
Key Features

The ProofPoint TRAP function package provides the following features:

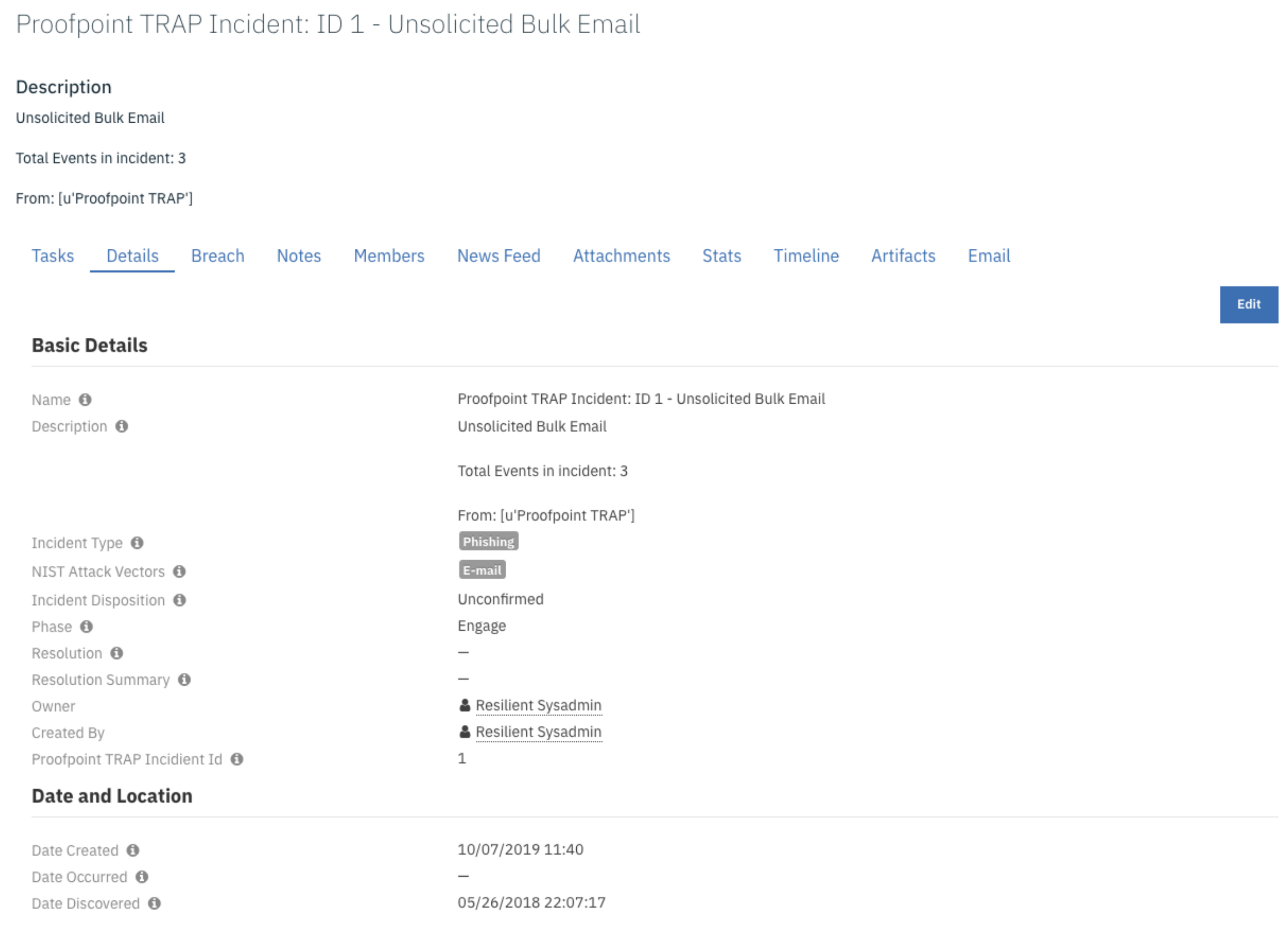
- Poll a Proofpoint TRAP server for incidents and create corresponding incidents in the Resilient platform.
- Get Proofpoint TRAP incident details.
- Get a Proofpoint TRAP list member or members.
- Add a member to a Proofpoint TRAP list for artifacts of type host, IP address, or URL.
- Update a member of a Proofpoint TRAP list.
- Delete a member from a Proofpoint TRAP list.

Poller:

Threaded Poller which runs continuously while the integration is running.

- Polls a Proofpoint TRAP server for incidents and creates corresponding incidents in the Resilient platform.
- Adds Proofpoint TRAP events to incident data table `Proofpoint TRAP Events` in the Resilient platform.
- Adds artifacts to incidents in the Resilient platform corresponding to hosts artifacts in Proofpoint TRAP incident events. The actual artifacts added are determined by the `host_categories` configuration option.
- Adds a note with Proofpoint TRAP events details to an incident in the Resilient platform.

Example incident created by the poller:



Proofpoint TRAP Incident: ID 1 - Unsolicited Bulk Email

Description

Unsolicited Bulk Email

Total Events in incident: 3

From: [u'Proofpoint TRAP']

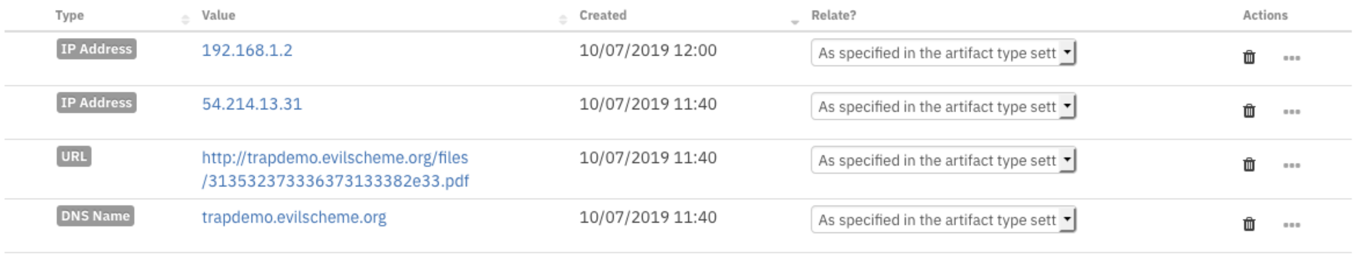
Basic Details

Name	Proofpoint TRAP Incident: ID 1 - Unsolicited Bulk Email
Description	Unsolicited Bulk Email Total Events in incident: 3 From: [u'Proofpoint TRAP']
Incident Type	Phishing
NIST Attack Vectors	E-mail
Incident Disposition	Unconfirmed
Phase	Engage
Resolution	–
Resolution Summary	–
Owner	Resilient Sysadmin
Created By	Resilient Sysadmin
Proofpoint TRAP Incident Id	1

Date and Location

Date Created	10/07/2019 11:40
Date Occurred	–
Date Discovered	05/26/2018 22:07:17

Examples of incident artifacts created by the poller:



Type	Value	Created	Relate?	Actions
IP Address	192.168.1.2	10/07/2019 12:00	As specified in the artifact type sett	
IP Address	54.214.13.31	10/07/2019 11:40	As specified in the artifact type sett	
URL	http://trapdemo.evilscheme.org/files/31353237333637313333382e33.pdf	10/07/2019 11:40	As specified in the artifact type sett	
DNS Name	trapdemo.evilscheme.org	10/07/2019 11:40	As specified in the artifact type sett	

Functions:

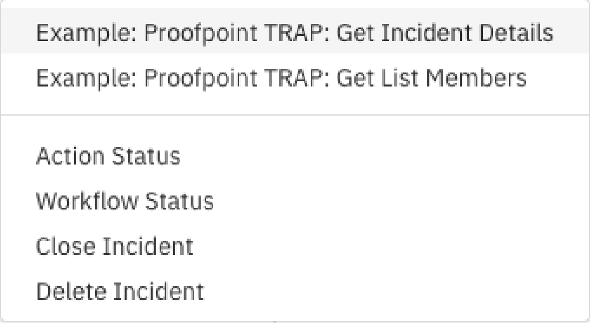
Function - Proofpoint TRAP: Get Incident Details

Fetch Incident Details from Proofpoint TRAP.

- Adds a note to the Resilient incident with ProofPoint TRAP incident details.
- Provides an example workflow which uses this Resilient Function, `Example: Proofpoint TRAP: Get Incident Details`.

The workflow is initiated by the incident rule, `Example: Proofpoint TRAP: Get Incident Details`.

1. Open an incident and select `Example: Proofpoint TRAP: Get Incident Details` from Actions.


Example: Proofpoint TRAP: Get Incident Details
Example: Proofpoint TRAP: Get List Members
Action Status
Workflow Status
Close Incident
Delete Incident

2. Click Action-> Example: Proofpoint TRAP: Get Incident Details. A note is added to the incident with the incident details in JSON format.

This invokes the `Example: Proofpoint TRAP: Get Incident Details` workflow, which calls the `Proofpoint TRAP: Get Incident Details` function.

▶ Inputs:

Config	Type	Required	Example	Description
`trap_incident_id`	`number`	Yes	`1`	Proofpoint TRAP Incident ID

▶ Outputs:

```
results = {
    # TODO: Copy and paste an example of the Function Output within this code block.
    # To see view the output of a Function, run resilient-circuits in DEBUG mode and invoke the Function.
    # The Function results will be printed in the logs: "resilient-circuits run --loglevel=DEBUG"
}
```

▶ Example Pre-Process Script:

```
inputs.trap_incident_id = incident.properties.proofpoint_trap_incident_id
```

▶ Example Post-Process Script:

```
note = "{}".format(unicode(results.data))
incident.addNote(helper.createRichText(note))
```

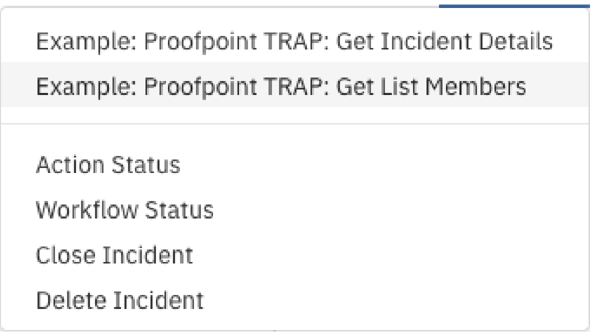
Function - Proofpoint TRAP: Get List Members

Get member or members of a Proofpoint TRAP list.

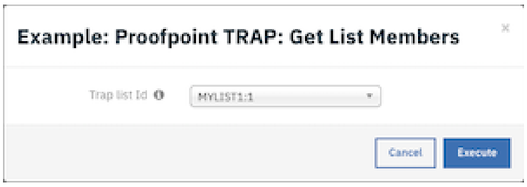
- Retrives all the members of a Proofpoint TRAP list. Lists can be host list, url list, user list, or file list.
- Adds Proofpoint TRAP list members to incident data table `Proofpoint TRAP List Members` in the Resilient platform.
- Example workflows which use this Resilient Function include `Example: Proofpoint TRAP: Get List Members`.

The workflow is initiated by the incident rule, `Example: Proofpoint TRAP: Get List Members`.

1. Open an incident and select `Example: Proofpoint TRAP: Get List Members` from Actions.


Example: Proofpoint TRAP: Get Incident Details
Example: Proofpoint TRAP: Get List Members
Action Status
Workflow Status
Close Incident
Delete Incident

2. Select a list ID from the displayed drop-down list and click Execute.

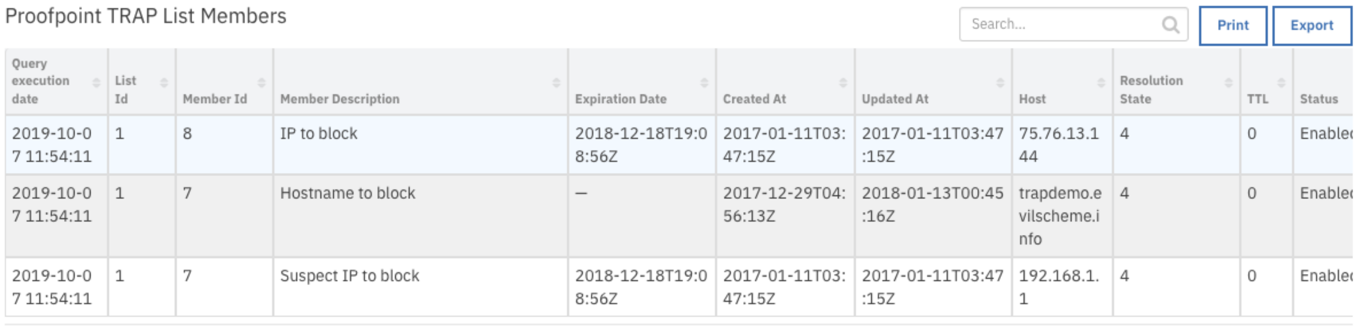


Example: Proofpoint TRAP: Get List Members

Trap list Id: MYLIST1:1

Note: The drop-down for list ID selection uses the format '<list description>:<list id>'.

This invokes the `Example: Proofpoint TRAP: Get List Members` workflow, which calls the `Proofpoint TRAP: Get List Members` function. The data table `Proofpoint TRAP List Members` will be updated in the Resilient platform with the member details for the selected list.



Proofpoint TRAP List Members

Query execution date	List Id	Member Id	Member Description	Expiration Date	Created At	Updated At	Host	Resolution State	TTL	Status
2019-10-07 11:54:11	1	8	IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	75.76.13.144	4	0	Enablec
2019-10-07 11:54:11	1	7	Hostname to block	—	2017-12-29T04:56:13Z	2018-01-13T00:45:16Z	trapdemo.evilscheme.info	4	0	Enablec
2019-10-07 11:54:11	1	7	Suspect IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	192.168.1.1	4	0	Enablec

▶ Inputs:

Config	Type	Required	Example	Description
`trap_list_id`	`number`	Yes	`1`	Proofpoint TRAP List ID.
`trap_member_id`	`number`	No	`1`	Proofpoint TRAP List member ID.
`trap_members_type`	`number`	Yes	`members.json`	The Proofpoint TRAP information format to get in result for list membership. Default is members.json.

▶ Outputs:

```
results = { 'inputs': {u'trap_list_id': 1, u'trap_members_type': u'members.json'},
          'metrics': {'package': 'fn-proofpoint-trap', 'timestamp': '2019-08-29
14:02:01', 'package_version': '1.0.3',
                      'host': 'myhost.com', 'version': '1.0', 'execution_time_ms': 27},
'success': True,
          'content': [{u'user_id': None, u'description': u'IP to block', u'deleted':
False, u'created_at': u'2017-01-11T03:47:15Z',
                       u'enabled': True, u'updated_at': u'2017-01-11T03:47:15Z',
                       u'host': {u'created_at': u'2017-01-11T03:47:15Z', u'updated_at':
u'2017-01-11T03:47:15Z',
                                 u'host': u'75.76.13.144', u'ttl': 0,
u'resolution_state': 4, u'id': 20
                                 },
                       u'response_id': None, u'expiration': u'2018-12-18T19:08:56Z',
u'list_id': 2, u'host_id': 20,
                       u'hash_reputation_id': None, u'id': 8, u'reverse_user_id': None
                      },
                      {u'user_id': None, u'description': u'test', u'deleted': False,
u'created_at': u'2017-01-11T03:43:54Z',
                       u'enabled': True, u'updated_at': u'2017-01-11T03:43:54Z',
                       u'host': {u'created_at': u'2016-12-29T04:56:13Z', u'updated_at':
u'2017-01-13T00:45:16Z',
                                 u'host': u'string', u'ttl': 0, u'resolution_state': 4,
u'id': 6
                                },
                       u'response_id': None, u'expiration': None, u'list_id': 2,
u'host_id': 6, u'hash_reputation_id': None,
                       u'id': 6, u'reverse_user_id': None
                       }],
          'raw': '[{"user_id": null, "description": "IP to block", "deleted": false,
"created_at": "2017-01-11T03:47:15Z", '
                 '"enabled": true, "updated_at": "2017-01-11T03:47:15Z", "host":
{"created_at": "2017-01-11T03:47:15Z", '
                 '"updated_at": "2017-01-11T03:47:15Z", "host": "75.76.13.144", "ttl":
0, "resolution_state": 4, "id": 20}, '
                 '"response_id": null, "expiration": "2018-12-18T19:08:56Z", "list_id":
2, "host_id": 20, '
                 '"hash_reputation_id": null, "id": 8, "reverse_user_id": null},
{"user_id": null, "description": "test", '
                 '"deleted": false, "created_at": "2017-01-11T03:43:54Z", "enabled":
true, "updated_at": "2017-01-11T03:43:54Z", '
                 '"host": {"created_at": "2016-12-29T04:56:13Z", "updated_at": "2017-01-
13T00:45:16Z", "host": "string", '
                 '"ttl": 0, "resolution_state": 4, "id": 6}, "response_id": null,
"expiration": null, "list_id": 2, '
                 '"host_id": 6, "hash_reputation_id": null, "id": 6, "reverse_user_id":
null}]',
          'reason': None,
          'version': '1.0'
}
```

▶ Example Pre-Process Script:

```
import re
if re.match("^.*:\d+$", rule.properties.trap_list_id):
    inputs.trap_list_id = rule.properties.trap_list_id.split(":")[1]
else:
    raise ValueError("Required field: '{0}' with value: '{1}' is in an incorrect
format.".format("rule.properties.trap_list_id", rule.properties.trap_list_id))
inputs.trap_list_id = rule.properties.trap_list_id.split(":")[1]
inputs.trap_members_type = "members.json"
```

▶ Example Post-Process Script:

```
##  ProofPoint Trap - fn_proofpoint_trap_get_list_members ##
# Example result:
"""
Result: { 'inputs': {u'trap_list_id': 1, u'trap_members_type': u'members.json'},
          'metrics': {'package': 'fn-proofpoint-trap', 'timestamp': '2019-08-29
14:02:01', 'package_version': '1.0.3',
                      'host': 'myhost.com', 'version': '1.0', 'execution_time_ms': 27},
'success': True,
          'content': [{u'user_id': None, u'description': u'IP to block', u'deleted':
False, u'created_at': u'2017-01-11T03:47:15Z',
                       u'enabled': True, u'updated_at': u'2017-01-11T03:47:15Z',
                       u'host': {u'created_at': u'2017-01-11T03:47:15Z', u'updated_at':
u'2017-01-11T03:47:15Z',
                                 u'host': u'75.76.13.144', u'ttl': 0,
u'resolution_state': 4, u'id': 20
                                 },
                       u'response_id': None, u'expiration': u'2018-12-18T19:08:56Z',
u'list_id': 2, u'host_id': 20,
                       u'hash_reputation_id': None, u'id': 8, u'reverse_user_id': None
                      },
                      {u'user_id': None, u'description': u'test', u'deleted': False,
u'created_at': u'2017-01-11T03:43:54Z',
                       u'enabled': True, u'updated_at': u'2017-01-11T03:43:54Z',
                       u'host': {u'created_at': u'2016-12-29T04:56:13Z', u'updated_at':
u'2017-01-13T00:45:16Z',
                                 u'host': u'string', u'ttl': 0, u'resolution_state': 4,
u'id': 6
                                },
                       u'response_id': None, u'expiration': None, u'list_id': 2,
u'host_id': 6, u'hash_reputation_id': None,
                       u'id': 6, u'reverse_user_id': None
                       }],
          'raw': '[{"user_id": null, "description": "IP to block", "deleted": false,
"created_at": "2017-01-11T03:47:15Z", '
                 '"enabled": true, "updated_at": "2017-01-11T03:47:15Z", "host":
{"created_at": "2017-01-11T03:47:15Z", '
                 '"updated_at": "2017-01-11T03:47:15Z", "host": "75.76.13.144", "ttl":
0, "resolution_state": 4, "id": 20}, '
                 '"response_id": null, "expiration": "2018-12-18T19:08:56Z", "list_id":
2, "host_id": 20, '
                 '"hash_reputation_id": null, "id": 8, "reverse_user_id": null},
{"user_id": null, "description": "test", '
                 '"deleted": false, "created_at": "2017-01-11T03:43:54Z", "enabled":
true, "updated_at": "2017-01-11T03:43:54Z", '
                 '"host": {"created_at": "2016-12-29T04:56:13Z", "updated_at": "2017-01-
13T00:45:16Z", "host": "string", '
                 '"ttl": 0, "resolution_state": 4, "id": 6}, "response_id": null,
"expiration": null, "list_id": 2, '
                 '"host_id": 6, "hash_reputation_id": null, "id": 6, "reverse_user_id":
null}]',
          'reason': None,
          'version': '1.0'
}
"""
#  Globals
# List of fields in datatable fn_proofpoint_trap_get_list_members script
DATA_TBL_FIELDS = ["query_execution_time", "member_id", "list_id", "member_description",
"expiration", "created_at", "status"]
DATA_TBL_FIELDS_HOST = ["created_at", "updated_at", "host", "resolution_state", "ttl"]
FN_NAME = "fn_proofpoint_trap_get_list_members"
WF_NAME = "Example: Proofpoint TRAP: Get List Members"
MEMBERS = results.content
INPUTS = results.inputs
QUERY_EXECUTION_DATE = results["metrics"]["timestamp"]

# Processing
```

```
def main():
    note_text = ''
    if MEMBERS is not None:
        note_text = "ProofPoint Trap Integration Integration: Workflow <b>{0}</b>: There
were <b>{1}</b> results returned for Resilient function " \
                     "<b>{2}</b>".format(WF_NAME, len(MEMBERS), FN_NAME)
        for i in range(len(MEMBERS)):
            newrow = incident.addRow("trap_list_members")
            newrow.query_execution_date = QUERY_EXECUTION_DATE
            newrow.member_id = MEMBERS[i]["id"]
            newrow.member_description = MEMBERS[i]["description"]
            for f in DATA_TBL_FIELDS:
                if f in ["query_execution_time", "member_id", "member_description",
"status"]:
                    continue
                if MEMBERS[i][f] is not None:
                    newrow[f] = MEMBERS[i][f]
                if MEMBERS[i]["enabled"]:
                    newrow.status = "Enabled"
                host = MEMBERS[i]["host"]
                if host is not None:
                    for d in DATA_TBL_FIELDS_HOST:
                        newrow[d] = host[d]
    else:
        noteText += "ProofPoint Trap Integration: Workflow <b>{0}</b>: There were
<b>no</b> results returned  for " \
                    "list id <b>{1}</b> for Resilient function <b>{3}
</b>".format(WF_NAME, INPUTS["trap_list_id"], FN_NAME)

    incident.addNote(helper.createRichText(note_text))

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

Function - Proofpoint TRAP: Add Members to List

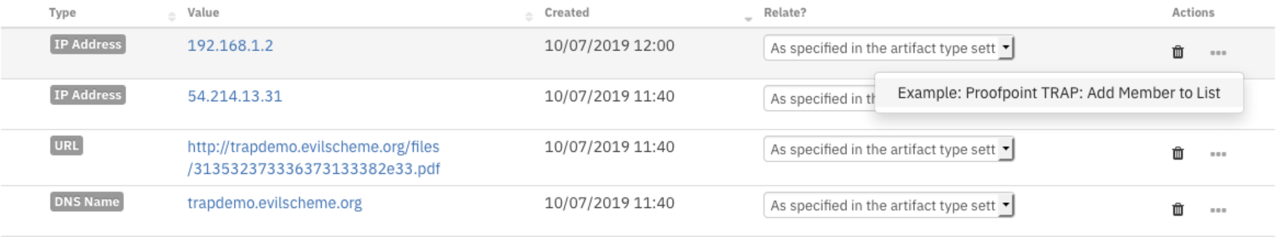
Add a member or members to a Proofpoint TRAP list.

- Adds a new host, IP address, URL, user account or hash member to a Proofpoint TRAP list. List can be host list, URL list, user list, or file list.
- Adds the new Proofpoint TRAP list member details to the incident data table `Proofpoint TRAP List Members` in the Resilient platform.
- An example workflow that uses this Resilient Function includes `Example: Proofpoint TRAP: Add Member to List`. The workflow is applicable for Resilient artifact types: 'URL', 'User Account', 'DNS Name', 'IP Address', System Name', 'Malware SHA-256 Hash', 'Malware SHA-1 Hash' and 'Malware MD5 Hash'.

The workflow is initiated by the incident rule, `Example: Proofpoint TRAP: Add Member to List`.

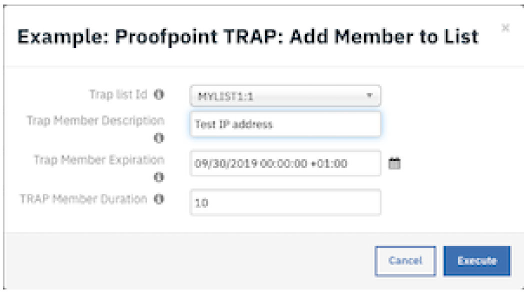
1. Open an incident and select the 'Artifacts' tab.

2. For a Resilient artifact of an applicable type, such as 'IP Address,' click Action-> `Example: Proofpoint TRAP: Add Member to List`.



Type	Value	Created	Relate?	Actions
IP Address	192.168.1.2	10/07/2019 12:00	As specified in the artifact type sett	
IP Address	54.214.13.31	10/07/2019 11:40	As specified in th	
URL	http://trapdemo.evilscheme.org/files /31353237333637313338 2e33.pdf	10/07/2019 11:40	As specified in the artifact type sett	
DNS Name	trapdemo.evilscheme.org	10/07/2019 11:40	As specified in the artifact type sett	

3. Select an input from a list of user defined inputs. For example, select Proofpoint TRAP list ID 1 from the drop-down.



Example: Proofpoint TRAP: Add Member to List

Trap list Id: MYLIST1:1

Trap Member Description: Test IP address

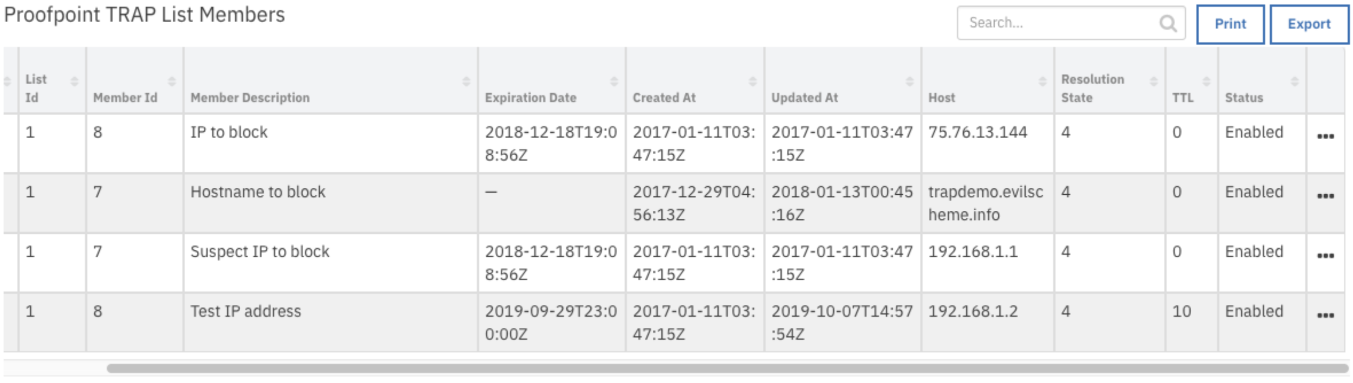
Trap Member Expiration: 09/30/2019 00:00:00 +01:00

TRAP Member Duration: 10

Note: The drop-down for list ID selection uses the format '<list description>:<list id>'.

4. Input values for the remainder of the fields and click Execute. This invokes the `Example: Proofpoint TRAP: Add Member to List` workflow, which calls the `Proofpoint TRAP: Add Members to List` function.

The data table `Proofpoint TRAP List Members`, is updated in the Resilient platform with the member details for the selected list.



Proofpoint TRAP List Members

List Id	Member Id	Member Description	Expiration Date	Created At	Updated At	Host	Resolution State	TTL	Status
1	8	IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	75.76.13.144	4	0	Enabled
1	7	Hostname to block	–	2017-12-29T04:56:13Z	2018-01-13T00:45:16Z	trapdemo.evilscheme.info	4	0	Enabled
1	7	Suspect IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	192.168.1.1	4	0	Enabled
1	8	Test IP address	2019-09-29T23:00:00Z	2017-01-11T03:47:15Z	2019-10-07T14:57:54Z	192.168.1.2	4	10	Enabled

▶ Inputs:

Name	Type	Required	Example	Tooltip
`trap_description`	`text`	No	`-`	Short description of Proofpoint TRAP list member.
`trap_duration`	`number`	No	`-`	Number of minutes after which to expire Proofpoint TRAP list membership.
`trap_expiration`	`datetimepicker`	No	`-`	Timestamp to expire Proofpoint TRAP list member.
`trap_list_id`	`number`	Yes	`-`	Proofpoint TRAP List ID.
`trap_member`	`text`	Yes	`-`	Proofpoint TRAP List member to add. Can be of type host, IP address, or URL.

▶ Outputs:

```
results = {
    # TODO: Copy and paste an example of the Function Output within this code block.
    # To see view the output of a Function, run resilient-circuits in DEBUG mode and
invoke the Function.
    # The Function results will be printed in the logs: "resilient-circuits run --
loglevel=DEBUG"
}
```

▶ Example Pre-Process Script:

```
import re
inputs.trap_member = artifact.value
inputs.trap_description = rule.properties.trap_description
if re.match("^.*:\d+$", rule.properties.trap_list_id):
    inputs.trap_list_id = rule.properties.trap_list_id.split(":")[1]
else:
    raise ValueError("Required field: '{0}' with value: '{1}' is in an incorrect
format.".format("rule.properties.trap_list_id", rule.properties.trap_list_id))
inputs.trap_expiration =  rule.properties.trap_expiration
inputs.trap_duration =  rule.properties.trap_duration
```

▶ Example Post-Process Script:

```
##  ProofPoint Trap - fn_proofpoint_trap_add_members_to_list ##
# Example result:
"""
Result: {'inputs': {u'trap_list_id': 1, u'trap_member': u'75.76.13.144',
u'trap_description': u'A test',
                    u'trap_expiration': 1567526694000, u'trap_duration': 10},
         'metrics': {'package': 'fn-proofpoint-trap', 'timestamp': '2019-09-03
17:05:06', 'package_version': '1.0.3',
                     'host': 'myhost.ibm.com', 'version': '1.0', 'execution_time_ms':
3512},
         'success': True,
         'content': {u'user_id': None, u'description': u'IP to block', u'deleted':
False,
                     u'created_at': u'2017-01-11T03:47:15Z', u'enabled': True,
u'updated_at': u'2017-01-11T03:47:15Z',
                     u'host': {u'created_at': u'2017-01-11T03:47:15Z', u'updated_at':
u'2017-01-11T03:47:15Z',
                               u'host': u'75.76.13.144', u'ttl': 0, u'resolution_state':
4, u'id': 20
                               },
                     u'response_id': None, u'expiration': u'2018-12-18T19:08:56Z',
u'list_id': 1, u'host_id': 20,
                     u'hash_reputation_id': None, u'id': 8, u'reverse_user_id': None},
         'raw': '{"user_id": null, "description": "IP to block", "deleted": false,
"created_at": "2017-01-11T03:47:15Z", '
                '"enabled": true, "updated_at": "2017-01-11T03:47:15Z", "host":
{"created_at": "2017-01-11T03:47:15Z", '
                '"updated_at": "2017-01-11T03:47:15Z", "host": "75.76.13.144", "ttl": 0,
"resolution_state": 4, "id": 20}, '
                '"response_id": null, "expiration": "2018-12-18T19:08:56Z", "list_id":
1, "host_id": 20, '
                '"hash_reputation_id": null, "id": 8, "reverse_user_id": null}',
         'reason': None,
         'version': '1.0'
}
"""
#  Globals
# List of fields in datatable fn_proofpoint_trap_add_members_to_list script
DATA_TBL_FIELDS = ["query_execution_time", "member_id", "list_id", "member_description", "expiration", "created_at", "status"]
DATA_TBL_FIELDS_HOST = ["created_at", "updated_at", "host", "resolution_state", "ttl"]
FN_NAME = "fn_proofpoint_trap_add_members_to_list"
WF_NAME = "Example: Proofpoint TRAP: Get List Members"
MEMBER = results.content
INPUTS = results.inputs
QUERY_EXECUTION_DATE = results["metrics"]["timestamp"]

# Processing


def main():
    note_text = ''
    if MEMBER is not None:
        note_text = "ProofPoint Trap Integration Integration: Workflow <b>{0}</b>: There was a valid result returned for Resilient function " \
                   "<b>{1}</b>".format(WF_NAME, FN_NAME)

        newrow = incident.addRow("trap_list_members")
        newrow.query_execution_date = QUERY_EXECUTION_DATE
        newrow.member_id = MEMBER["id"]
        newrow.member_description = MEMBER["description"]
        for f in DATA_TBL_FIELDS:
            if f in ["query_execution_time", "member_id", "member_description", "status"]:
                continue
            if MEMBER[f] is not None:
                  newrow[f] = MEMBER[f]
        if MEMBER["enabled"]:
            newrow.status = "Enabled"
        host = MEMBER["host"]
        if host is not None:
            for d in DATA_TBL_FIELDS_HOST:
                newrow[d] = host[d]


    else:
        noteText += "ProofPoint Trap Integration: Workflow <b>{0}</b>: There were <b>no</b> results returned  for " \
                    "list id <b>{1}</b> for Resilient function <b>{3}</b>".format(WF_NAME, INPUTS["trap_list_id"], FN_NAME)

    incident.addNote(helper.createRichText(note_text))

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

Function - Proofpoint TRAP: Update List Member

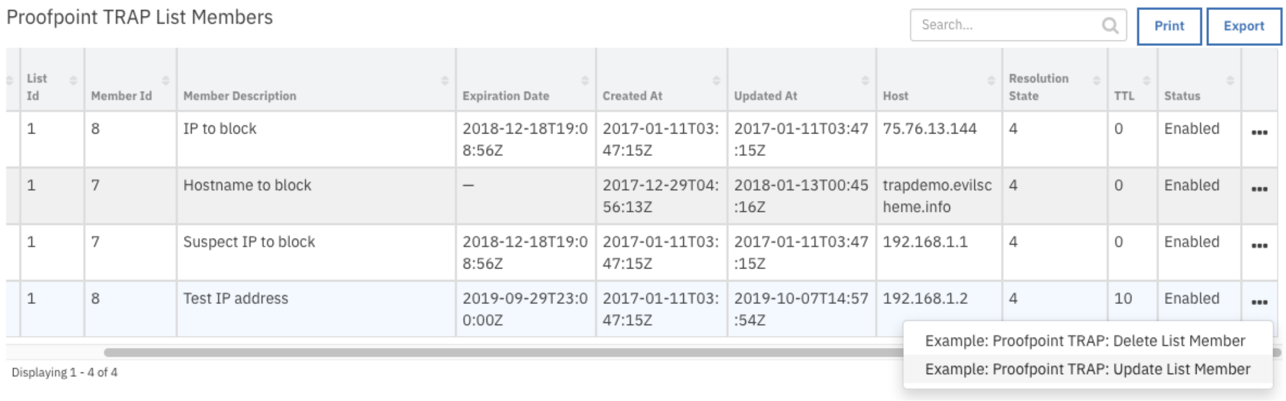
Update a member of a Proofpoint TRAP list.

- Updates an existing member of a Proofpoint TRAP list by specifying the list and the member to update.
- Refreshes Proofpoint TRAP list member details for the incident data table `Proofpoint TRAP List Members` in the Resilient platform.

- An example workflow that uses this Resilient Function includes `Example: Proofpoint TRAP: Update List Member`.
- The workflow is applicable for populated rows of incident data table `Proofpoint TRAP List Members` in the Resilient platform.

The workflow is initiated by the data table rule, `Example: Proofpoint TRAP: Update List Member`.

1. Open an incident and select the row of data table `Proofpoint TRAP List Members`corresponding to the list member to update.
2. From the selected row's actions menu, select `Example: Proofpoint TRAP: Update List Member`.



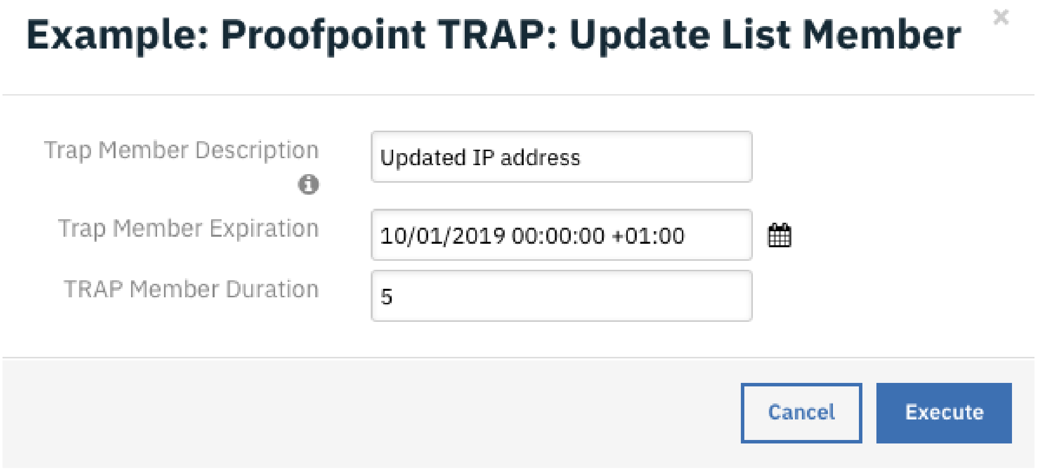
Proofpoint TRAP List Members

List Id	Member Id	Member Description	Expiration Date	Created At	Updated At	Host	Resolution State	TTL	Status	
1	8	IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	75.76.13.144	4	0	Enabled	•••
1	7	Hostname to block	–	2017-12-29T04:56:13Z	2018-01-13T00:45:16Z	trapdemo.evilscheme.info	4	0	Enabled	•••
1	7	Suspect IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	192.168.1.1	4	0	Enabled	•••
1	8	Test IP address	2019-09-29T23:00:00Z	2017-01-11T03:47:15Z	2019-10-07T14:57:54Z	192.168.1.2	4	10	Enabled	•••

Displaying 1 - 4 of 4

Example: Proofpoint TRAP: Delete List Member
Example: Proofpoint TRAP: Update List Member

3. In the list of user defined inputs, enter values for the displayed fields and click Execute.



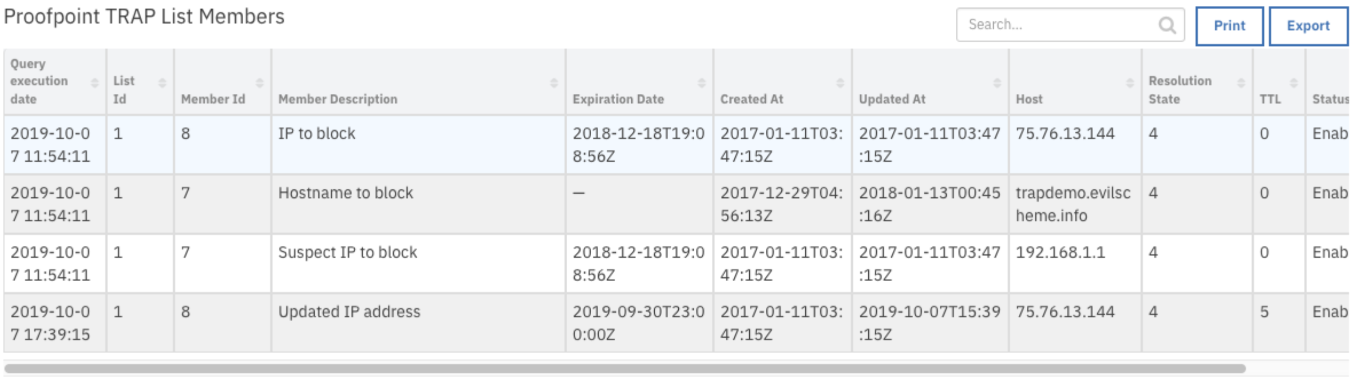
Example: Proofpoint TRAP: Update List Member

Trap Member Description: Updated IP address

Trap Member Expiration: 10/01/2019 00:00:00 +01:00

TRAP Member Duration: 5

This invokes the `Example: Proofpoint TRAP: Update List Member` workflow, which calls the `Proofpoint TRAP: Update List Member` function. The data table `Proofpoint TRAP List Members` is refreshed in the Resilient platform with the updated member details for the selected list.



Proofpoint TRAP List Members

Query execution date	List Id	Member Id	Member Description	Expiration Date	Created At	Updated At	Host	Resolution State	TTL	Status
2019-10-07 11:54:11	1	8	IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	75.76.13.144	4	0	Enab
2019-10-07 11:54:11	1	7	Hostname to block	–	2017-12-29T04:56:13Z	2018-01-13T00:45:16Z	trapdemo.evilscheme.info	4	0	Enab
2019-10-07 11:54:11	1	7	Suspect IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	192.168.1.1	4	0	Enab
2019-10-07 17:39:15	1	8	Updated IP address	2019-09-30T23:00:00Z	2017-01-11T03:47:15Z	2019-10-07T15:39:15Z	75.76.13.144	4	5	Enab

▶ Inputs:

Name	Type	Required	Example	Tooltip
`trap_description`	`text`	No	`-`	Short description of Proofpoint TRAP list member.
`trap_duration`	`number`	No	`-`	Number of minutes after which to expire Proofpoint TRAP list membership.
`trap_expiration`	`datetimepicker`	No	`-`	Timestamp to expire Proofpoint TRAP list member.
`trap_list_id`	`number`	Yes	`-`	Proofpoint TRAP List ID.
`trap_member_id`	`number`	Yes	`-`	Proofpoint TRAP List member ID.

▶ Outputs:

```
results = {
    # TODO: Copy and paste an example of the Function Output within this code block.
    # To see view the output of a Function, run resilient-circuits in DEBUG mode and
invoke the Function.
    # The Function results will be printed in the logs: "resilient-circuits run --
loglevel=DEBUG"
}
```

▶ Example Pre-Process Script:

```
inputs.trap_list_id = row.list_id
inputs.trap_member_id = row.member_id
inputs.trap_description = rule.properties.trap_description
inputs.trap_expiration =  rule.properties.trap_expiration
inputs.trap_duration =  rule.properties.trap_duration
```

▶ Example Post-Process Script:

```
##  ProofPoint Trap - fn_proofpoint_trap_update_list_member ##
# Example result:

"""
Results {'inputs': {u'trap_list_id': 2, u'trap_expiration': 1567527591000,
u'trap_description': u'Test',
                    u'trap_member_id': 8, u'trap_duration': 10},
         'metrics': {'package': 'fn-proofpoint-trap', 'timestamp': '2019-09-03
17:19:57', 'package_version': '1.0.3',
                     'host': 'myhost.ibm.com', 'version': '1.0', 'execution_time_ms': 33
                     }, 'success': True,
         'content': {u'user_id': None, u'description': u'Updated IP to block',
u'deleted': False,
                     u'created_at': u'2017-01-11T03:47:15Z', u'enabled': True,
u'updated_at': u'2017-01-11T03:47:15Z',
                     u'host': {u'created_at': u'2017-01-11T03:47:15Z', u'updated_at':
u'2017-01-11T03:47:15Z',
                               u'host': u'75.76.13.144', u'ttl': 0, u'resolution_state':
4, u'id': 20
                               },
                     u'response_id': None, u'expiration': u'2018-12-18T19:08:56Z',
u'list_id': 2, u'host_id': 20,
                     u'hash_reputation_id': None, u'id': 8, u'reverse_user_id': None
                     },
         'raw': '{"user_id": null, "description": "Updated IP to block", "deleted":
false, "created_at": "2017-01-11T03:47:15Z", '
                '"enabled": true, "updated_at": "2017-01-11T03:47:15Z", "host":
{"created_at": "2017-01-11T03:47:15Z", '
                '"updated_at": "2017-01-11T03:47:15Z", "host": "75.76.13.144", "ttl": 0,
"resolution_state": 4, "id": 20}, '
                '"response_id": null, "expiration": "2018-12-18T19:08:56Z", "list_id":
2, "host_id": 20, "hash_reputation_id": null, '
                '"id": 8, "reverse_user_id": null}',
         'reason': None,
         'version': '1.0'
}
"""

#  Globals
# List of fields in datatable "Proofpoint TRAP List Members" for
fn_proofpoint_trap_get_list_members script
DATA_TBL_FIELDS = ["query_execution_time", "member_id", "list_id", "member_description",
"expiration", "created_at", "status"]
DATA_TBL_FIELDS_HOST = ["created_at", "updated_at", "host", "resolution_state", "ttl"]
FN_NAME = "fn_proofpoint_trap_update_list_member"
WF_NAME = "Example: Proofpoint TRAP: Update List Member"
MEMBER = results.content
QUERY_EXECUTION_DATE = results["metrics"]["timestamp"]

# Processing


def main():
    note_text = ''
    if MEMBER is not None:
        note_text = "ProofPoint Trap Integration Integration: Workflow <b>{0}</b>: There
was a valid result returned for Resilient function " \
                   "<b>{1}</b>".format(WF_NAME, FN_NAME)

        row.query_execution_date = QUERY_EXECUTION_DATE
        row.member_id = MEMBER["id"]
        row.member_description = MEMBER["description"]
        for f in DATA_TBL_FIELDS:
            if f in ["query_execution_time", "member_id", "member_description",
"status"]:
                continue
            if MEMBER[f] is not None:
                  row[f] = MEMBER[f]
        if MEMBER["enabled"]:
            row.status = "Enabled"
        host = MEMBER["host"]
        if host is not None:
            for d in DATA_TBL_FIELDS_HOST:
                row[d] = host[d]
    else:
        noteText += "ProofPoint Trap Integration: Workflow <b>{0}</b>: There were
<b>no</b> results returned  for " \
                    "list id <b>{1}</b> for Resilient function <b>{3}
</b>".format(WF_NAME, INPUTS["trap_list_id"], FN_NAME)

    incident.addNote(helper.createRichText(note_text))

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

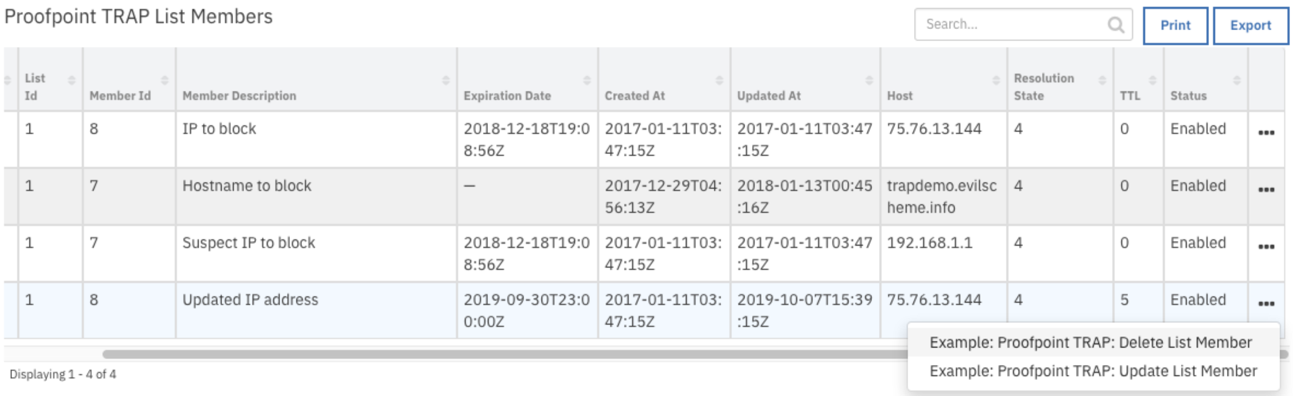
Function - Proofpoint TRAP: Delete List Member

Delete the member of a Proofpoint TRAP list.

- Deletes a member of a Proofpoint TRAP list by specifying the list and the member to update.
- Updates the deleted Proofpoint TRAP list member details for the selected incident data table `Proofpoint TRAP List Members` row in the Resilient platform.
- An example workflow that uses this Resilient Function includes `Example: Proofpoint TRAP: Delete List Member`.
- The workflow is applicable for incident data table `Proofpoint TRAP List Members` populated rows in the Resilient platform.

The workflow is initiated by the data table rule, `Example: Proofpoint TRAP: Delete List Member`.

1. Open an incident and select the row of data table `Proofpoint TRAP List Members`corresponding to the list member to delete.
2. From the selected row's actions menu, select `Example: Proofpoint TRAP: Delete List Member`.

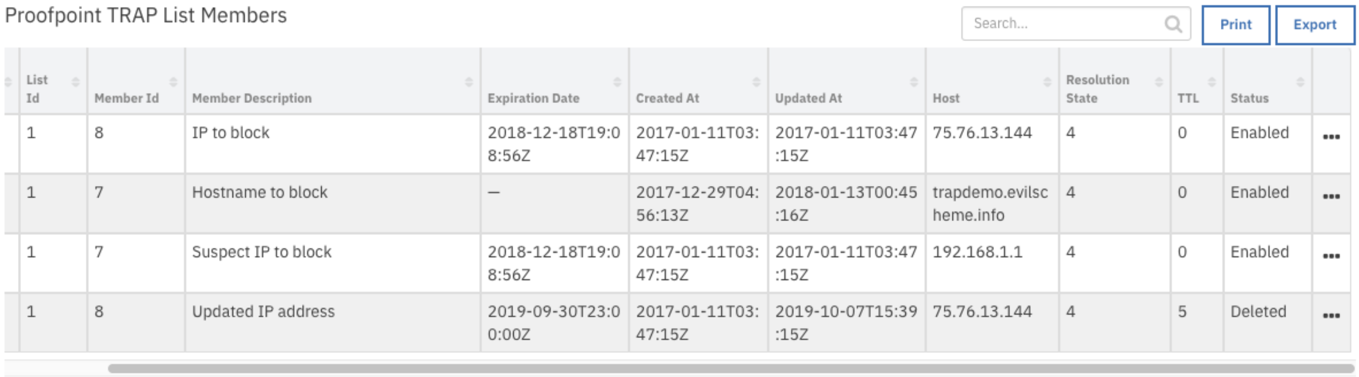

Proofpoint TRAP List Members
Search...
Print
Export
Example: Proofpoint TRAP: Delete List Member
Example: Proofpoint TRAP: Update List Member

List Id	Member Id	Member Description	Expiration Date	Created At	Updated At	Host	Resolution State	TTL	Status	
1	8	IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	75.76.13.144	4	0	Enabled	•••
1	7	Hostname to block	—	2017-12-29T04:56:13Z	2018-01-13T00:45:16Z	trapdemo.evilscheme.info	4	0	Enabled	•••
1	7	Suspect IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	192.168.1.1	4	0	Enabled	•••
1	8	Updated IP address	2019-09-30T23:00:00Z	2017-01-11T03:47:15Z	2019-10-07T15:39:15Z	75.76.13.144	4	5	Enabled	•••

Displaying 1 - 4 of 4

3. Click on the action to execute. This invokes the `Example: Proofpoint TRAP: Delete List Member` workflow, which calls the `Proofpoint TRAP: Delete List Member` function.

The data table `Proofpoint TRAP List Members` will be refreshed in the Resilient platform for the deleted member with `Status` set to `Deleted`.



List Id	Member Id	Member Description	Expiration Date	Created At	Updated At	Host	Resolution State	TTL	Status	
1	8	IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	75.76.13.144	4	0	Enabled	•••
1	7	Hostname to block	—	2017-12-29T04:56:13Z	2018-01-13T00:45:16Z	trapdemo.evilscheme.info	4	0	Enabled	•••
1	7	Suspect IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	192.168.1.1	4	0	Enabled	•••
1	8	Updated IP address	2019-09-30T23:00:00Z	2017-01-11T03:47:15Z	2019-10-07T15:39:15Z	75.76.13.144	4	5	Deleted	•••

▶ Inputs:

Name	Type	Required	Example	Tooltip
`trap_list_id`	`number`	Yes	`-`	Proofpoint TRAP List ID.
`trap_member_id`	`number`	Yes	`-`	Proofpoint TRAP List member ID.

▶ Outputs:

```
results = {
    # TODO: Copy and paste an example of the Function Output within this code block.
    # To see view the output of a Function, run resilient-circuits in DEBUG mode and
invoke the Function.
    # The Function results will be printed in the logs: "resilient-circuits run --
loglevel=DEBUG"
}
```

▶ Example Pre-Process Script:

```
inputs.trap_list_id = row.list_id
inputs.trap_member_id = row.member_id
```

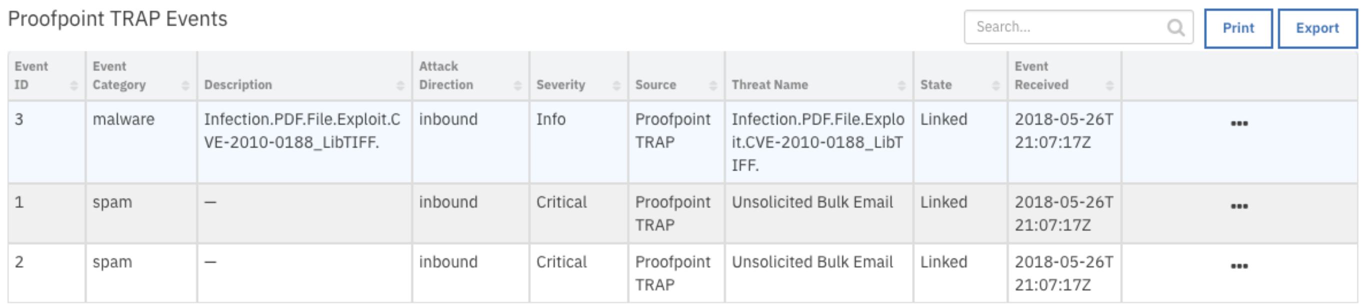
▶ Example Post-Process Script:

```
##  ProofPoint Trap - fn_proofpoint_trap_delete_list_member##
# Example result:

"""
Results {'inputs': {u'trap_list_id': 1, u'trap_member_id': 8},
         'metrics': {'package': 'fn-proofpoint-trap', 'timestamp': '2019-09-04
17:06:02', 'package_version': '1.0.3',
                     'host': 'myhost.ibm.com', 'version': '1.0', 'execution_time_ms': 39
                     },
         'success': True,
         'content': 'OK',
         'raw': '"OK"',
         'reason': None,
         'version': '1.0'
}
"""
#  Globals
# List of fields in datatable "Proofpoint TRAP List Members" for
fn_proofpoint_trap_delete_list_member script
DATA_TBL_FIELDS = ["query_execution_time", "status"]
FN_NAME = "fn_proofpoint_trap_delete_list_member"
WF_NAME = "Example: Proofpoint TRAP: Delete List Member"
STATUS = results.content
INPUTS = results.inputs
QUERY_EXECUTION_DATE = results["metrics"]["timestamp"]

# Processing


def main():
    note_text = ''
    if STATUS is not None:
        note_text = "ProofPoint Trap Integration Integration: Workflow <b>{0}</b>: There
was a valid result returned for Resilient function " \
                    "<b>{1}</b>".format(WF_NAME, FN_NAME)

        row.query_execution_date = QUERY_EXECUTION_DATE
        if STATUS.lower() == "ok":
            row.status = "Deleted"
    else:
        noteText += "ProofPoint Trap Integration: Workflow <b>{0}</b>: There were 
<b>no</b> results returned  for " \
                    "list id <b>{1}</b> for Resilient function <b>{3}
</b>".format(WF_NAME, INPUTS["trap_list_id"], FN_NAME)

    incident.addNote(helper.createRichText(note_text))

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

Data tables:

Data Table - Proofpoint TRAP Events

This data table is populated by the `poller` for each Resilient incident and has an entry for each event detected in the corresponding Proofpoint incident.



Proofpoint TRAP Events

Event ID	Event Category	Description	Attack Direction	Severity	Source	Threat Name	State	Event Received	
3	malware	Infection.PDF.File.Exploit.CVE-2010-0188_LibTIFF.	inbound	Info	Proofpoint TRAP	Infection.PDF.File.Exploit.CVE-2010-0188_LibTIFF.	Linked	2018-05-26T21:07:17Z	•••
1	spam	—	inbound	Critical	Proofpoint TRAP	Unsolicited Bulk Email	Linked	2018-05-26T21:07:17Z	•••
2	spam	—	inbound	Critical	Proofpoint TRAP	Unsolicited Bulk Email	Linked	2018-05-26T21:07:17Z	•••

API Name:

proofpoint_trap_events

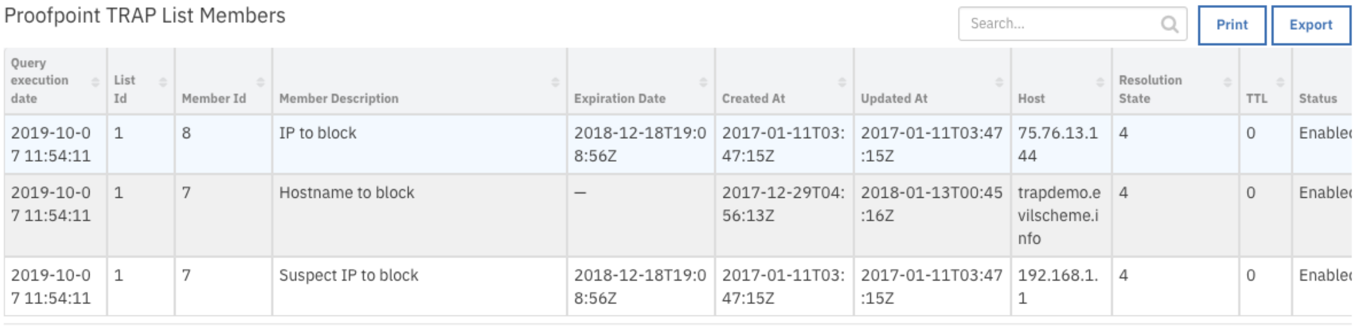
Columns:

Column Name	API Access Name	Type	Tooltip
Attack Direction	`event_attackdirection`	`text`	-
Event Category	`event_category`	`text`	-
Description	`event_description`	`text`	-
Event Id	`event_id`	`number`	-
Event Received	`event_received`	`text`	-
Severity	`event_severity`	`text`	-
Source	`event_source`	`text`	-
State	`event_state`	`text`	-
Threat Name	`event_threatname`	`text`	-

Data Table - Proofpoint TRAP List Members

This data table is typically initiallly populated by a workflow using the Function `Proofpoint TRAP: Get List Members`. An entry is created for all members selected in the workflow.

The data table is also updated by functions `Proofpoint TRAP: Add Members to List`, `Proofpoint TRAP: Update List Member` and `Proofpoint TRAP: Delete List Member`.



Proofpoint TRAP List Members

Query execution date	List Id	Member Id	Member Description	Expiration Date	Created At	Updated At	Host	Resolution State	TTL	Status
2019-10-07 11:54:11	1	8	IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	75.76.13.144	4	0	Enabled
2019-10-07 11:54:11	1	7	Hostname to block	—	2017-12-29T04:56:13Z	2018-01-13T00:45:16Z	trapdemo.evilscheme.info	4	0	Enabled
2019-10-07 11:54:11	1	7	Suspect IP to block	2018-12-18T19:08:56Z	2017-01-11T03:47:15Z	2017-01-11T03:47:15Z	192.168.1.1	4	0	Enabled

API Name:

trap_list_members

Columns:

Column Name	API Access Name	Type	Tooltip
Created At	`created_at`	`text`	-
Expiration Date	`expiration`	`text`	-
Host	`host`	`text`	-
List Id	`list_id`	`number`	-
Member Description	`member_description`	`text`	-
Member Id	`member_id`	`number`	-
Query Execution Date	`query_execution_date`	`text`	-
Resolution State	`resolution_state`	`text`	-
Status	`status`	`text`	-
TTL	`ttl`	`number`	-
Updated At	`updated_at`	`text`	-

Custom Fields

Label	API Access Name	Type	Prefix	Placeholder	Tooltip
proofpoint_trap_incident_id	`proofpoint_trap_incident_id`	`number`	`properties`	-	Proofpoint TRAP incident ID.

Rules

Rule Name	Object	Workflow Triggered

Rule Name	Object	Workflow Triggered
Example: Proofpoint TRAP: Get Incident Details	incident	`wf_proofpoint_trap_get_incident_details`
Example: Proofpoint TRAP: Get List Members	incident	`wf_proofpoint_trap_get_list_members`
Example: Proofpoint TRAP: Add Member to List	artifact	`wf_proofpoint_trap_add_member_to_list`
Example: Proofpoint TRAP: Update List Member	trap_list_members	`wf_proofpoint_trap_update_list_member`
Example: Proofpoint TRAP: Delete List Member	trap_list_members	`wf_proofpoint_trap_delete_list_member`