DEPLOYMENT GUIDE

Integration with Tufin

Table of Contents

Introduction

Infoblox and Tufin together help empower actionable insight into the entire infrastructure, discovering security risks quicker and accurately, investigating and ranking security policies while improving on the organizations security and compliance.

Infoblox provides Tufin with resources such as networks and potential threats, and in exchange, Tufin gets improved management on networks and security risks, including the ability to create policies and keep companies within compliance with the data received through Infoblox. The integration with Infoblox and Tufin allows faster policy management and more insight into the entire network.

Tufin can offer Infoblox networks to improve management on the Infoblox DDI and keep Infoblox within the compliance with policies set by the organization through Tufin. Tufin can add networks to any network view inside Infoblox, allowing further control of the network structure and policies from a single hub on Tufin.

Prerequisites

The following are prerequisites for the integration using Outbound API notifications:

- Infoblox:

  - NIOS 8.3 or higher.

  - Security Ecosystem License.

  - Outbound API integration templates.

  - Prerequisites for the templates (e.g. configured and set extensible attributes).

  - Pre-configured services: RPZ, ADP and Threat Analytics.

  - NIOS API user with the following permissions (access via API only):

    - All Network Views – RW.

    - All IPv4 Networks – RW.

    - All IPv6 Networks – RW.

- Tufin:

  - Tufin SecureChange and SecureTrack

  - Add zones to add networks.

  - Add User with correct permissions.

○ Tufin version 19.1 or higher

Best practices

Outbound API templates can be found on the Infoblox community site on the partners integration page. After registering an account, you can subscribe to the relevant groups and forums. If additional templates come out, they will be found on the Infoblox community site.

For production systems, it is highly recommended to set the log level for an endpoint to "**Info**" or higher ("**Warning**", "**Error**").

Please refer to the Infoblox NIOS Administrator's Guide about other best practices, limitations and any detailed information on how to develop notification templates. The NIOS Administrator's Guide can be found through the Help panel in your Infoblox GUI, or on the Infoblox Support portal.

Known Limitations

The current templates support DNS Firewall (RPZ), Threat Insight (DNS Tunneling), Advanced DNS Protection, Network IPv4, Network IPv6 events only. The asset management template does not support delete or modify events on IPv6 Network events to delete or modify IPv6 networks from Tufin due to limitations with Tufin API. If additional templates become available, they will be found on the Infoblox community site.

Configuration

Workflow

Tufin:

1. Configure Permissions.

2. Create a Network Zone.

Infoblox:

1. Install the Security Ecosystem license if it was not installed.

2. Check that the necessary services and features are properly configured and enabled, including RPZ, ADP and Threat Analytics.

3. Create the required Extensible Attributes.

4. Download (or create your own) notification templates (Tufin_Assets.txt, Tufin_Security.txt, Tufin_Sync.txt, Tufin_Management.txt) from the Infoblox community website.

5. Add the templates.

---

6. Add a REST API Endpoint.

7. Add Notifications.

8. Emulate an event, check Rest API Endpoint debug log and/or verify changes on the grid.

Before you get Started

Download Templates from the Infoblox Community Web-Site

Outbound API templates are an essential part of the configuration. Templates fully control the integration and steps required to execute the outbound notifications. Detailed information on how to develop templates can be found in the NIOS Administrator's guide.

Infoblox does not distribute any templates (out-of-the-box) with the NIOS releases. Templates are available on the Infoblox community web-site. Templates for the Tufin integration will be located in the "Partners Integrations". You can find other templates posted in the "API & Integration" forum.

Templates may require additional extensible attributes, parameters or WAPI credentials to be created or defined. The required configuration should be provided with a template. Don't forget to apply any changes required by the template before testing a notification.

Extensible Attributes

For this integration, the following Extensible Attributes need to be created on the grid.

Table 1. Extensible Attributes

Extensible Attributes	Description	Type
Tufin_Last_Incident	Contains the last date and time when an asset had an incident sent from Infoblox	String
Tufin_Send_Incident	Defines if an asset should send an incident if RPZ, ADP or DNS Tunneling events occur	List (true, false)
Tufin_Sync	Defines if a network should be synced with Tufin.	List (true, false)
Tufin_SyncTime	Contains date/time when the network was synchronized.	String
Tufin_Zone	Defines a List of possible Tufin zones to push networks from Infoblox.	List (Tufin Zone)

Tufin_Sync_Zones	Defines a List of possible Tufin zones to be synced to an Infoblox Network View.	List (Tufin Zones)

Editing Instance Variables

Tufin templates use instance variables to adjust the templates' behavior. Instance variables can be entered through the grid GUI at "**Grid**" → "**Ecosystem**" → "**Notification**" and then selecting the notification you created at "**Edit**" → "**Templates**".

Table 2. Instance Variables

Instance Variable	Description
Network_View_Sync	Defines a Network View on Infoblox that Tufin Zones should be synced to. (used for schedule events only)

Editing Session Variables

The Tufin templates don't use session variables to login to the Tufin instance however, session variables can be entered through the grid GUI at "**Grid**" → "**Ecosystem**" → "**Outbound Endpoint**" and then selecting the endpoint you created at "**Edit**" → "**Session Management**".

Supported Notification

A notification can be considered as a "**link**" between a template, an endpoint and an event. In the notification properties, you define which event triggers the notification, which template is executed and with which API endpoint NIOS will establish the connection to. The Tufin templates support a subset of available notifications (refer to the limitations chapter in this guide for more details). In order to simplify the deployment, only create required notifications and use the relevant filters. It is highly recommended to configure deduplication for RPZ events and exclude a feed that is automatically populated by Threat Analytics.

Table 3. Supported Notifications

Notification	Description
DNS RPZ	DNS queries that are malicious or unwanted
DNS Tunneling	Data exfiltration that occurs on the network
ADP	DNS queries that are malicious or unwanted.
Object Change Network IPv4	Added/Deleted/Modify IPv4 network objects.

| Object Change Network IPv6	Added network IPv6 objects.
Schedule Events	Scheduled time to trigger templates.

Infoblox Permissions

The Infoblox and Tufin integration requires a few permissions for the integration to work. Navigate to "**Administration**" → "**Administrators**" and add "**Roles**", "**Permissions**", "**Groups**" and "**Admins**" to include permissions that are required for the integrations. When creating a new group, under the "**Groups**" tab, select the "**API**" interface under the "**Allowed Interfaces**" category.

Tufin Configuration

Create a Network Zone

In order to create a Network zone in Tufin:

1. In SecureTrack, navigate to "**Network**" → "**Zones**" and click the "**Add Zone**" button.
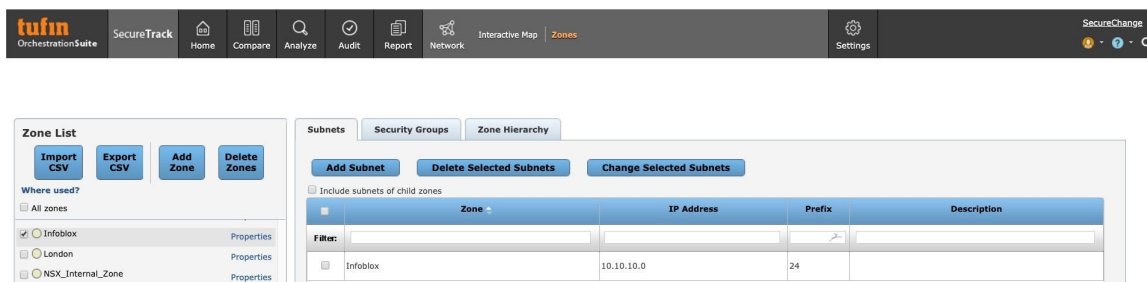
2. On the "**New Zone**" tab enter any name and then click "**Save**".
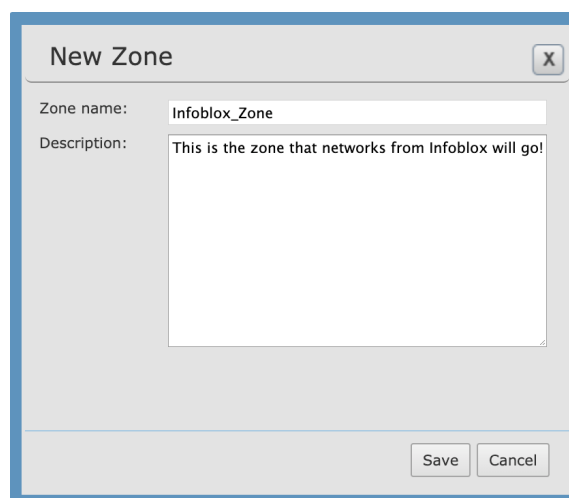
Configure Permissions

In order to configure permissions on the Tufin Appliance you will need to reference the Tufin admin guide.

Infoblox NIOS Configuration

Check if the Security Ecosystem License is Installed

Security Ecosystem License is a "**Grid Wide**" License. Grid wide licenses activate services on all appliances in the same Grid.
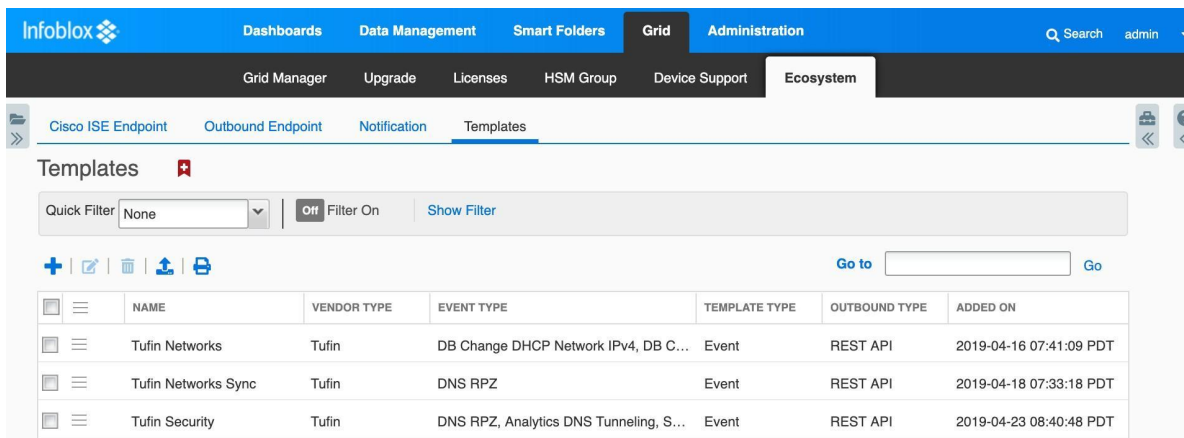
1. In order to check if the license was installed navigate to "**Grid**" → "**Licenses**" → "**Grid Wide**".
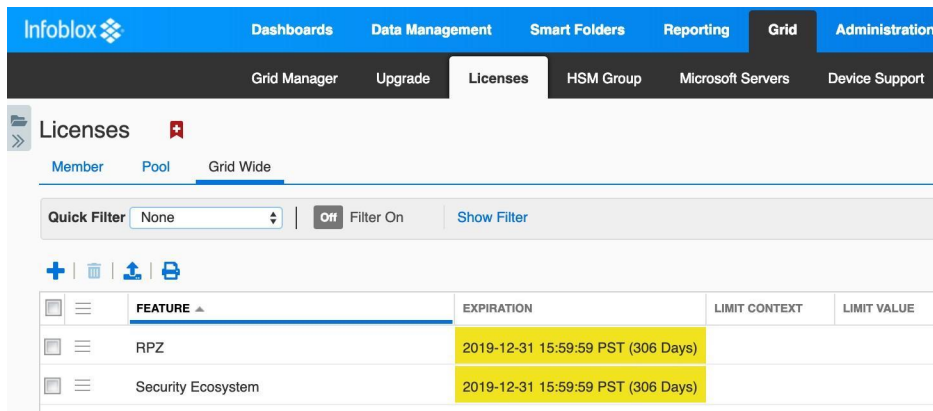

Add/Upload Templates
In order to upload/add templates:

1. Navigate to "**Grid**" → "**Ecosystem**" → "**Templates**" and click "+" or "**+ Add Template**".

2. Click the "**Select**" button on the "**Add template**" window.

3. Click the "**Select**" button on the "**Upload**" window. The standard file selection dialog will open.

4. Select the file and Click the "**Upload**" button on the "**Upload**" window.

5. Click the "**Add**" button and the template will be added/uploaded.
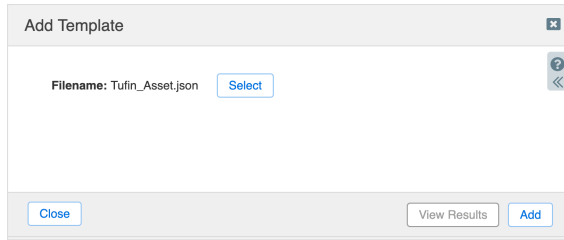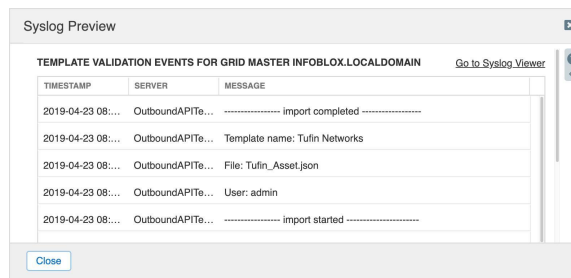
6. If a template was previously uploaded, click "**Yes**" to overwrite the template.
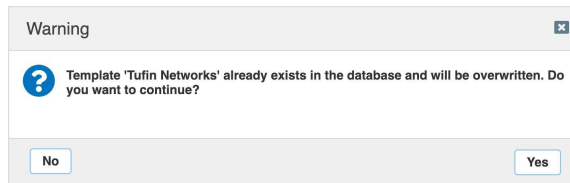
7. You can review the uploaded results in the syslog or by clicking the "**View Results**" button.
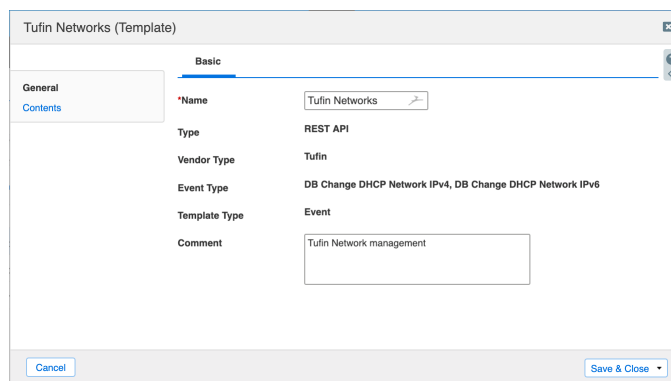

Note: There is no difference between uploading session management and action templates.
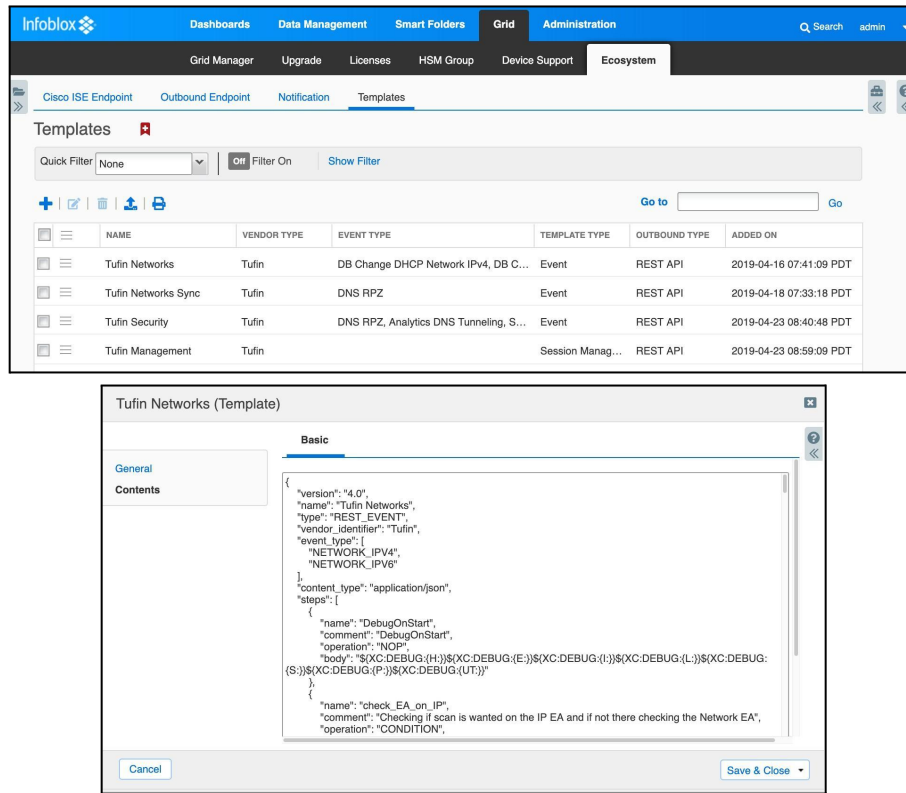
Modifying Templates

NIOS provides the facility to modify the templates via the web-interface.

1. Navigate to "**Grid**" → "**Ecosystem**" → "**Templates**", and then click the triple bar icon next to the template you want to modify.

2. Click the "**Edit**" button to open up the "**Template**" window.

3. Click on the **Contents** tab to view/edit the template.

Note: You cannot delete a template if it is used by an endpoint or by a notification.
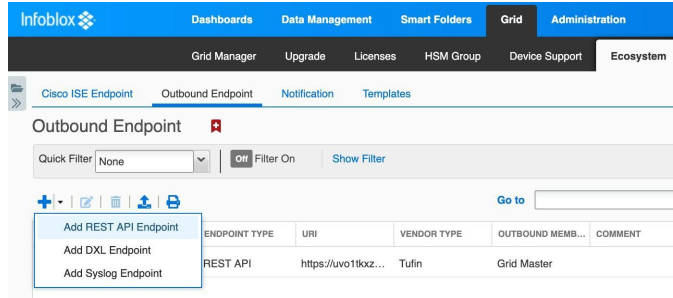
The template editor is a simple interface for making changes to templates. It is recommended to only use the template editor to make minor changes. You can also edit, cut and paste template snippets from a text editor of your choice.Add a Rest API Endpoint.

Add a Rest API Endpoint

A "REST API Endpoint" is basically a remote system which should receive changes based on a notification and a configured template. A Grid, for example, can not only send notifications, it can also receive the notifications from itself (e.g. for testing purposes).

In order to add REST API Endpoints:

1. Navigate to "**Grid**" → "**Ecosystem**" → "**Outbound Endpoints**" and click "**+**" or "**+ Add REST API Endpoint**" buttons. The "**Add REST API Endpoint Wizard**" window will open.

2. Input the **URI** and **Name** for the appliance you are integrating with. *Note: The URI should be the IP/FQDN of the appliance you are integrating with, with the correct URI scheme.*

3. Specify "**WAPI Integration Username**" and "**WAPI Integration Password**" (NIOS credentials).

4. (Optional) For debug purposes only: Under "**Session Management**", set "**Log Level**" to "**Debug**".

5. Select the **Tufin Management** template.


Note: When possible, it is recommended to send notifications from a Grid Master Candidate instead of from the Grid Master.

Add a Notification

A notification can be considered a "link" between a template, an endpoint, and an event. In the notification properties, you define which event triggers the notification, which template is executed and with which API endpoint the NIOS will establish its connection. To simplify the deployment, only create the required notifications and use relevant filters. It is highly recommended to configure deduplication for RPZ events and exclude RPZ feeds that are automatically populated by Threat Analytics.

1. Navigate to "**Grid**" → "**Ecosystem**" → "**Notification**".

---
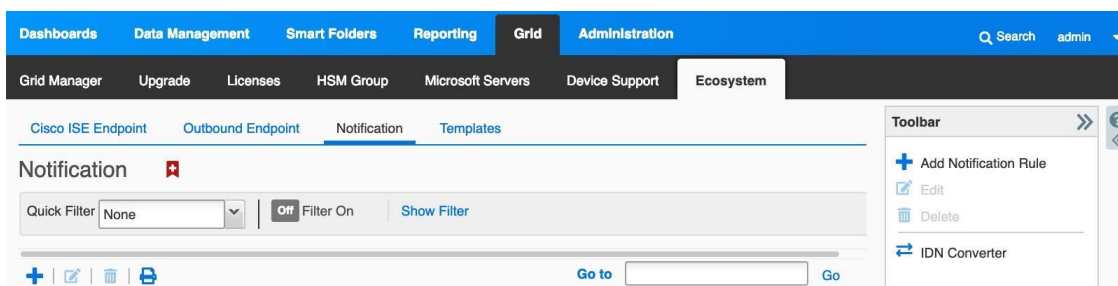
2. Click "**+**" or "**+ Add Notification Rule**" then the "Add Notification Wizard" window will open.

3. Repeat the previous step to add additional Notifications.

Check the Configuration

If required, checking the configuration may be needed. For example, if the templates are not performing as expected. NIOS has a debug log that can assist with this. To use the debug log, perform the following steps

1. (Optional) On the Infoblox grid, navigate to "**Grid**" → "**Ecosystem**" → "**Outbound Endpoint**", select the Tufin endpoint, click on the triple bar icon and select **View Debug** log.
2. (Optional) To clear the debug log, select the **Tufin** endpoint, click on the triple bar icon and select "**Clear Debug Log**". *Note, it is suggested to clear the debug log before running any tests.*
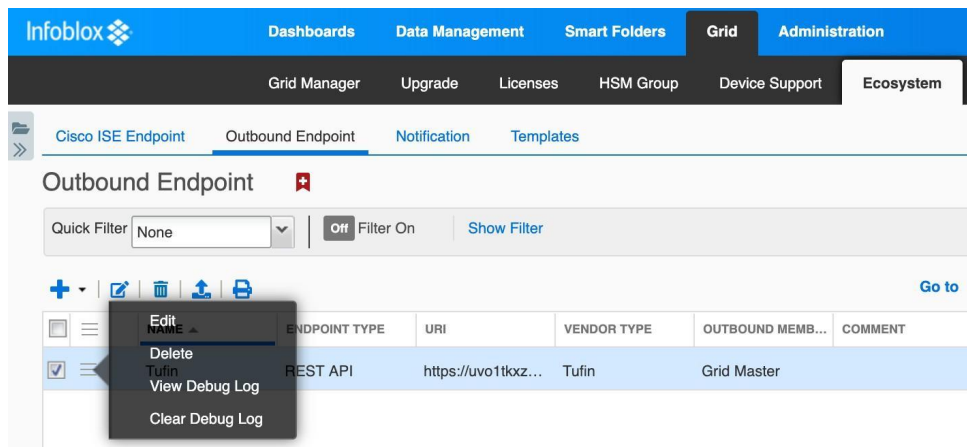
Network Object Management Test

The templates support IPv4/IPv6 Network events. This use case demonstrates how to manage networks on the Tufin.

1. To create an IPv4 Network, navigate to "**Data Management**" → "**IPAM**" → "**Add**" → "**Network**" → "**IPv4**".
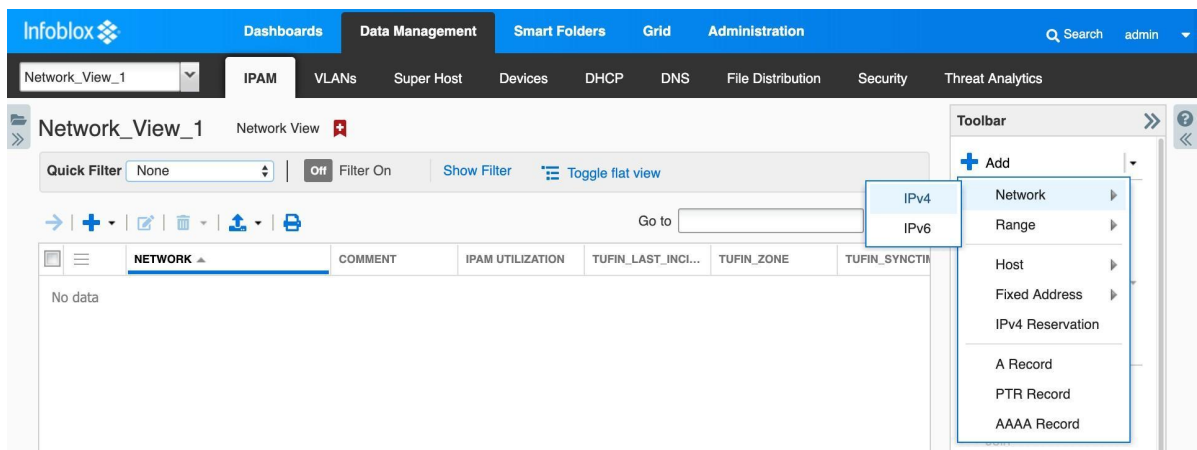
2. Click "**Next**", then insert the network of your choice in this case "**10.10.10.0**" into the "**Networks**" field.

3. Click on "**Next**" till you reach the Extensible Attributes window. If the Extensible Attributes have not already been inherited from the network view, set them.

4. Click "**Save & Close**".

5. Refresh. The "**Tufin_SyncTime**" EA is now updated.
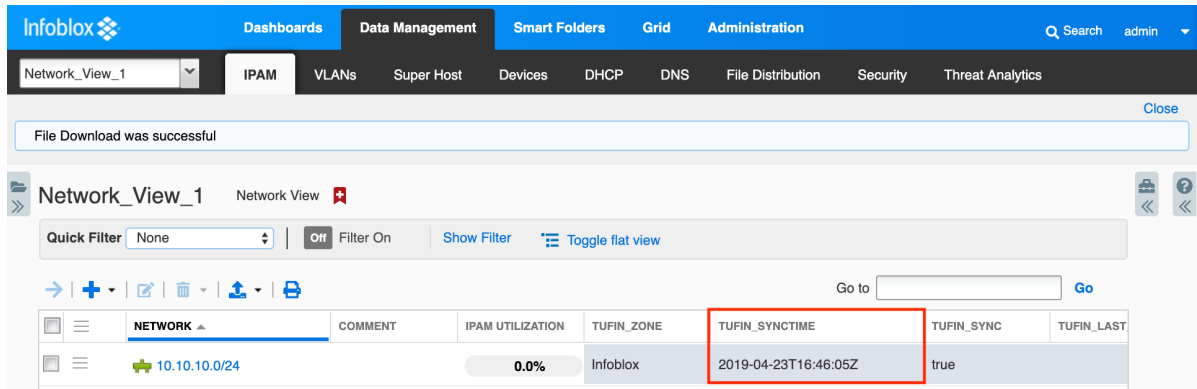
6. In Tufin SecureTrack, navigate to "**Network**" → "**Zones**" then select the zone you sent the asset to. The "**10.10.10.0**" network has been added to the "**Targets**" list. Refresh the page if necessary.
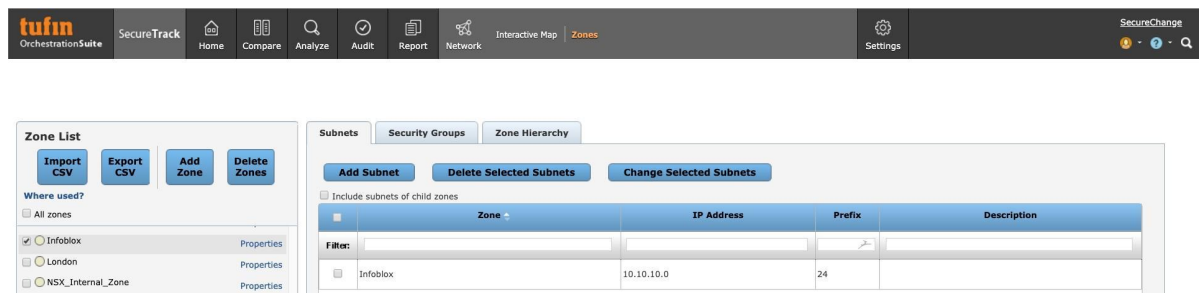
Additional Information

Modifying Security Template for Tufin Workflows

Currently the Tufin Security template is only set up for a very basic workflow and modification of the template is required in order to use it in customized workflows.

In order to modify the security template to fit your customized workflows you will need to grab the "**body_list**" contents on the Step called "**Post_Ticket**" inside the "**Tufin_Security**" template and change the JSON body list to match the specific workflow you would like.

You can check Tufin documentation for how the "**securechange/tickets**" API call works. Currently body used in the Security Template:

```json
{
    "ticket": {
        "subject": "API
        WebApp", "requester":
        "api", "priority":
        "Normal",
        "domain_name":
        "Default", "workflow": {
            "name": "Close Policy"
        },
        "steps": {
            "step": [
                {
                    "name": "Create
                    request", "tasks": {
                        "task": {
                            "fields": {
                                "field": [
                                    {
                                        "@xsi.type": "multi_access_request",
                                        "name": "Flows",
                                        "access_request":
                                        { "targets": {
                                            "target": {
                                                "@type": "ANY"
                                            }
                                        },
                                        "sources": {
                                            "source": [
                                                {
                                                    "@type": "IP",
                                                    "ip_address":${E::source_ip},
                                                    "netmask": "32"
                                                }
                                            ]
                                        },
                                        "destinations":
                                        {
                                            "destination
                                            ": [
                                                {
                                                    "@type": "ANY"
                                                }
                                            ]
                                        },
                                        "services": {
                                            "service": [
                                                {
                                                    "@type": "ANY"
                                                }
                                            ]
                                        },
                                        "action": "Drop",
                                        "comment": "Infoblox ${L:A:Event_Type}
                                        Event", "labels": ""
                                    }
                                },
                                {
                                    "@xsi.type": "text_area",
                                    "name": "Reason",
                                    "text": "Infoblox ${L:A:Event_Type} event from ${L:A:Source} to ${L:A:Target} at ${E:A:timestamp}"
                                }
                            ]
                        }
                    }
                }
            ]
        }
    }
}
```