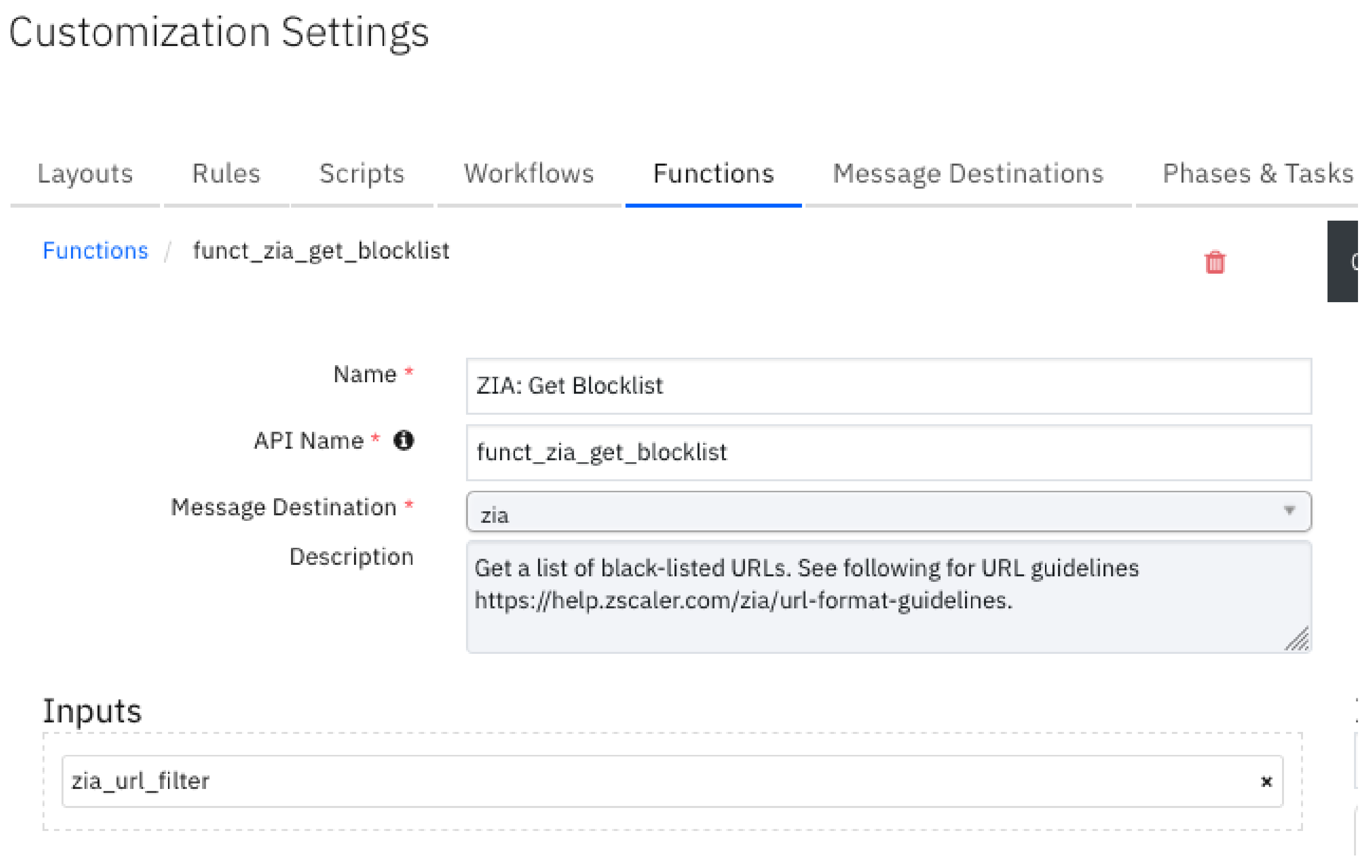
```
content = workflow.properties.get_categories_results.content
configured_name = rule.properties.zia_configured_name
cats = content.get("categories")
inputs.zia_category_id = [c["id"] for c in cats if configured_name ==
c["configuredName"]][0]
inputs.zia_configured_name = configured_name
inputs.zia_urls = artifact.value
inputs.zia_activate = rule.properties.zia_activate
```

▶ Example Post-Process Script:

```
##  ZIA - wf_zia_add_to_customlist post processing script ##
import re
#  Globals
FN_NAME = "funct_zia_add_to_url_category"
WF_NAME = "ZIA: Add Artifact To Customlist"
CONTENT = results.content
INPUTS = results.inputs

# Processing
def main():
    note_text = u''
    urls = INPUTS.get("zia_urls")
    category_id = INPUTS.get("zia_category_id")
    configured_name = INPUTS.get("zia_configured_name")
    if CONTENT:
        response = CONTENT.get("response")
        activation = CONTENT.get("activation")
        # In order to test all urls have been successfully added, convert string of urls
        # to a list and convert urls to the format used by ZIA. e.g. https://user:password@domain.com:port/index.html ->
        # domain.com:port/index.html
        customlist_urls = [re.sub(r'^.*\/\/(.*@)*(.*)', r'\2', u) for u in re.split("\s+|,", urls)]
        updated_customlist = response.get("urls")
        if all(a in updated_customlist for a in customlist_urls):
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Successfully added URLs <b>{1}</b> to customlist "\
                        u"with category ID <b>{2}</b> and configured name <b>{3}</b> for SOAR function <b>{4}</b>."\
                        .format(WF_NAME, urls, category_id, configured_name, FN_NAME)
            note_text += u" Activation status: <b>{0}</b>.".format(activation["status"])
        else:
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Not all urls added while attempting "\
                        u"to add URLs <b>{1}</b> to customlist with category ID <b>{2}</b> and configured name <b>{3}</b> "\
                        u"for SOAR function <b>{4}</b>.".format(WF_NAME, urls, category_id,  FN_NAME)
    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: There was <b>no</b> result returned while attempting "\
                     u"to add URLs <b>{1}</b> to to customlist of category ID <b>{2}</b> for SOAR function <b>{2}</b>."\
            .format(WF_NAME, urls, category_id,  FN_NAME)

    incident.addNote(helper.createRichText(note_text))

main()
```

Function - ZIA: Get Blocklist

Get a list of block-listed URLs. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.



Name * ZIA: Get Blocklist

API Name * funct_zia_get_blocklist

Message Destination * zia

Description Get a list of black-listed URLs. See following for URL guidelines https://help.zscaler.com/zia/url-format-guidelines.

Inputs

zia_url_filter

The function provides the following functionality.

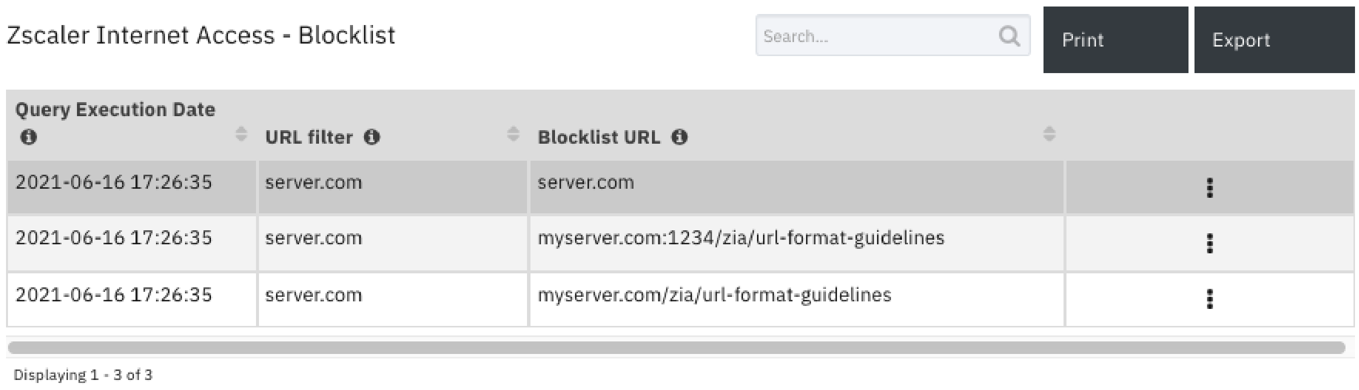
- Query ZIA for information on the main block list.
- The result can be filtered by URL values using regular expressions.
- A data table is updated.
- A note is added to the IBM SOAR incident with the status of the query.
- If the URL count is > 50, the raw JSON is included in the note and the data table is not updated.

An example workflow that uses this IBM SOAR function is `Get Blocklist`.

The workflow is initiated by the manual incident rule `Get Blocklist`.

The data table `Zscaler Internet Access - Blocklist` is updated.

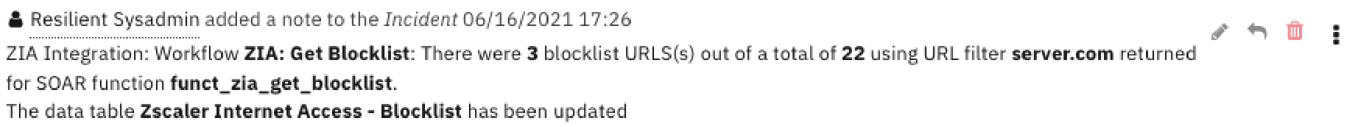
The following screenshot shows an example of data table updated by the function:



Query Execution Date	URL filter	Blocklist URL	
2021-06-16 17:26:35	server.com	server.com	⋮
2021-06-16 17:26:35	server.com	myserver.com:1234/zia/url-format-guidelines	⋮
2021-06-16 17:26:35	server.com	myserver.com/zia/url-format-guidelines	⋮

Displaying 1 - 3 of 3

The following screenshFot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 17:26
ZIA Integration: Workflow **ZIA: Get Blocklist**: There were **3** blocklist URLS(s) out of a total of **22** using URL filter **server.com** returned for SOAR function **funct_zia_get_blocklist**.
The data table **Zscaler Internet Access - Blocklist** has been updated

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_url_filter`	`text`	No	`<REGEX>|<STRING>`	Filter by URL. Can be a string or regular expression. e.g. host.ibm.com, '^host.*' or ^(?:[0-9]{1,3}.){3}[0-9]{1,3}$. Leaving the filter blank will return the full list.

▶ Outputs:

```
results = {
    "content": {
        "blacklistUrls": [
            "badhost.com",
            "192.168.12.2"
        ],
        "url_counts": {
            "total": 3,
            "filtered": 3
        }
    },
    "inputs": {
        "zia_url_filter": ".*"
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "raw": "{'blacklistUrls': ['badhost.com'], ['192.168.12.2']}",
    "version": "1.0",
    "success": True,
    "reason": None,
}
```

▶ Example Pre-Process Script:

```
##  ZIA - wf_zia_get_blocklist pre processing script ##
import re

URL_FILTER = rule.properties.zia_url_filter
```

```
def is_regex(regex_str):
    """"Test if sting is a correctly formed regular expression.

    :param regex_str: Regular expression string.
    :return: Boolean.
    """
    try:
        re.compile(regex_str)
        return True
    except (re.error, TypeError):
        return False


def main():
    # Test filter to ensure it is a valid regular expressions.
    if URL_FILTER and not is_regex(URL_FILTER):
        raise ValueError("The query filter '{}' is not a valid regular
expression.".format(unicode(URL_FILTER)))

    inputs.zia_url_filter = rule.properties.zia_url_filter

if __name__ == "__main__":
    main()
```

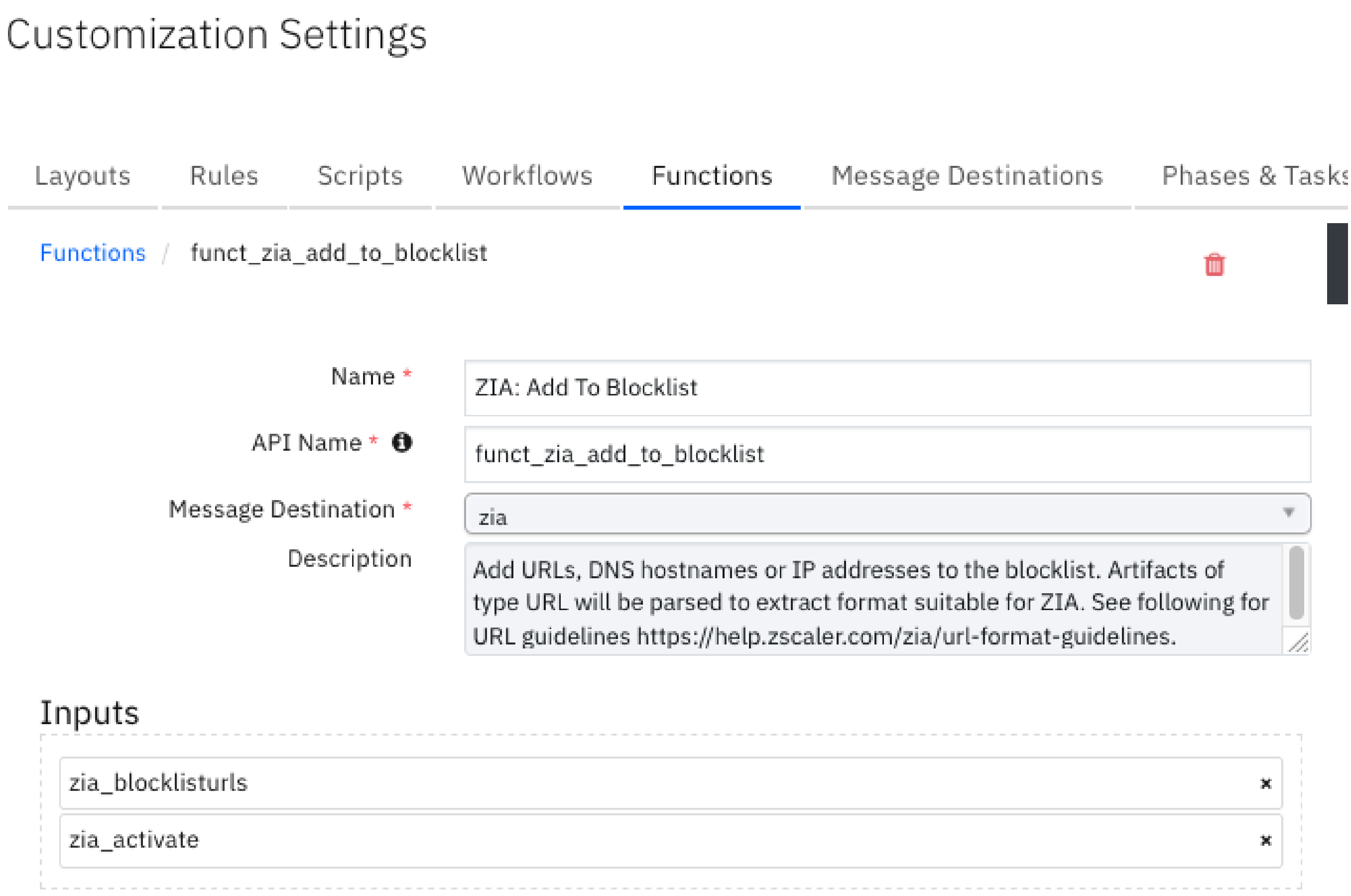
▶ Example Post-Process Script:

```

##  ZIA - wf_zia_get_blocklist post processing script ##

#  Globals
FN_NAME = "funct_zia_get_blocklist"
WF_NAME = "ZIA: Get Blocklist"
CONTENT = results.content
INPUTS = results.inputs
QUERY_EXECUTION_DATE = results["metrics"]["timestamp"]

# Processing
def main():
    note_text = u''
    url_filter = INPUTS.get("zia_url_filter")
    if CONTENT:
        blocklist_urls = CONTENT.blacklistUrls
        url_counts = CONTENT.url_counts
        note_text = u"ZIA Integration: Workflow <b>{0}</b>: There were <b>
{1}</b> blocklist URLS(s) out of a total of "\
                    u"<b>{2}</b> using URL filter <b>{3}</b> returned for
SOAR function <b>{4}</b>."\
        .format(WF_NAME, url_counts["filtered"], url_counts["total"],
url_filter, FN_NAME)
        if blocklist_urls:
            if url_counts["filtered"] <= 50:
                note_text += u"<br>The data table <b>{0}</b> has been
updated".format("Zscaler Internet Access - Blocklist")
```

```
                for url in blocklist_urls:
                    newrow = incident.addRow("zia_blocklist")
                    newrow.query_execution_date = QUERY_EXECUTION_DATE
                    newrow.blocklist_url = url
                    newrow.query_filter = url_filter
            else:
                note_text += "<br>Blocklisted URLS: <b>{0}</b>".format(",
".join(blocklist_urls))
    else:
        note_text += u"ZIA Integration: Workflow <b>{0}</b>: There were
<b>no</b> results using URL filter <b>{1}</b> "\
                     u"returned for SOAR function <b>{2}</b>."\
            .format(WF_NAME, url_filter, FN_NAME)

    incident.addNote(helper.createRichText(note_text))

main()
```

Function - ZIA: Add To Blocklist

Add URLs, DNS hostnames or IP addresses to the blocklist. Artifacts of type URL are parsed to extract a format suitable for ZIA. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.



Name *	ZIA: Add To Blocklist
API Name *	funct_zia_add_to_blocklist
Message Destination *	zia
Description	Add URLs, DNS hostnames or IP addresses to the blocklist. Artifacts of type URL will be parsed to extract format suitable for ZIA. See following for URL guidelines https://help.zscaler.com/zia/url-format-guidelines.

Inputs

- zia_blocklisturls
- zia_activate

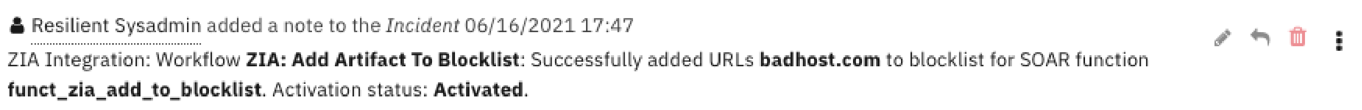
The function provides the following functionality.

- An action to add a URL or list of URLs to the main block list.
- A URL or list of URLs must be included in the action.
- There is an option to activate the configuration change.
- A note is added to the IBM SOAR incident with the status of the action.

An example workflow that uses this IBM SOAR function is `Add Artifact To Blocklist`.

The workflow is initiated by the manual artifact rule `Add Artifact To Blocklist`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



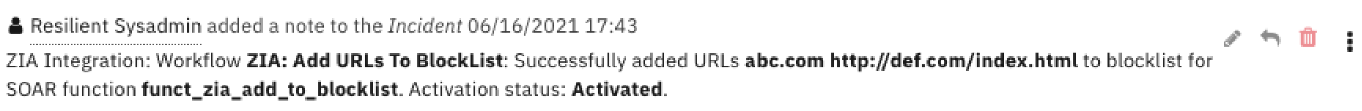
Resilient Sysadmin added a note to the *Incident* 06/16/2021 17:47

ZIA Integration: Workflow **ZIA: Add Artifact To Blocklist**: Successfully added URLs **badhost.com** to blocklist for SOAR function **funct_zia_add_to_blocklist**. Activation status: **Activated**.

Another example workflow that uses this IBM SOAR function is `ZIA: Add URLs To BlockList`.

The workflow is initiated by the manual incident rule `ZIA: Add URLs To BlockList`.

The following screenshot shows an example of a note added to an IBM SOAR incident:



Resilient Sysadmin added a note to the *Incident* 06/16/2021 17:43

ZIA Integration: Workflow **ZIA: Add URLs To BlockList**: Successfully added URLs **abc.com http://def.com/index.html** to blocklist for SOAR function **funct_zia_add_to_blocklist**. Activation status: **Activated**.

▶ Inputs:

Name	Type	Required	Example	Tooltip
`zia_activate`	`boolean`	Yes	`-`	Activate the configuration on ZIA.
`zia_blocklisturls`	`text`	No	`-`	The URLs, URIs, DNS hostnames or IP addresses to add to the blocklist. See the URL guidelines at https://help.zscaler.com/zia/url-format-guidelines.

▶ Outputs:

```
results = {
    "content": {
        "response": {
            "status": "OK"
        },
        "activation": {
            "status": "Activated"
        }
    },
    "inputs": {
        "zia_blocklisturls": "1.1.1.1"
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-zia",
        "package_version": "1.0.0",
        "host": "host.ibm.com",
        "execution_time_ms": 22199,
        "timestamp": "2021-05-11 10:33:14"
    },
    "raw": "{'status': 'OK'}",
    "version": "1.0",
    "success": True,
    "reason": None
}
```

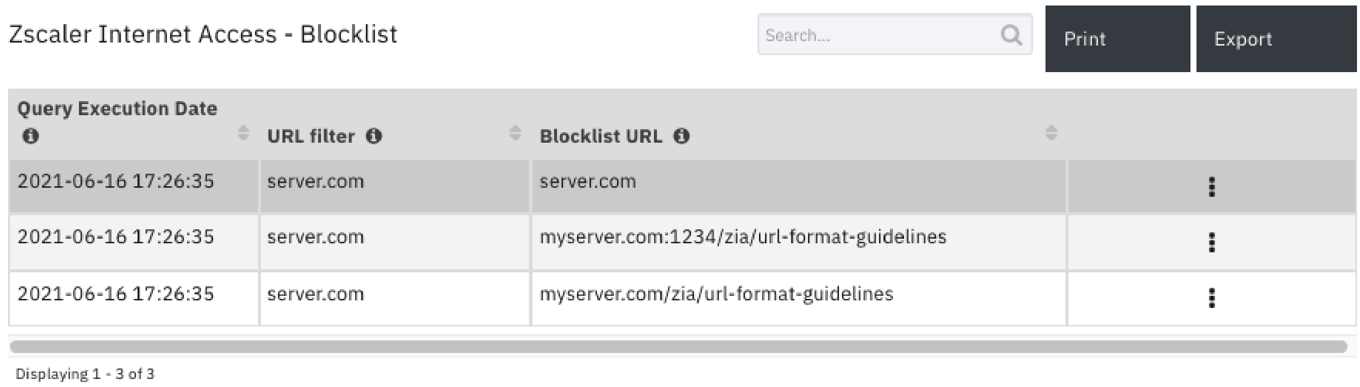
▶ Example Pre-Process Script:

```
inputs.zia_blocklisturls = artifact.value
inputs.zia_activate = rule.properties.zia_activate
```

▶ Example Post-Process Script:

```
##  ZIA - wf_zia_add_artifact_to_blocklist post processing script ##

#  Globals
FN_NAME = "funct_zia_add_to_blocklist"
WF_NAME = "ZIA: Add Artifact To Blocklist"
CONTENT = results.content
INPUTS = results.inputs

# Processing
def main():
    note_text = u''
    urls = INPUTS.get("zia_blocklisturls")
    if CONTENT:
        response = CONTENT.get("response")
        activation = CONTENT.get("activation")
        status = response.get("status")
        if status == "OK":
            note_text = u"ZIA Integration: Workflow <b>{0}</b>:
Successfully added URLs <b>{1}</b> to blocklist " \
                        u"for SOAR function <b>{2}</b>.".format(WF_NAME,
urls, FN_NAME)
            note_text += u" Activation status: <b>{0}
</b>.".format(activation["status"])
        else:
            note_text = u"ZIA Integration: Workflow <b>{0}</b>: Unexpected
status <b>{1}</b> returned while attempting " \
                        u"to add URLs <b>{2}</b> to blocklist for SOAR
function <b>{2}</b>."\
                .format(WF_NAME, urls, FN_NAME)
    else:
            note_text += u"ZIA Integration: Workflow <b>{0}</b>: There was
<b>no</b> result returned while attempting " \
                         u"to add URLs <b>{1}</b> to blocklist for SOAR
function <b>{2}</b>."\
                .format(WF_NAME, urls, FN_NAME)

    incident.addNote(helper.createRichText(note_text))

main()
```

Data Table - Zscaler Internet Access - Blocklist



Query Execution Date	URL filter	Blocklist URL	
2021-06-16 17:26:35	server.com	server.com	⋮
2021-06-16 17:26:35	server.com	myserver.com:1234/zia/url-format-guidelines	⋮
2021-06-16 17:26:35	server.com	myserver.com/zia/url-format-guidelines	⋮

Displaying 1 - 3 of 3

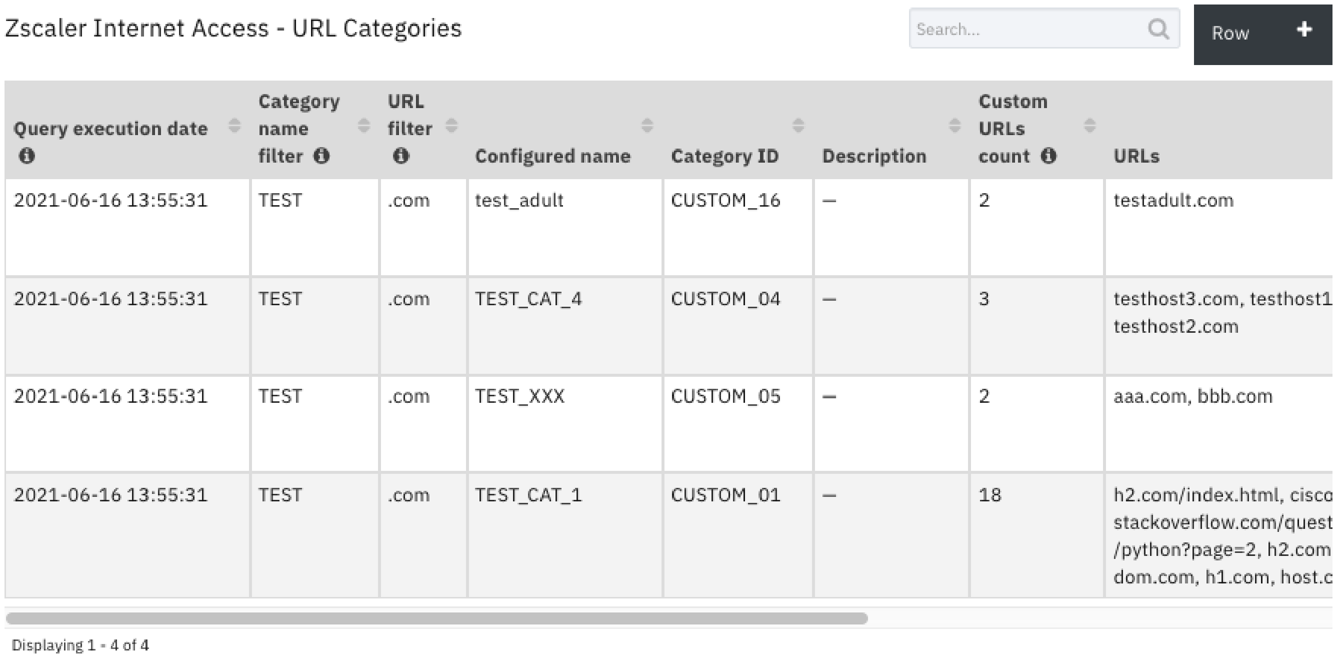
API Name:

zia_blocklist

Columns:

Column Name	API Access Name	Type	Tooltip
Blocklist URL	`blocklist_url`	`text`	A URL which is on the blocklist.
Query Execution Date	`query_execution_date`	`text`	Date and time when query was run.
Query filter	`query_filter`	`text`	Filter, if any, used in query.

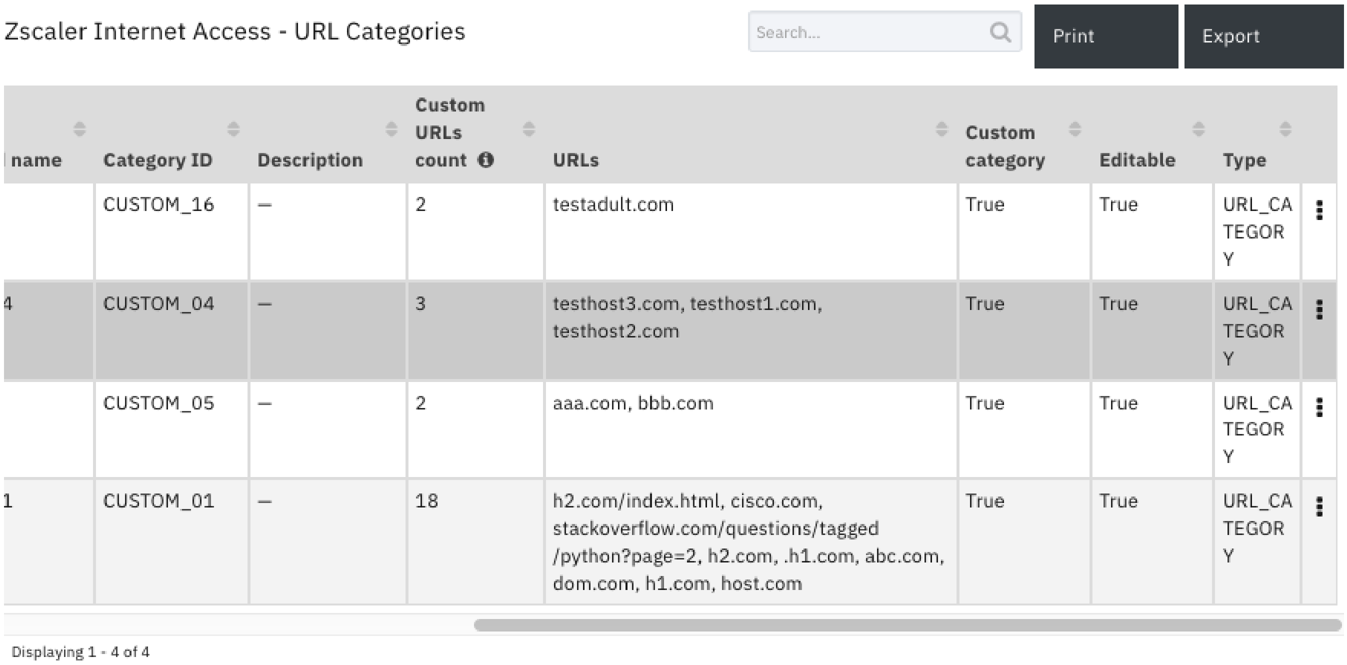
Data Table - Zscaler Internet Access - URL Categories



Zscaler Internet Access - URL Categories

Query execution date	Category name filter	URL filter	Configured name	Category ID	Description	Custom URLs count	URLs
2021-06-16 13:55:31	TEST	.com	test_adult	CUSTOM_16	—	2	testadult.com
2021-06-16 13:55:31	TEST	.com	TEST_CAT_4	CUSTOM_04	—	3	testhost3.com, testhost1 testhost2.com
2021-06-16 13:55:31	TEST	.com	TEST_XXX	CUSTOM_05	—	2	aaa.com, bbb.com
2021-06-16 13:55:31	TEST	.com	TEST_CAT_1	CUSTOM_01	—	18	h2.com/index.html, cisco stackoverflow.com/quest /python?page=2, h2.com dom.com, h1.com, host.c

Displaying 1 - 4 of 4



Zscaler Internet Access - URL Categories

name	Category ID	Description	Custom URLs count	URLs	Custom category	Editable	Type	
	CUSTOM_16	—	2	testadult.com	True	True	URL_CATEGORY	⋮
4	CUSTOM_04	—	3	testhost3.com, testhost1.com, testhost2.com	True	True	URL_CATEGORY	⋮
	CUSTOM_05	—	2	aaa.com, bbb.com	True	True	URL_CATEGORY	⋮
1	CUSTOM_01	—	18	h2.com/index.html, cisco.com, stackoverflow.com/questions/tagged /python?page=2, h2.com, .h1.com, abc.com, dom.com, h1.com, host.com	True	True	URL_CATEGORY	⋮

Displaying 1 - 4 of 4

API Name:

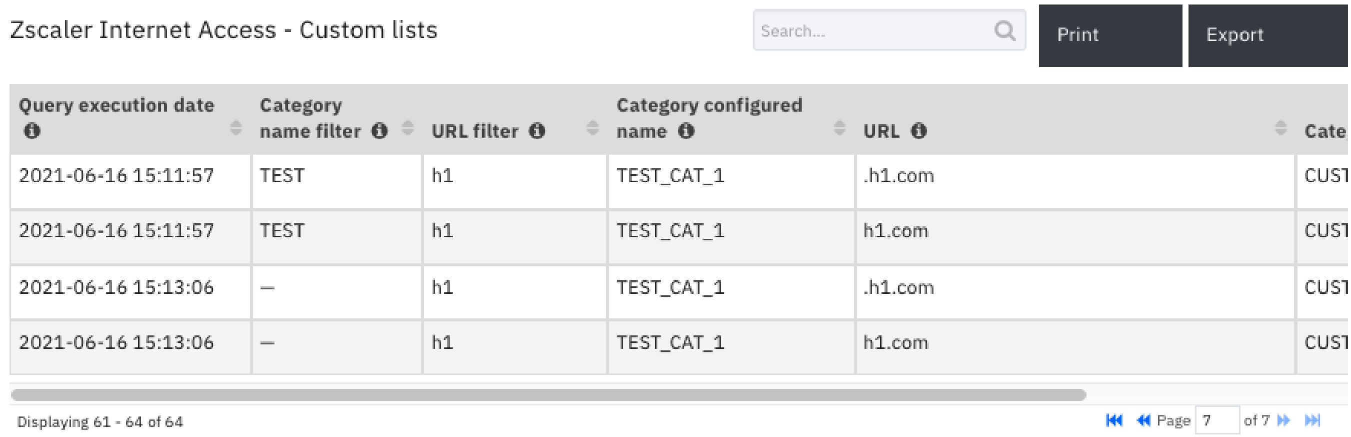
zia_url_categories

Columns:

Column Name	API Access Name	Type	Tooltip
Description	`cat_description`	`text`	-
Category ID	`cat_id`	`text`	-
Configured name	`configuredName`	`text`	-
Custom category	`customCategory`	`text`	-

Column Name	API Access Name	Type	Tooltip
Custom URLs count	`customUrlsCount`	`text`	Total custom URL count before any results filtering.
Editable	`editable`	`text`	-
Category name filter	`name_filter`	`text`	Configured name filter, if any, used in the query.
Query execution date	`query_execution_date`	`text`	Date and time when query was run.
Type	`type`	`text`	-
URL filter	`url_filter`	`text`	URL filter, if any, used in the query.
URLs	`urls`	`text`	-

Data Table - Zscaler Internet Access - Custom lists



Zscaler Internet Access - Custom lists

Query execution date	Category name filter	URL filter	Category configured name	URL	Cate
2021-06-16 15:11:57	TEST	h1	TEST_CAT_1	.h1.com	CUST
2021-06-16 15:11:57	TEST	h1	TEST_CAT_1	h1.com	CUST
2021-06-16 15:13:06	—	h1	TEST_CAT_1	.h1.com	CUST
2021-06-16 15:13:06	—	h1	TEST_CAT_1	h1.com	CUST

Displaying 61 - 64 of 64

API Name:

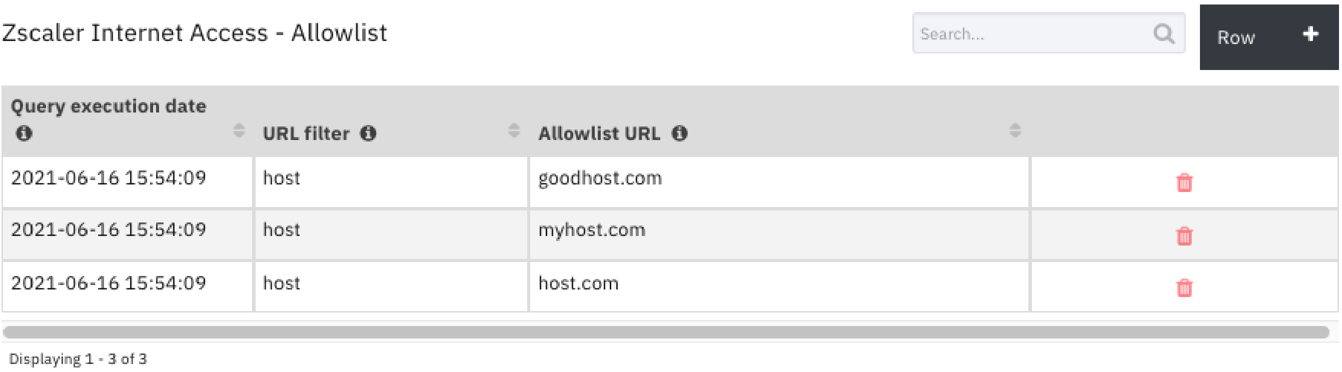
zia_customlists

Columns:

Column Name	API Access Name	Type	Tooltip
Category ID	`cat_id`	`text`	Category ID of custom list.
Category configured name	`configuredName`	`text`	configured name of the custom list.
Category name filter	`name_filter`	`text`	Configured name filter, if any, used in the query.
Query execution date	`query_execution_date`	`text`	Date and time when query was run.
URL	`url`	`text`	A URL on the custom list.

Column Name	API Access Name	Type	Tooltip
URL filter	`url_filter`	`text`	URL filter, if any, used in the query.

Data Table - Zscaler Internet Access - Allowlist



Query execution date	URL filter	Allowlist URL	
2021-06-16 15:54:09	host	goodhost.com	🗑
2021-06-16 15:54:09	host	myhost.com	🗑
2021-06-16 15:54:09	host	host.com	🗑

Displaying 1 - 3 of 3

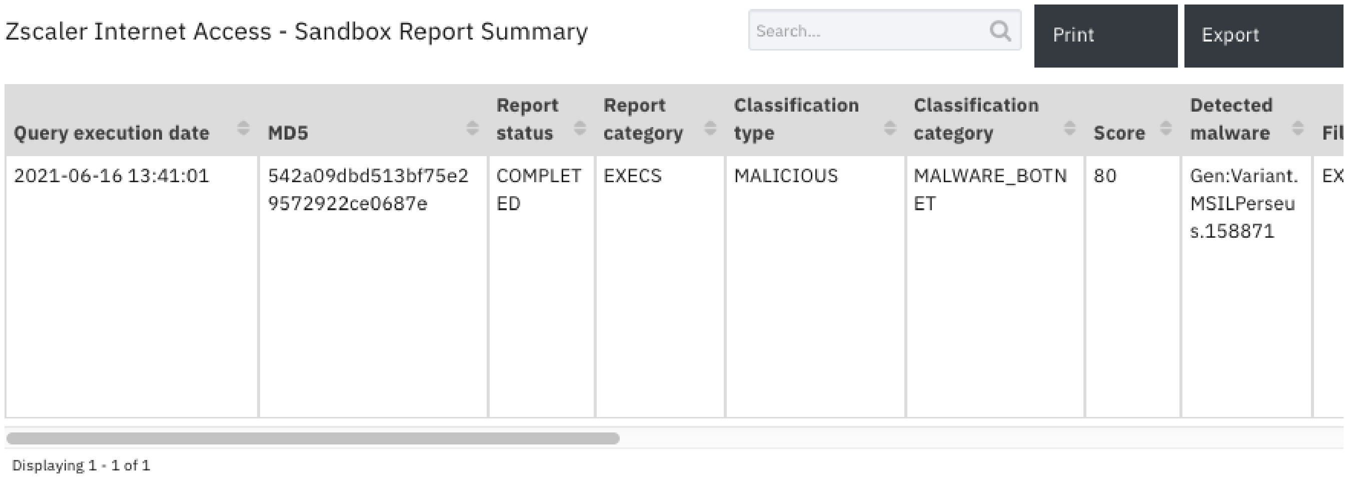
API Name:

zia_allowlist

Columns:

Column Name	API Access Name	Type	Tooltip
Allowlist URL	`allowlist_url`	`text`	A URL which is on the allowlist
Query execution date	`query_execution_date`	`text`	Date and time when query was run.
Query filter	`query_filter`	`text`	Filter, if any, used in query.

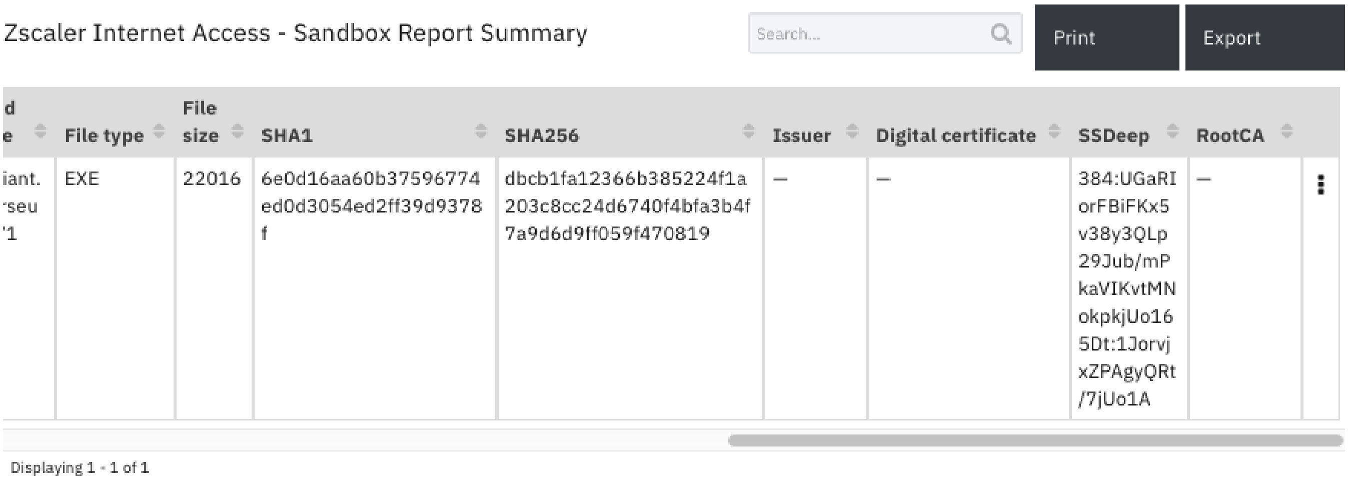
Data Table - Zscaler Internet Access - Sandbox Report Summary



Zscaler Internet Access - Sandbox Report Summary

Query execution date	MD5	Report status	Report category	Classification type	Classification category	Score	Detected malware	Fil
2021-06-16 13:41:01	542a09dbd513bf75e29572922ce0687e	COMPLETED	EXECS	MALICIOUS	MALWARE_BOTNET	80	Gen:Variant.MSILPerseus.158871	EX

Displaying 1 - 1 of 1



Zscaler Internet Access - Sandbox Report Summary

d e	File type	File size	SHA1	SHA256	Issuer	Digital certificate	SSDeep	RootCA
iant. ·seu ·1	EXE	22016	6e0d16aa60b37596774ed0d3054ed2ff39d9378f	dbcb1fa12366b385224f1a203c8cc24d6740f4bfa3b4f7a9d6d9ff059f470819	–	–	384:UGaRIorFBiFKx5v38y3QLp29Jub/mPkaVIKvtMNokpkjUo165Dt:1JorvjxZPAgyQRt/7jUo1A	–

Displaying 1 - 1 of 1

API Name:

zia_sandbox_report_summary

Columns:

Column Name	API Access Name	Type	Tooltip
Classification category	`Category`	`text`	-
Detected malware	`DetectedMalware`	`text`	-
Digital certificate	`DigitalCerificate`	`text`	-
File size	`FileSize`	`text`	-
File type	`FileType`	`text`	-
Issuer	`Issuer`	`text`	-
MD5	`MD5`	`text`	-
RootCA	`RootCA`	`text`	-
SHA1	`SHA1`	`text`	-
SSDeep	`SSDeep`	`text`	-
Score	`Score`	`text`	-

Column Name	API Access Name	Type	Tooltip
SHA256	`Sha256`	`text`	-
Report status	`Status`	`text`	-
Classification type	`Type`	`text`	-
Query execution date	`query_execution_date`	`text`	-
Report category	`report_Category`	`text`	-

Rules

Rule Name	Object	Workflow Triggered
ZIA: Add URLs To AllowList	incident	`wf_zia_add_urls_to_allowlist`
ZIA: Remove From Allowlist	zia_allowlist	`wf_zia_remove_from_allowlist`
ZIA: Remove From Customlist	zia_customlists	`wf_zia_remove_from_customlist`
ZIA: Remove Artifact From Customlist	artifact	`wf_zia_remove_artifact_from_customlist`
ZIA: Get Blocklist	incident	`wf_zia_get_blocklist`
ZIA: Add Artifact To Customlist	artifact	`wf_zia_add_artifact_to_customlist`
ZIA: Add Artifact To Allowlist	artifact	`wf_zia_add_artifact_to_allowlist`
ZIA: Get Allowlist	incident	`wf_zia_get_allowlist`
ZIA: Add URLs To BlockList	incident	`wf_zia_add_urls_to_blocklist`
ZIA: Add URLs To CustomList	incident	`wf_zia_add_urls_to_customlist`
ZIA: Get Customlist	incident	`wf_zia_get_customlist`
ZIA: Add Artifact To Blocklist	artifact	`wf_zia_add_artifact_to_blocklist`
ZIA: Get URL Categories	incident	`wf_zia_get_url_categories`
ZIA: Remove Artifact From Allowlist	artifact	`wf_zia_remove_artifact_from_allowlist`
ZIA: Get Sandbox Report	artifact	`wf_zia_get_sandbox_report`
ZIA: Remove From Blocklist	zia_blocklist	`wf_zia_remove_from_blocklist`
ZIA: Remove Artifact From Blocklist	artifact	`wf_zia_remove_artifact_from_blocklist`
ZIA: URL Lookup	artifact	`wf_zia_url_lookup`
ZIA: Add Custom Category	incident	`wf_zia_add_custom_category`

Troubleshooting & Support

Refer to the documentation listed in the Requirements section for troubleshooting information.

For Support

This is a IBM Community provided App. Please search the Community https://ibm.biz/resilientcommunity for assistance.