Blue Planet Security Guide

Blue Planet Release 23.08

November 2023: Issue 1.3

Table of Contents

Publication history

The following table lists the 23.08 *Blue Planet Security Guide* publication history.

Table 1. Publication history

DATE	VERSION	NOTES
08/31/23	1.0	Initial 23.08 release
09/18/23	1.1	Updated `Using an SSL Certificate` section with user and host details.
10/30/23	1.2	Added BP Firewall section under the `General security procedures` chapter.
11/28/2023	1.3	Added a note under the `Configuring Custom Rules` section regarding restoring the bpfirewall customization after system bundle upgrade.

Overview

This document provides security guidelines for Blue Planet products. Blue Planet products can be deployed using the Blue Planet native platform, or using the Kubernetes opensource container orchestration platform. When you deploy Blue Planet products using the Blue Planet native platform, you use the Blue Planet installer (bpi) to deploy the BP platform and a lineup file to deploy the Blue Planet product. In this release, the document covers the guidelines for the following line of Blue Planet products deployed on Enterprise IT infrastructure (on-premises) using native platform only:

- Blue Planet Inventory (BPI)
- Blue Planet Orchestration (BPO)
- Blue Planet Route Optimzation and Assurance (BPROA)
- Blue Planet Unified Assurance and Analytics (BPUAA)
- BPI with Service Order Orchestrator (SOO)
- BPI with Multi Domain Service Orchestrator (MDSO)
- BPI with Discovery

In the following sections, the term BP product or BP2 site/platform/host/system will be used to represent prior listed products deployed using native platform in general.

NOTE This Security guide is applicable for BP2 platform environment only. Make sure to not apply any host based security settings to a Kubernetes environment.

General security procedures

You can manage security using the BP product UI. Security tasks relate to:

- Active sessions
- Configuration
- Geographical redundancy
- Permissions
- Roles
- Password policies and rules
- System backups
- User accounts

For more information refer to respective *Administrator Guides*.

The following procedures are provided to manage security:

- Using an SSL Certificate
- SSHD whitelisting
- Enabling port 123 for proxy NTP setup (Optional)
- Changing the nagios admin password

Using an SSL Certificate

To override a Secure Sockets Layer (SSL) certificate:

1. Login as a root user to perform this task.
2. Create the following directory on all hosts (both active and standby clusters), if it does not already exist.

   ```
   mkdir -p /etc/bp2/haproxy/ssl
   ```

3. Create a "pem" file by concatenating the ssl '.crt' and '.key' files:

   ```
   cat <domain>.crt <domain>.key > server.pem
   ```

4. Remove the default ssl certificate located in '/etc/bp2/haproxy/ssl':
5. Install the 'server.pem' file created above:

   ```
   mv server.pem /etc/bp2/haproxy/ssl/
   ```

   ```
   chmod 600 /etc/bp2/haproxy/ssl/server.pem
   ```

6. On a multi-host site, you can perform this manually on all hosts or run the following command to synchronize the haproxy configuration across all hosts:

   ```
   sudo bp2-site sync-site-config
   ```

7. Verify the above step:

   ```
   sudo bp2-site diff-site-config
   ```

8. Restart the haproxy containers using the solman CLI:

   (Change the name of the solution under which the haproxy is deployed, as needed.)

   For single node:

```
solution_app_restart artifactory.ciena.blueplanet.xp:<version> haproxy
```

For an HA node:

```
solution_app_restart artifactory.ciena.blueplanet.xp:ha-<version> haproxy
```

where <version> is the version of the platform solution.

SSHD whitelisting

During the course of installing any BP product in the BP2 environment, the installer automatically updates the `/etc/ssh/sshd_config` file on all hosts. nodes in a cluster. This is done to ensure that communication between nodes in the cluster or GR pair continue to work correctly if the hosts have any hardening applied.

You may still have to perform additional steps, see Host OS CIS Hardening section for more information.

In a BP2 cluster, the SSHD whitelist includes:

- IP address associated with each host in the cluster.
- Private (ILAN) subnets associated with each host in the cluster.

Example of lines added to the `/etc/ssh/sshd_config` file:

```
# ## BP2 ## {"tag_name": "BP_ROOT_SSH_WHITELIST_V1", "kind": "START"}
Match Address 10.186.0.149,10.186.0.90,10.186.1.104,172.16.0.0/24,172.16.1.0/24,
172.16.2.0/24
  PermitRootLogin without-password
# ## BP2 ## {"kind": "END", "tag_name": "BP_ROOT_SSH_WHITELIST_V1"}
```

Additionally, if GR is enabled there will be a second **Match Address** block with the IP addresses of the remote site. For example,

```
# ## BP2 ## {"tag_name": "BP_ROOT_SSH_WHITELIST_V1", "kind": "START"}
Match Address 10.78.69.176,10.78.69.205,10.78.69.229,172.16.0.0/24,172.16.1.0/24,
172.16.2.0/24
  PermitRootLogin without-password
# ## BP2 ## {"kind": "END", "tag_name": "BP_ROOT_SSH_WHITELIST_V1"}
# ## BP2 ## {"tag_name": "BP_REMOTE_ROOT_SSH_WHITELIST_V1", "kind": "START"}
Match Address 10.78.69.152,10.78.69.153,10.78.69.215,172.17.0.0/24, 172.17.1.0/24,
172.17.2.0/24
  PermitRootLogin without-password
# ## BP2 ## {"kind": "END", "tag_name": "BP_REMOTE_ROOT_SSH_WHITELIST_V1"}
```

Use the following two commands to test the status of the SSHD whitelist on a host:

- To verify access to local cluster:

  ```
  show-sshd-whitelist-status
  ```

- To verify access to remote site:

  ```
  show-remote-sshd-whitelist-status
  ```

These commands require root access – either login as root or use sudo with bpadmin. Example:

```
[bpadmin ~]$ sudo show-sshd-whitelist-status
SSHD Whitelist enabled: True
SSHD Whitelist status: config is as needed
[bpadmin ~]$ sudo show-remote-sshd-whitelist-status
Remote Sshd Whitelist Enabled for 10.78.69.152
Remote Sshd Whitelist status: config is as needed
[bpadmin ~]$
```

Enabling port 123 for proxy NTP setup (Optional)

This procedure supports network elements that use the BP2 system as an NTP server. By default, the bp-installer configures the hosts in the cluster to run the network time protocol daemon (ntpd) to synchronize against a list of remote time servers. This minimizes the clock drift between the hosts in the cluster. If you want to allow other hosts or systems to synchronize against the BP hosts using them as an ntp server, you must follow these steps to open the NTP port in the firewall for external access.

To configure the proxy NTP setup:

1. Log in to BP product as bpuser on Host 0.

   CAUTION Before you enable Port 123, which is a well-known NTP Port, on the BPFirewall, review and understand the security risks.

2. Ensure the Host OS CIS hardening has been performed to enable the BP Firewall.
3. On one of the hosts, create and edit the file /etc/bp2/site/bpfirewall and do a sync-site-config. This allows the bpfirewall to add Port 123 (NTP) to the iptables list of accepted ports.

   ```
   INPUT -p udp --dport 123 -j ACCEPT
   ```

4. Enter the commands below to sync the firewall changes to all the hosts in the cluster and to add new firewall rules to iptables by restarting the firewall service.

   ```
   sudo bp2-site sync-site-config
   bpssh sudo systemctl restart bpfirewall
   ```

5. Confirm that the bpfirewall rule was added to iptables using the command below. If successful, the output should match that shown below with the port "ntp" listed. If it returns nothing, this means that the change did not take effect and you need to investigate the bpfirewall service.

```
iptables -L | grep ntp
```

The output should match the text below.

```
ACCEPT udp – anywhere anywhere udp dpt:ntp
```

NOTE For deployments with geographical redundancy (GR), perform the steps 1 to 5 on both Primary and Backup sites.

Changing the nagios admin password

This section deals with changing the password for two default accounts:

- nagiosadmin
- adminpw

NOTE The procedure to change password is same for nagiosadmin and adminpw.

NOTE To change the password for a cluster, run the Sync config command after completing the Changing the password for active host procedure.

Changing the password for active host

Once nagios is activated for the first time, a default password is set in nagios. The login credential's are:

```
nagiosadmin/nagiosadmin
```

The nagios admin password can be changed in:

```
/etc/bp2/nagios/htpasswd.users
```

You can change the password as stated below:

- Using the htpasswd command
- Using the openssl command

Using the htpasswd command

Follow the below procedure to change the password if htpasswd is installed on your system:

1. Run the following command to update your password:

   ```
   htpasswd /etc/bp2/nagios/htpasswd.users nagiosadmin
   ```

2. Enter the password when prompted.

Using openssl command

Follow the below procedure to change the password with openssl command:

1. Generate the new nagios admin password by using the below command:

   ```
   root@localhost:~# openssl
   OpenSSL> passwd -apr1 mynewpassword
   $apr1$qOX4iXdY$laV4O8oNLKdJWLIh8uGHJ0
   OpenSSL> exit
   ```

2. Edit the below mentioned file:

   ```
   /etc/bp2/nagios/htpasswd.users
   ```

3. Update the password with the newly generated password:

   ```
   nagiosadmin:$apr1$qOX4iXdY$laV4O8oNLKdJWLIh8uGHJ0
   ```

Sync config

If you are using a cluster environment, you must push the Blue Planet configurations to other servers. This can be done by using the below command:

```
root@localhost:~# bp2-site sync-site-config
```

BP Firewall

BP Firewall is a systemd service that restricts access to host ports and is available in the bp2hosttools package. The bpfirewall service is designed to operate without interfering with the iptables rules required by docker and bp applications.

Other services that manage iptables, specifically firewalld and iptables.service, are incompatible with the correct operation of a site and must not be used.

NOTE The following procedure make reference to BP configuration files under the `/ect/bp2` default directory. If a site is configured with an alternate base config directory, the location of the referenced config file must be updated accordingly.

Functional Overview

When started, bpfirewall sets the default policy on the INPUT, FORWARD and OUTPUT chain to "DROP" and adds ACCEPT rules for ports needed for operation of the site and additional ACCEPT rules as per the `/etc/bp2/site/bpfirewall` config file.

When stopped, bpfirewall sets the default policy on the INPUT, FORWARD and OUTPUT chain to "ACCEPT" and removes ACCEPT rules that it had added when last started.

Configuring Custom Rules

The bpfirewall service can be configured to add additional rules to the INPUT, FORWARD and OUTPUT chains to allow access for non-standard, customer-specific purposes.

All custom rules must be added to the `/etc/bp2/site/bpfirewall` configuration file.

NOTE The `/etc/bp2/site/bpfirewall` file may be set back to default anytime a System Bundle is upgraded. For this reason, it is recommended that any customizations are also stored in a separate file (for example, bpfirewall.cust). After a System Bundle upgrade, the {{bpfirewall }} file should be reviewed to determine if the customizations should be re-applied.

Configuring an Open TCP and UDP port

Following is an example of `/etc/bp2/site/bpfirewall` file content for opening tcp port 5556 and udp port 8001 on INPUT chain and allow any packet leaving via output interface lo on OUTPUT chain.

```
INPUT -p tcp --dport 5556 -j ACCEPT  # Open tcp port 5556
-p udp --dport 8001 -j ACCEPT  # Open udp port 8001 (INPUT could be omitted)
OUTPUT -o lo -j ACCEPT
```

Activating bpfirewall Configuration Changes

After editing the bpfirewall configuration the following two steps are required in order to activate any changes.

1. The `/etc/bp2/site/bpfirewall` config file must be distributed to all of the hosts on the site using the following command:

   ```
   sudo bp2-site sync-site-config
   ```

2. The bp2firewall must be restarted on all hosts in the site using the following command:

   ```
   sudo bpssh systemctl restart bpfirewall
   ```

NOTE Custom rules should not be used to provide port forwarding to a bp app. All bp apps should utilize the natd service to enable any needed port access and redirection.

Life-cycle Management of the bpfirewall Service

The bpfirewall service is implemented as a systemd service. Standard systemd commands should be used to manage the bpfirewall service.

Table 2. Standard systemd commands

FUNCTION	COMMAND
Start the service	`sudo bpssh systemctl start bpfirewall.service`
Stop the service	`sudo bpssh systemctl stop bpfirewall.service`
Configuring bpfirewall to be Enabled on Start Up/Reboot	`sudo bpssh systemctl enable bpfirewall.service`

FUNCTION	COMMAND
Configuring bpfirewall to be Disabled on Start Up/Reboot	`sudo bpssh systemctl disable bpfirewall.service`
Getting Status	`bpssh systemctl status bpfirewall.service`
Instantaneous status information	`sudo /usr/local/bin/bpssh bpfirewall status`

NOTE Instantaneous status information must be queried by root directly from the init script since systemd only reports its own internal statistics plus messages generated by the service at start/stop.

Examples

Configuring an allow list for access to haproxy

To allow access to haproxy from IP addresses 1.1.1.1 and 2.2.2.2 only:

```
FORWARD -s 1.1.1.1 -p tcp --dport 443 -j ACCEPT
FORWARD -s 2.2.2.2 -p tcp --dport 443 -j ACCEPT
FORWARD -p tcp --dport 443 -j DROP
FORWARD -j ACCEPT
```

Moving base configuration directory to alternate location

NOTE This feature is not supported for ROA in the 23.08 release.

The default base configuration directory is `/etc/bp2`. Files located under this directory used to customise the configuration of Blue Planet applications. It is recommended to locate the base configuration directory to an alternate location if you require write access with security for Blue Planet applications. When you are configuring for a non-default base configuration directory all `bp2hosttools` provided commands that set or get site configuration use the alternate base configuration directory. Additionally, the solution manager replaces any host volume mounts starting with `/etc/bp2`, for the host side of the mount, with the directory specified by the base configuration directory. The container side of the mount is unchanged.

Example of a typical host volume mount of the default base configuration directory.

```
volumes:
       - /etc/bp2/site/:/etc/bp2/site
```

Example of host volume mount automatically changed when the base configuration directory is configured for `/opt/ciena/etc/bp2`.

```
volumes:
       - /opt/ciena/etc/bp2/site/:/etc/bp2/site
```

Configuring new site to alternate location

To configure new installation, without deployed solutions or existing containers, a site must be configured to use an alternate base configuration directory before running any commands that set configuration content under the base configuration directory.

To configure base configuration directory to alternate location during installation, user need to modify the `/home/bpadmin/bpi/playbooks/group_vars/all.yml` parameter before runing any installation command, the default configuration content of `all.yml` as below:

```
# Variables listed here are applicable to all cluster group and all roles

##########
# global #
##########
# BPI Version - Don't Update
bpi_version: 21.10-xx

# Name of the sudo limited BluePlanet install and maintenance user
ansible_ssh_user: bpuser

# Common directory for output from miscellaneous bpi commands - e.g. --get-logs
output_dir: ../target

# Log location when retrieving logs from hosts
log_dir: "{{ output_dir }}/logs"

# Storage location
storage_location: /opt/ciena

# bp2 data location for RedHat, Oracle Linux and CentOS installations
# the default - /bp2 will be used for Ubuntu installations regardless of the value
below
bp2_dir: "{{ storage_location }}/bp2"

# bp2 config location dir
bp2_config_dir: /etc/bp2
```

Example of `all.yml` after editing for changing default base configuration directory from `/etc/bp2` to `/opt/ciena/etc/bp2`.

```
# Variables listed here are applicable to all cluster group and all roles

##########
# global #
##########
# BPI Version - Don't Update
bpi_version: 21.10-xx

# Name of the sudo limited BluePlanet install and maintenance user
ansible_ssh_user: bpuser

# Common directory for output from miscellaneous bpi commands - e.g. --get-logs
output_dir: ../target

# Log location when retrieving logs from hosts
log_dir: "{{ output_dir }}/logs"

# Storage location
storage_location: /opt/ciena

# bp2 data location for RedHat, Oracle Linux and CentOS installations
# the default - /bp2 will be used for Ubuntu installations regardless of the value
below
bp2_dir: "{{ storage_location }}/bp2"

# bp2 config location dir
bp2_config_dir: /opt/ciena/etc/bp2
```

After changing to above content, then you can proceed the installation with `BPinstaller` command.

Existing site migration through back up restore with downtime

This procedure describes how to update an existing site's base config dir by bringing the entire site down. This procedure is the fastest way to migrate the entire site via backup and restore method. There will be application downtime during the duration of the procedure.

To update an existing site's base configuration directory:

1. Log in to host 0 of site as bpadmin

   NOTE Do not use the site IP, use the IP address of host 0.

2. Access the solution manager (solman) and back up your existing data.

```
sudo solman
```

3. Perform the backup.

```
site_backup --label <snapshot_name>
```

4. Undeploy all solutions.

```
sudo solman solution_undeploy ${SolutionName}:${SolutionVersion} --hard --purge
-host-vols -y
```

5. Perform the Host and core platform configuration.

6. Deploy all previously deployed solutions.

```
$ sudo solman solution_deploy
${ SolutionName }:${ SolutionVersion}
```

7. Scale solutions if they are required to be scale to 3.

```
$ sudo solman solution_scale ${ SolutionName }:${ SolutionVersion} 3
```

8. Restore site with previous backup snapshot file:

   a. Log in to host 0 of Site as bpadmin. Do not use the site IP; use the IP address of host 0.

   b. Stop all the solutions and remove its data volume:

   ```
   ./bpi --utility playbooks/solution-stopall.yml --playbook-args='-e
   solution_remove_volume=true'
   ```

   c. Ensure that you have sufficient additional disk space before starting the restore process.

   d. Restore the snapshot.

   ```
   $ sudo solman "site_restore <snapshot_name> --force"
   ```

   e. If some solutions are in stopped state. Make sure you start these solutions.

   ```
   $ solutions=`sudo solman sps | grep ^artifactory.ciena.com.blueplanet` ; for
   s in $solutions ; do sudo solman "solution_start $s" ; done ;
   ```

f. Verify that all solutions are started:

```
$ sudo solman sps
```

g. Verify that nagios in active site and ensure that status of all services is OK(green).

Running site migration host by host with no downtime on current site

This procedure describes how to update base configuration directory of a running site. In this procedure, host by host migration will happen and will have no downtime (two hosts are alive during migration).

To update base configuration directory of a running site:

1. Perform the Host and core platform configuration. This procedure prepares all site hosts for subsequent steps.
2. For each host, in series, perform the following steps:

a. Stop all container on the host being updated using the solman cli. This command will run for every solution:

```
solutions=`sudo solman sps | grep ^artifactory.ciena.com.blueplanet`; for s
in $solutions; do sudo solman "solution_stop $s --hosts <host-id>"; done;
```

b. Remove all containers from the host.

```
docker ps -qa | xargs docker rm -f
```

c. Install the core platform on the host if the host is a controller host (typically hosts 0, 1 and 2). Skip this step on non controller hosts.

```
bp2-site install --version <core-platform-version> --hosts <host-id>
```

d. Deploy solution containers to the host.

For each deployed solution:

```
solutions=`sudo solman sps | grep -P
^artifactory\.ciena\.com\.blueplanet\.[^[:cntrl:]]* -o`; for s in $solutions;
do echo "$s"; sudo solman "api solutions $s create_app_containers post
{appinsts: default}"; sudo solman "solution_deploy $s"; done;
```

e. Restore any non-default app scaling (not typically needed).

```
solution_app_scale <solution-name> <app-name> <instance-count>
```

f. Do not proceed until all Nagios alerts are cleared.

g. Repeat above steps until all hosts are migrated.

Host and core platform configuration

To configure host and core platform:

1. Run `bp2-site diff-site-config` and `bp2-site check-platform` to ensure that the site is problem free prior to proceeding with subsequent steps.
2. Install **bp2hosttools 0.4.47** (or newer) on all hosts.
3. Move old base config directory (typically /etc/bp2) to the new location and use the set-base-config-dir command to configure the new location. Perform this step on all hosts prior to proceeding to the next step.

NOTE On existing migrations, the `--force` option is required to allow the `set-base-config-dir` command function when containers are present. When run as part of a migration procedure described in this document, containers with host volume mounts that are broken by these steps, will be re-created to ensure correct operation in later steps.

a. Move the directory:

```
mv /etc/bp2 <new-location>
```

Example:

```
mv /etc/bp2 /opt/ciena/etc
```

   b. Perform the configuration:

      ```
      set-base-config-dir <new-location> --force
      ```

      Example:

      ```
      set-base-config-dir /opt/ciena/etc/bp2 --force
      ```

4. Verify the previous step has been correctly performed on all hosts by running `bp2-site check-platform`.
5. Install or re-install core platform version 21.06.02 (or newer). This step only needs to be run on one host as it automatically updates other hosts as needed.
   If the current core platform is older that 21.06.02, version 21.06.02 (or newer) must be install.

   ```
   bp2-site install --version <new-version>
   ```

   Example:

   ```
   bp2-site install --version 21.06.02
   ```

6. If core platform 21.06.02 (or newer) is already installed, the core platform must be redeployed.

   ```
   bp2-site redeploy
   ```

Configuring BP Products to use an external authentication server

This section focuses on how to configure BP products to use any of the following external authentication systems: RADIUS (through the BP product APIs), LDAP (through a command line interface), or TACACS (through a command line interface).

NOTE For RADIUS/LDAP users, please contact your system administrator to reset your password. The **forgot password** link is solely for local users.

Starting with release XP 22.02, tron supports multiple external authentication systems (LDAP, RADIUS, and TACACS) at a time. Tron authenticates against the configured external authentication systems in order until either there is a successful authentication or all methods have failed. The order of authentication is fixed: **LDAP Primary** > **LDAP Secondary** > **RADIUS Primary** > **RADIUS Secondary** > **TACACS Primary** > **TACACS Secondary** > **Local(Blue Planet)**.

For more information, see RADIUS, LDAP, and TACACS configuration.

This section includes:

- RADIUS/LDAP and Blue Planet Authentication
- RADIUS configuration
- LDAP configuration
- TACACS configuration
- RADIUS, LDAP, and TACACS configuration
- SAML configuration

NOTE

The Following cases are valid if a fallback to local authentication is not enabled: see RADIUS, LDAP, and TACACS configuration.

- If two or three (LDAP, RADIUS, and TACACS) external authentications are enabled, and the RADIUS or TACACS external servers are reachable, then admin and sysadmin users will be denied to login.
- If all three (LDAP, RADIUS, and TACACS) are enabled, and both RADIUS and TACACS servers are unreachable, then admin and sysadmin users are permitted to login.

RADIUS/LDAP and Blue Planet Authentication

The following procedure describes how RADIUS and LDAP authenticates its users in Blue Planet:

Data Synchronization

The below section describes Data Synchronization procedure:

1. User logs in to Blue Planet with RADIUS or LDAP-enabled password.
2. Blue Planet updates the user and provides a 24 hour token.
3. If the user is deleted from RADIUS/LDAP server, changes are reflected in Blue Planet on the next login. The current token is valid until expiration.

Authentication Order

The below section describes procedure to authenticate with RADIUS/LDAP server:

1. Blue Planet authentication service attempts to authenticate with RADIUS/LDAP server.
2. If there are both a primary server and a backup server, the order of authentication is primary than secondary.
3. Blue Planet user with a `sysadmin` role bypasses LDAP authentication and authenticates with local server directly.
4. Blue Planet user with `sysadmin` role bypasses RADIUS authentication and authenticates with local server directly when RADIUS server is not reachable.

HMAC Authentication

1. System administrator creates an API key for RADIUS/LDAP users using the Blue Planet UAC UI.
2. Users can perform API requests to UAC using HMAC authentication.
3. Blue Planet Server does not talk to RADIUS/LDAP server-side, we recommend the system administrator disables the user in Blue Planet using the UAC app to remove their API key. Otherwise, API keys associated with the disabled user remain active.

Review these important notes before configuring RADIUS or LDAP on the Blue Planet server:

- If a user logs in with Blue Planet UAC sysadmin privileges, authentication always succeeds based on local authentication, regardless if LDAP authentication is enabled or disabled.
- Ensure the radius-config or ldap-configs name is either *primary_config* or *backup_config*.

- Once RADIUS/LDAP is configured, you can authenticate to Blue Planet similar to a local user using RADIUS/LDAP credentials. You can configure RADIUS/LDAP for any tenant, as long as you include the tenant name in the radius-config or ldap-configs definition. Use the same tenant name to authenticate users. If you do not include a tenant name, the system defaults to the master tenant. Blue Planet RADIUS authentication supports single tenants only in this release. LDAP authentication supports multi-tenancy in this release.
- With RADIUS or LDAP enabled, if you log in with your Blue Planet user credentials (and are not a privileged sysadmin), authentication fails.
- To configure the LDAP role name feature to allow UAC to map LDAP role names to a specific role in a Blue Planet application, use the app. role_map JSON string. The string format is:

  ```
  `{"<LDAP_ROLE_NAME>": {"uac_role_name": "<ROLE_NAME_TO_MAP_TO>", "app_name":
  "<DISPLAY_NAME_OF_THE_APP>"}}`.
  ```

  For example: ``"roleMap":"{"Administrators": {"uac_role_name": "admin", "app_name": "UAC"}}"} RADIUS does not support the role_map feature.

RADIUS configuration

You can configure a RADIUS server for BP2 system by performing the following procedure.

- Integrate a RADIUS Server

Integrate a RADIUS server

The following topics are covered:

- Overview of RADIUS support
- External RADIUS BP2 System configuration
- Users and roles
- Use cases

Overview of RADIUS support

BP2 supports the delegation of authentication and authorization to an external customer-supplied RADIUS server. For more information about the RADIUS protocol, see https://tools.ietf.org/html/rfc2865.

RADIUS two-factor authentication

- BP2 supports the RADIUS Access-Challenge to allow 2FA as per RFC2865.

  If BP2 system receives an Access-Challenge reply in response to an Access-Request, BP2 system handles that and supports the specific optional attributes aligned with this message, like 'Reply-Message', 'State'.

- This Access-Challenge response is passed through to the user-facing client (BP product-UI or other REST interface).
- The user needs to provide a response code (token) in the password field and the State value as received from the RADIUS server. BP product creates a new Access-Request message with these values to the RADIUS server. The RADIUS server returns Access-Accept, Access-Reject or another Access-Challenge based on the validation result. If another Access-Challenge is received, the user is prompted for a valid response code (token) without ending the authentication session. Once the code is provided, BP product again sends another Access-Request method to send back to RADIUS server.
- If the authentication is successful, the user can now log successfully into BP product based on the privileges assigned to the user.
- NOTE: If the 'State' attribute is sent back in the Access-Challenge, the user must include the 'State' attribute (and its value) in the subsequent Access-Request through the user-facing client.

"Radius server unreachable" alarm

- The radius server unreachable alarm indicates when the site is unable to reach the configured RADIUS server. It indicates a separate alarm for the primary and secondary (backup) servers, if configured.
- The `heartbeat_user` is used for this purpose. Check External RADIUS BP2 System configuration about how to configure this parameter.
- If one is not setup, then a default blank user is used to test connectivity with the RADIUS server. This can function as is but may cause authentication failure logs at the RADIUS server. However, it proves that the connectivity to the site is ok, and hence no 'Radius server unreachable' alarm is raised.

Support limitations

- RADIUS Accounting (RFC 2866) is not supported.
- If no roles are specified, the following log message is generated: "No Role details found in RADIUS reply, user will have no Roles."
- Password policy settings under RADIUS should be set to the default (0) settings in order to allow all password related checks and settings to be handled by the RADIUS server.

Users and roles

The BP User details, including passwords, are stored on the respective BP product database.

BP product performs authentication (checks the user’s password) against the local database. With RADIUS mode enabled, BP product delegates authentication (password checking) to an external RADIUS server. BP product creates a local user, but without a password entry. This way the UAC user interface can list users, but does not allow setting of passwords for RADIUS users.

Local and RADIUS users can both exist in the same UAC instance, but not with the same usernames. UAC can also optionally accept authorization data from RADIUS. Authorization data is used to assign a list of rules to the user. Without the optional RADIUS authorization mode, the UAC administrator can:

- Specify a default set of roles for RADIUS users.
- Assign roles directly to users once the users logs into BP product.

External RADIUS BP2 System configuration

Two RADIUS profiles (primary and backup) and two TACACS are automatically created as part of BP product installation. You need to update their default configuration to setup the external RADIUS system as the authentication service for BP product.

In order to configure a primary and backup RADIUS server for BP product, you need to retrieve the uuid of the two default BP product RADIUS configuration profiles and then, for each of them, perform the configuration (IP, and so on) as indicated in the steps in this section.

Before you start, make sure that:

- You have the RADIUS IP address (if RADIUS redundancy is used, you need both the Primary and Backup RADIUS IP addresses),
- You have the RADIUS share secret to be established between RADIUS and BP product,
- The RADIUS server is configured with an admin privilege user/password, which can be used to log in (and/or revert the BP product RADIUS configuration back to local), before executing this procedure,
- If BP product is configured in redundant mode, to avoid latency in authentication requests, the “primary_config” RADIUS server is chosen/configured based on the proximity of the RADIUS server to the BP2 site.

CAUTION

Any misconfiguration or unreachability of RADIUS servers being configured with this procedure can block access to BP product. So it is recommended that you keep the admin user (local to BP product and having Application admin and sysadmin roles) credentials handy to recover from such a situation.

1. Login to BP product as a user having Security Admin privileges.
2. From the App Bar UI, navigate to **System > Platform > Swagger Ui**.
3. Select **UAC > radius-configs v1 > GET /api/v1/radius-configs**.
4. Click **Try it out** and click **Execute**.
5. Copy the `uuid` value of the `primary_config` from the response body.

Sample response

```
{
  "count": 2,
  "previous": null,
  "results": [
    {
      "description": "",
      "tenant": "master",
      "serverIp": "",
      "authport": 1812,
      "createdTime": "2019-09-27T21:45:59Z",
      "modifiedTime": "2019-09-27T21:45:59Z",
      "uuid": "c74ff04d-55f7-47ff-8b08-637fc8606209",
      "name": "primary_config",
      "enabled": false,
      "roleMap": {}",
      "authoritativeRoleSource": false
      "timeout": 10,
      "heartbeat_user": ""
    },
    {
      "description": "",
      "tenant": "master",
      "serverIp": "",
      "authport": 1812,
      "createdTime": "2019-09-27T21:45:59Z",
      "modifiedTime": "2019-09-27T21:45:59Z",
      "uuid": "140206d9-f98a-42c2-a66a-e5084dcae06e",
      "name": "backup_config",
      "enabled": false,
      "roleMap": {}",
      "authoritativeRoleSource": false
      "timeout": 10,
      "heartbeat_user": ""
    }
  ],
  "page": 1,
  "next" null
}
```

6. Click on radius-configs v1 → PATCH /api/v1/radius-configs/{uuid}
7. Click **Try it out**.
8. Enter the fields below in the form:

    - uuid - value as copied from step 5 for “primary_config”
    - server_ip – IP address of RADIUS server matching the specific profile
    - enabled – true
    - server secret – “shared secret” configured on the remote RADIUS server to be used for the BP2 client

   - authoritative_role_sources – true if the external RADIUS system will also perform user roles validation or false if the user roles will be authorized by the local BP2 security system
   - authport - Port on which RADIUS server is listening (default 1812).

   Leave other fields unchanged.

9. Click **Execute**. You will see a response similar to the one below:

   Sample response

   ```
   {
     "count": 1,
     "previous": null,
     "results": [
       {
         "description": "",
         "tenant": "master",
         "serverIp": "10.121.250.179",
         "createdTime": "2018-05-18T06:47:06Z",
         "modifiedTime": "2018-05-23T16:52:16Z",
         "uuid": "ad81b01c-2579-4b17-9283-76398461d9cd",
         "name": "primary_config",
         "enabled": true,
         "authoritativeRoleSource": true,
         "timeout": 10,
         "authport": 1812
       },
      "page": 1,
     "next": null
   }
   ```

10. Repeat Steps 8 and 9, replacing `primary_config` with `backup_config` if you are integrating with an external RADIUS server with redundancy.

Limitations of RADIUS or LDAP users

In BP2 system, RADIUS or LDAP users:

- Cannot be edited in the UAC user interface (Edit button is disabled).
- Cannot change their password in BP product (Change password, Reset password, Forgot password).
- Are visible on the Users page, but most fields are blank (first name, last name, email) as we do not retrieve these attributes from the RADIUS/LDAP database.
- Are visible on the Active Sessions page,
- Can have their sessions terminated just like local users,
- Must be configured with the "Ciena-Roles" attribute indicating their associated permissions, and

- Must have at least one role assigned.

Authenticating users

Once RADIUS is configured, you can authenticate in a similar manner to a local user using RADIUS credentials. You can only configure RADIUS for a single tenant and the tenant must be provided in the configuration request. The same tenant name must be used while authenticating users. If no tenant name is used, or not provided, the tenant name should default to that of the **master tenant**.

Sample authentication

```
  curl -k -H "Content-Type:application/json" -d
'{"username":"user1","password":"secretpass"}' https://localhost/tron/api/v1/tokens
```

Configuring RADIUS as the roles authority

To configure RADIUS as the roles authority, simply change "authoritativeRoleSource" to "true" in your request.

Roles to authentication

User roles can include both `system` user roles and `customer defined` user roles.

For more information about managing user roles, refer to respective *Blue Planet User Guide*.

NOTE The examples provided in this section are in the format of the FreeRADIUS server implementation. For more information about the FreeRADIUS server implementation, see https://freeradius.org/radiusd/man/users.html.

The system user roles are described as follows:

- Application admin
    - administration of Orchestrate
    - application-level security
    - basic visibility of network elements and services
- Observer
    - Read only access with basic visibility of network elements and services.
- Provisioner
    - Create, modify, or delete resources. Cannot create or delete domains and products.

- sysadmin
  - Master tenant sysadmin can create or deactivate a tenant. It works with Application Admin role.
- admin
  - All privileges except create tenants, deactivate tenants, create roles, and delete roles.
- UAC user
  - Privileges on identified tenant, can change user password.

Add the following to your "dictionary" configuration file:

```
#
#   Ciena specific codes
#
VENDOR          Ciena           1271
BEGIN-VENDOR    Ciena
ATTRIBUTE       Ciena-Roles   220     string
END-VENDOR      Ciena
```

This is a Multi-Valued attribute for inclusion in Access-Accept messages. If the attribute is present more than once in the Access-Accept, BP product adds the roles together.

Acceptable values are comma-separated strings matching (case-sensitively) any roles defined in BP product (these can be retrieved from the API with 'GET /tron/api/v1/roles/?isInternal=false'). Leading and trailing whitespace is removed and values that are not mapped to existing roles are ignored.

Sample users

```
user1   Cleartext-Password := "secretpass"
        Ciena-Roles = "Application admin, sysadmin",
user2   Cleartext-Password := "secretpass2"
        Ciena-Roles = "Observer, UAC user",
BP2user1   Cleartext-Password := "secret1"
        Ciena-Roles = "Application admin"
BP2user4   Cleartext-Password := "secret4"
        Ciena-Roles = "Observer"
```

NOTE Role names must match the role names in the roles list or they are ignored. Roles are selected according to the role.name and the default application with the following exceptions:

- The role.name is "sysadmin" and the application is "UAC". If the role doesn’t exist, the default is used.
- The role.name is "admin" and the application is "UAC". If the role doesn’t exist, the default is used.

- The role.name is "user" and the application is "UAC". If the role doesn’t exist, the default is used.

Limitations or assumptions

1. When RADIUS is enabled, the UAC local database user roles are deactivated except users with role UAC:sysadmin, UAC:admin and internal users.
2. The best way to ensure proper external RADIUS integration is to test with existing and valid RADIUS servers.
3. The RADIUS configuration names **primary_config** and **backup_config** are pre-created within BP2 system;

Data synchronization

When a user tries to log in with RADIUS enabled, the user object is updated if RADIUS authentication is successful. UAC only communicates with RADIUS during password-based authentication. The token created during user login expires after 24 hours or a configured timeout. Disabling the user in RADIUS does not cause the token to expire before the timeout. After the token has expired, the user must log in again.

NOTE	A password-based login attempt updates a RADIUS user’s enable/disable status if that RADIUS user account’s control attribute indicates disabled or the user is deleted from the RADIUS server.

Authentication order

When RADIUS is enabled, UAC first tries to authenticate with the RADIUS server. You can specify one primary RADIUS server and one backup RADIUS server. When both RADIUS servers are enabled, UAC attempts authentication in order of: primary, secondary, then local.

HMAC authentication

A system administrator is able to create an API key for RADIUS users in the UAC user interface. Users are then able to perform API requests to UAC using HMAC authentication. UAC does not communicate with RADIUS for HMAC authentication. If a user is disabled on the RADIUS server, the system administrator is recommended to disable the user or delete the API keys associated with the disabled user. Otherwise, API keys associated with the disabled user remain active.

Use cases

RADIUS enabled

1. RADIUS enabled, login with valid RADIUS credentials:
    - an application user opens the BP product URL
    - BP product redirects the user to the UAC login page.
    - the BP2 user provides a valid RADIUS credential
    - the user redirects back to BP product page
    - login succeeds
2. RADIUS enabled, login with invalid RADIUS credentials
    - an application user provides a valid RADIUS username but invalid password
    - login fails
3. RADIUS enabled, login with native normal user credentials fails.
4. RADIUS enabled, RADIUS server is down
    - an application user provides valid RADIUS credentials
    - login fails
5. RADIUS enabled, RADIUS server is down
    - any native UAC admin or sysadmin user attempts to login
    - login succeeds (after a longer login time)

RADIUS disabled

1. RADIUS disabled, application user provides valid RADIUS credentials
    - login fails
2. RADIUS disabled, application user provides invalid credential
    - login fails
3. RADIUS disabled, login with native sysadmin credentials succeeds.
4. RADIUS disabled, login with native non-sysadmin credentials succeeds.

Get an admin token. Use this to configure RADIUS.

```
  PREFIX=https://localhost/tron
  TOKEN=$(curl \
              --silent -k \
              -H "Content-Type:application/json" \
              -d '{"username":"admin","password":"adminpw"}' \
              $PREFIX/api/v1/tokens | \
                  python -c "import sys, json; print
json.load(sys.stdin)['token']")
  echo $TOKEN
```

Display the current state of the RADIUS configuration

```
  curl --silent -k -H "Authorization: token $TOKEN" $PREFIX/api/v1/radius-configs

  RADIUS_UUID=$(curl \
                    --silent -k \
                    -H "Authorization: token $TOKEN" \
                    $PREFIX/api/v1/radius-configs | \
                        python -c "import sys, json; print
json.load(sys.stdin)['results'][0]['uuid']")

  curl \
      --silent -k \
      -H "Authorization: token $TOKEN" \
      $PREFIX/api/v1/radius-configs/$RADIUS_UUID | \
          python -c "import sys, json, pprint; pprint.pprint(json.load(sys.stdin))"
```

Configure UAC to point to the local RADIUS server in the Docker container

```
  RADIUS_IP=$(docker inspect -f '{{range
.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' freeradius_1.1.0_0)
  curl \
      --silent -k \
      -H "Authorization: token $TOKEN" \
      -H "Content-Type:application/json" \
      -X PATCH \
      -d '{"name": "primary_config", "enabled": true, "authoritative_role_source":
true, "server_secret": "supersecret", "server_ip": "'$RADIUS_IP'", "tenant":
"master", "description": "primary radius config", "heartbeat_user":
"radiusheartbeatuser", "heartbeat_pwd": "secret"}' \
      $PREFIX/api/v1/radius-configs/$RADIUS_UUID

  curl \
      --silent -k \
      -H "Authorization: token $TOKEN" \
      $PREFIX/api/v1/radius-configs/$RADIUS_UUID | \
          python -c "import sys, json, pprint; pprint.pprint(json.load(sys.stdin))"
```

Display the current state of the users

```
  curl -k -H "Content-Type: application/json" -H "Authorization: token $TOKEN"
$PREFIX/api/v1/users | python -m json.tool
```

Get a token as a RADIUS mastered user

```
  curl
      -k
      -H "Content-Type:application/json"
      -d '{"username":"user2","password":"secretpass2"}'
      $PREFIX/api/v1/tokens

  curl
      -k
      -H "Content-Type:application/json"
      -d '{"username":"user1","password":"secretpass"}'
      $PREFIX/api/v1/tokens
```

Display the new state of the users

```
  curl
      -k
      -H "Content-Type:application/json"
      -H "Authorization: token $TOKEN" $PREFIX/api/v1/users | python -m json.tool
  curl
      -k
      -H "Content-Type:application/json"
      -H "Authorization: token $TOKEN" $PREFIX/api/v1/users\?username\=user1 |
python -m json.tool
```

LDAP configuration

Local and LDAP users can both exist in the same UAC instance, but not with the same usernames. When BP2 system is configured to point to an external LDAP server, Local (native) users are not allowed to login except UAC admin/sysadmin user. BP2 system supports authentication of users using an external LDAP server. Both LDAP and LDAPS (LDAP over SSL) are supported.

For setting up BP product to use with an external LDAP server and to install a new LDAP server, follow the below steps

1. Determine if an existing corporate LDAP server will be used (for example, your company may already have an Active Directory with corporate users defined), or if a new LDAP server instance will be installed/configured (for example, using freeware such as OpenLDAP or other 3rd-party software)
2. In BP product (UI or through API commands), decide on and define the user roles that are needed. All user role details and the permissions that are associated with them are still defined in BP product (not on the LDAP server). If you intend to customize the permissions within roles, you need to either clone the existing default roles or create new roles (the default roles defined in BP product are read-only).
3. If using an existing corporate LDAP:

a. Collect the required configuration parameters from your LDAP administrator.

b. Create groups matching the roles that exist and will be used in BP product.

> **NOTE** User group is supported in tron (BP product), only through API to set client inactivity time and token expiration.

- This involves simply defining user group names within the LDAP.
- The simplest option is to create one LDAP user group for every BP product role that exists. However, you can also create an LDAP user group that is associated with a union of multiple roles on BP product.
- Any name can be used on the LDAP server for the user groups. A simple approach is to simply reuse the role name as defined on BP product, with the BP product appended abbreviation to the beginning (for example, BPO_Observer).

c. Add existing LDAP users to the groups defined above.

d. Configure BP product (tron process) to point to the LDAP server.

- Use BP product API commands to set all of the the configuration parameters necessary to contact the LDAP server.
- Use BP product API commands to define the mapping of the LDAP user group name to the BP product role name with which it will be associated.

If installing a new LDAP server:

4. Follow the 3rd-party software documentation to install the LDAP server (includes setting up logging options, defining domain, LDAP administrator setup and access privileges, organization and organization role creation, and so on)
5. Create groups that match the roles that exist and that will be used in BP product.
   - This involves simply defining user group names within the LDAP.
   - The simplest option is to create one LDAP user group for every role that exists. However you can also create an LDAP user group that will be associated to a union of multiple roles on BP product.
   - Any name can be used on the LDAP server for the user groups. A simple approach is to simply reuse the role name as defined on BP product, with the BP product abbreviation appended to the beginning (for example, BPO_Observer).
6. Create users for every user that will need to login to BP product.
   - This involves defining the username in the LDAP, and associating the username with one or more user group names that were defined on the LDAP.

7. Configure BP product (tron process) to point to the LDAP server.
    - Use BP product API commands to set all of the configuration parameters necessary to contact the LDAP server.
    - Use BP product API commands to define the mapping of the LDAP user group name to the BP product role name with which it will be associated.

The following LDAP configuration parameters need to be entered when using the BP product API command to set the LDAP details. You must gather these parameters before BP product can be pointed to an LDAP sever (either from your LDAP administrator or from the person who installed/configured the LDAP server).

PARAMETER	DESCRIPTION	EXAMPLE
server_ip	IP address of server, including whether SSL will be used, and whether it is on default or custom port	Example (non-SSL, default LDAP port) ldap://10.132.241.111 Example (SSL, custom port 1636): ldaps://10.132.241.111:1636
enable_ssl	Leave this as false	-
ssl_level	Leave at default ALLOW (more restrictive options are supported for customers who implement self-signed certificates; contact Ciena for details)	-
domain_search_user domain_search_password	Username and password allowing queries to LDAP sever (only if required, many LDAP setups do not require LDAP user authentication)	-
base_dn	The base domain defined for the LDAP server	Example: dc=pbany1mm,dc=ott,dc=ciena,dc=com
user_name_attribute	The name of the attribute on the LDAP server that contains the username	Example: uid
tenant_attribute	The name of the attribute on the LDAP server that contains the tenant name.	BPtenant

PARAMETER	DESCRIPTION	EXAMPLE
group_name_attribute	The name of the attribute on the LDAP server that contains the groupname	Example: cn
group_object_filter	The filter to use defining which object type to filter on when authenticating	Example: (objectClass=posixGroup)
role_map	The mapping of LDAP group names to role names.	Example (assuming default roles are used on BPO; and there is 1 corresponding role defined in LDAP for each of these; and the group names defined on LDAP use the same name but with BPO pre-pended and dashes replacing the spaces in the names): {\"BPO-UAC-admin\": {\"uac_role_name\":\"UAC admin\",\"app_name\": \"UAC\"}, \"BPO-UAC-user\": {\"uac_role_name\":\"UAC user\",\"app_name\": \"UAC\"}, \"BPO-UAC-sysadmin\": {\"uac_role_name\":\"UAC sysadmin\",\"app_name\": \"UAC\"}, \"BPO-Application-admin\": {\"uac_role_name\":\"Application admin\",\"app_name\": \"BluePlanet\"}, \"BPO-Observer\": {\"uac_role_name\":\"Observer\",\"app_name\": \"BluePlanet\"}}

The following topics are covered:

- Configure LDAP directory
- Integrate an LDAP server

Configure LDAP directory

This procedure provides examples of some commands that can be used to configure a 3rd-party OpenLDAP server. Refer to your 3rd-party documentation for complete details.

Tailor the LDAP logging

1. Set the loglevel to suit your needs, for example:

   ```
   ldapmodify -Q -Y external -H ldapi:/// <<EOF
   dn: cn=config
   changetype: modify
   replace: olcLogLevel
   olcLogLevel: config stats none
   EOF
   ```

   The logs are sent to syslog facility local4 by default.

If your LDAP server is running on RedHat, Oracle Linux or CentOS OS, some structures must be created manually.

2. Enter the following:

   ```
   cd /etc/openldap/schema
    ldapadd -Q -Y external -H ldapi:/// -f cosine.ldif
    ldapadd -Q -Y external -H ldapi:/// -f inetorgperson.ldif
    ldapadd -Q -Y external -H ldapi:/// -f nis.ldif

    ldapmodify -QY external -H ldapi:/// <<EOF
    dn: olcDatabase={2}hdb,cn=config
    changetype: modify
    replace: olcSuffix
    olcSuffix: dc=lonlab,dc=ciena,dc=com
    -
    replace: olcRootDN
    olcRootDN: cn=admin,dc=lonlab,dc=ciena,dc=com
    EOF
   ```

3. Check the olcDatabase configuration:

   ```
   ldapsearch -QY EXTERNAL -H ldapi:/// -b cn=config
   '(objectClass=olcDatabaseConfig)'
   ```

4. Take note of the olcRootPW, and update it with slappasswd if necessary:

```
ldapadd -x -D "cn=admin,dc=lonlab,dc=ciena,dc=com" -w ciena123 -H
ldap://localhost $lt;%lt;EOF
 dn: dc=lonlab,dc=ciena,dc=com
 objectClass: dcObject
 objectClass: organization
 dc: lonlab
 o: Ciena
 description: lonlab test LDAP instance

 dn: cn=admin,dc=lonlab,dc=ciena,dc=com
 objectClass: organizationalRole
 cn: admin
 description: Directory Manager
 EOF
```

Create an attribute for assigning the Tenant

5. The tenant could also be stored in a pre-defined attribute (for example, from the inetorgperson or NIS schemas) if you want to avoid creating custom schemas. It is clearer to create a dedicated attribute though.

```
ldapadd -Q -Y external -H ldapi:/// <<EOF
dn: cn=CienaBP,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: CienaBP
olcAttributetypes: {0}( 2.3.6.1.4.1.1271.28533.1794.1
        NAME 'BPtenant'
        DESC 'Ciena BluePlanet Tenant name'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
olcObjectClasses: {0}( 1.3.6.1.4.1.1271.28533.1794.128
        NAME 'BluePlanetUser'
        DESC 'Ciena BluePlanetUser object class'
        SUP top
        AUXILIARY
        MUST ( BPtenant ) )
EOF
```

- The names are arbitrary, feel free to adjust them as required (but remember to use them consistently).
- The OID values must be unique within the scope of the LDAP instance, it is very strongly advised to abide by IANA allocation rules. The above example uses "Enterprise"."Ciena", with subdivision 28533 and followed by 1794 to reduce the risk of clashes.
- See LDAP schema documentation for further information. The definition above is just an example, it is by no means the only way a tenant attribute can be defined.

6. To see the result of your work:

```
ldapsearch -Q -Y external -H ldapi:/// -b cn=schema,cn=config
"(&(objectClass=olcSchemaConfig)(cn=*CienaBP))"
```

NOTE LDAP adds in a schema index number.

Create the Organizational Units for the users and groups

7. Enter the following:

```
ldapadd -x -D "cn=admin,dc=lonlab,dc=ciena,dc=com" -W <<EOF

dn: ou=People,dc=lonlab,dc=ciena,dc=com
objectClass: organizationalUnit
ou: People

dn: ou=Group,dc=lonlab,dc=ciena,dc=com
objectClass: organizationalUnit
ou: Group
EOF
```

- The empty line between both records is mandatory, or you can split this into separate ldapadd operations. You can also use ldapmodify to add objects.

Integrate an LDAP server

This section provides some examples of how to create groups and users on a 3rd-party OpenLDAP server. Refer to your 3rd-party documentation for complete details. This section also contains an example of how to point BP product to the LDAP server and create the role mapping between LDAP groups and BP product roles.

The following topics are covered:

- Creating groups
- Creating users
- Configuring the LDAP server on Blue Planet
- Associating LDAP groups with BP Product roles

Creating groups

NOTE Grid numbers must be unique in the scope of the LDAP instance.

An example follows on how to create a one-to-one mapping for the factory-default roles in UAC (as names are arbitrary).

To create groups:

1. Enter the following:

```
ldapmodify -x -D "cn=admin,dc=lonlab,dc=ciena,dc=com" -W <<EOF
dn: cn=BP-UAC-Admin,ou=Group,dc=lonlab,dc=ciena,dc=com
changetype: add
objectClass: posixGroup
gidNumber: 32129
cn: BP-UAC-Admin

dn: cn=BP-UAC-User,ou=Group,dc=lonlab,dc=ciena,dc=com
changetype: add
objectClass: posixGroup
gidNumber: 32130
cn: BP-UAC-User

dn: cn=BP-Application-Admin,ou=Group,dc=lonlab,dc=ciena,dc=com
changetype: add
objectClass: posixGroup
gidNumber: 32257
cn: BP-Application-Admin

dn: cn=BP-Application-Observer,ou=Group,dc=lonlab,dc=ciena,dc=com
changetype: add
objectClass: posixGroup
gidNumber: 32258
cn: BP-Application-Observer

dn: cn=BP-Network-admin,ou=Group,dc=lonlab,dc=ciena,dc=com
changetype: add
objectClass: posixGroup
gidNumber: 32259
cn: BP-Network-admin

dn: cn=BP-Planner-Admin,ou=Group,dc=lonlab,dc=ciena,dc=com
changetype: add
objectClass: posixGroup
gidNumber: 32260
cn: BP-Planner-Admin

dn: cn=BP-UAC-sysadmin,ou=Group,dc=lonlab,dc=ciena,dc=com
changetype: add
objectClass: posixGroup
gidNumber: 32261
cn: BP-UAC-sysadmin
EOF
```

2. If custom groups are defined in Tron, you can also create them in LDAP.

NOTE See the procedure Associating LDAP groups with BP Product roles (assigning multiple roles to a single group).

Creating users

NOTE uidNumbers must be unique in the scope of the LDAP instance.

To create users:

1. Enter the following:

```
NEWUID=16385
NEWUSER=bpuser1

ldapmodify -x -D "cn=admin,dc=lonlab,dc=ciena,dc=com" -W <<EOF
dn: uid=$NEWUSER,ou=People,dc=lonlab,dc=ciena,dc=com
changetype: add
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
objectClass: BluePlanet
uid: $NEWUSER
mail: bptest1@lonlab.ciena.com
sn: $NEWUSER
givenName: user
cn: $NEWUSER
displayName: $NEWUSER
uidNumber: $NEWUID
gidNumber: $NEWUID
userPassword: secret
gecos: $NEWUSER
loginShell: /bin/bash
homeDirectory: /home/$NEWUSER
BPtenant: master

dn: cn=BP-Application-Admin,ou=Group,dc=lonlab,dc=ciena,dc=com
changetype: modify
add: memberUid
memberUid: $NEWUSER
EOF
```

2. You can add the $NEWUSER to multiple groups; just add the memberUid to all the groups that are needed.

NOTE Tron requires the mail attribute which must be unique. You can omit all other fields that are optional for the posixAccount objectClass, although sn and givenName are shown in the BP Web UI and make for easy identification of user records.

Configuring the LDAP server on Blue Planet

Blue Planet ships with standard primary and backup configurations. To configure LDAP, update these configurations and enable or disable LDAP.

Ciena recommends you use the Swagger UI to configure LDAP as it is less error prone. Select **System > Platform > Swagger Ui** to access Blue Planet APIs. Then select **UAC** to find the `ldap-configs v1` commands.

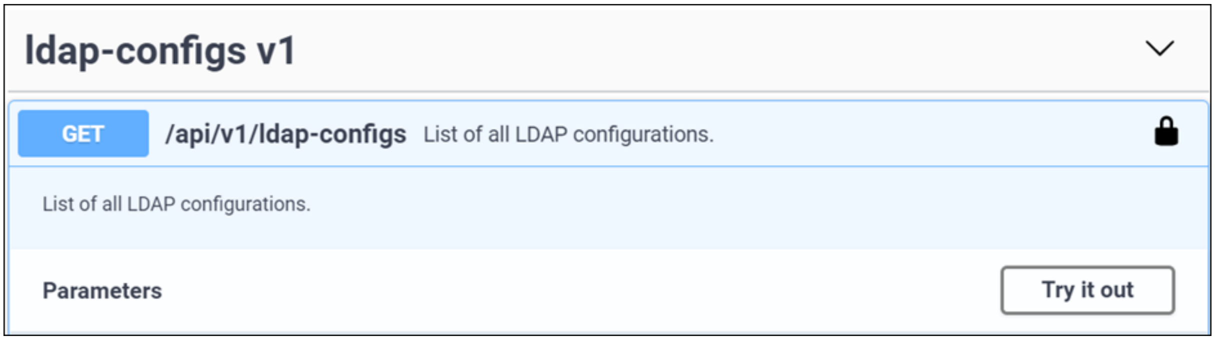
IMPORTANT	Ciena recommends that you gather the required LDAP information above, then manually perform an LDAP search to ensure your search is successful. Use any of the LDAP search tools available online. If your search is unsuccessful and the information is not retrieved from the LDAP server, in some cases one or more attributes are incorrect. It is also required that all users have valid email addresses in the LDAP server.

For example, search an existing user account on an Active Directory server using filter uid=<name> by entering:

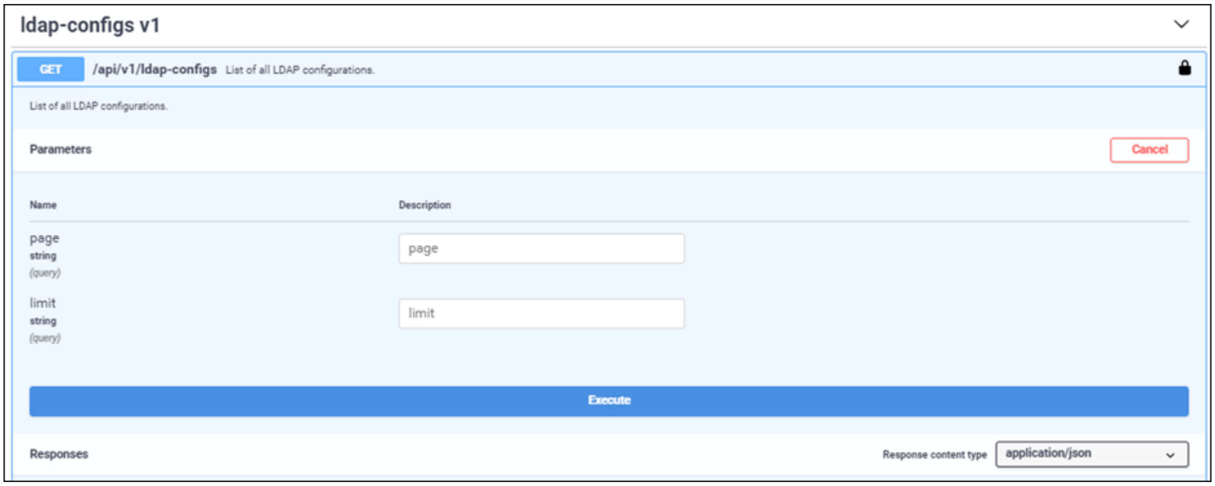
```
ldapsearch -x -H ldap://10.205.33.106 -D “user1@ad.ciena.com" -LLL -b
dc=ad,dc=ciena,dc=com -w NewPassword@ "(uid=user1)"
the server does not return an entry because the correct filter is
“(sAMAccountName=user1).”
```

If you choose to use Swagger UAC API to configure your LDAP server settings, see the procedure below.

1. Login to BP product as a user having Security Admin privileges.
2. From the App Bar UI, navigate to **System > Platform > Swagger Ui**.
3. Select **UAC > ldap-configs v1 > GET /api/v1/ldap-configs**.


ldap-configs v1
GET
/api/v1/ldap-configs
List of all LDAP configurations.
List of all LDAP configurations.
Parameters
Try it out

4. Click **Try it out** and then click **Execute**.



ldap-configs v1

GET /api/v1/ldap-configs List of all LDAP configurations.

List of all LDAP configurations.

Parameters

Name	Description
page string (query)	page
limit string (query)	limit

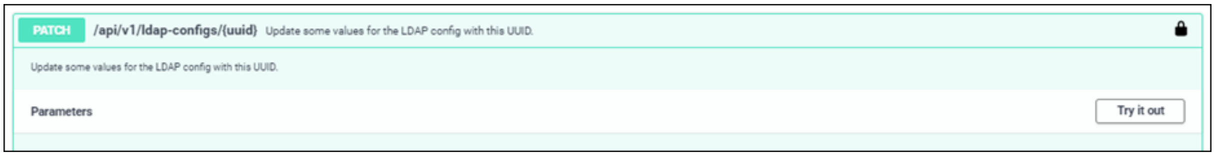
5. Copy the `uuid` value of the `primary_config` from the response body.

Sample response:

```
{
"count": 2,
"previous": null,
"results": [
{
"description": "",
"createdTime": "2018-11-16T00:33:14Z",
"modifiedTime": "2018-11-16T00:45:26Z",
"uuid": "ff1fecd5-731c-4a4e-bed1-37e2b652712f",
"name": "primary_config",
"enabled": true,
"serverIp": "ldap://10.200.33.106",
"enableSsl": false,
"sslLevel": "ALLOW",
"domainSearchUser": "CIENA\\userxyz",
"enableReferrals": false,
"baseDn": "DC=ciena,DC=com",
"userNameAttribute": "sAMAccountName",
"tenantAttribute": "",
"accessibleTenantsAttribute": "",
"groupNameAttribute": "cn",
"groupObjectFilter": "(objectClass=Group)",
"roleMap": "{\"list.PQA
Team\":{\"app_name\":\"BluePlanet\",\"uac_role_name\":\"Application admin\"}}"
},
{
"description": "",
"createdTime": "2018-06-13T19:30:58Z",
"modifiedTime": "2018-06-13T19:30:58Z",
"uuid": "f23b974c-93dd-4df3-95e4-30a405ad94cf",
"name": "backup_config",
"enabled": false,
"serverIp": "",
"enableSsl": false,
"sslLevel": "ALLOW",
"domainSearchUser": "",
"enableReferrals": false,
"baseDn": "",
"userNameAttribute": "sAMAccountName",
"tenantAttribute": "",
"accessibleTenantsAttribute": "companyA,companyB",
"groupNameAttribute": "cn",
"groupObjectFilter": "(objectClass=Group)",
"roleMap": "{}"
}
],
"page": 1,
"next": null
}
```

6. Select **ldap-configs v1 > PATCH /api/v1/ldap-configs/{uuid}**.

7. Click **Try it out**.



PATCH /api/v1/ldap-configs/{uuid} Update some values for the LDAP config with this UUID.

Update some values for the LDAP config with this UUID.

Parameters

8. The following list describes the required LDAP information you must set in the `ldap-configs` form in Blue Planet. Enter the parameters fields below in the form:

   - server_ip : The IP address of the LDAP server. For example, `ldap://<LDAP Server IP>`.
   - enabled : Enabled state of the config is true.
   - domain_search_user : The bind-DN username to perform LDAP search operations. Most systems allow any LDAP user to perform search operation. Otherwise, contact your company’s LDAP administrator to configure this step.

     NOTE The bind-DN format uses comma-separated Relative Distinguished Names (RDNs) such as “cn=manager,dc=ciena,dc=com” or email address-like format such as “Administrator@AD” or user1@ad.ciena.com, depending on the LDAP server.

   - domain_search_password: Password for binding user specified in `domain_search_user` field.
   - base_dn : The base dn format uses comma separated RDNs such as `dc=ciena,dc=com`, or `dc=ad,dc=cyaninc,dc=com`, or `cn=users,dc=cyanoptics,dc=com`, depending on the LDAP server setup.
   - user_name_attribute : The user naming attribute for the user account. For an Active Directory server, the attribute is `sAMAccountName`. For other types of servers or databases, the attribute is `uid`. It is critical to set this attribute correctly.
   - enable_referrals : Ciena recommends you do not change the LDAP referral default, `enableReferrals=false`. This setting works with Microsoft Active Directory. Microsoft AD does not support anonymous authentication on LDAP referral, so this attribute must be disabled.
   - tenant_attribute : Ciena recommends that you use the description attribute, but you can choose another attribute. You must replace this value with an available attribute from the LDAP user account (such as `description`).
   - accessible_tenants_attribute : If you have multiple tenants, to store Blue Planet tenant information use the `accessible_tenants` attribute in the LDAP user account. To use multiple tenants, configure the `accessible_tenants_attribute` in UAC via Swagger to list the tenants separated by commas. For more information see, Configuring multi-tenant access for LDAP users.
   - group_name_attribute : Attribute to load group name. default='cn'
   - group_object_filter : Filter when searching group. default='(objectClass=Group)'

- role_map : Since tron 11.0.9, you can map a single LDAP group to multiple UAC privileges using a list structure.

For example:

```
'role_map': {
{"ldap_group": [{"uac_role_name": "non-existent-role111", "app_name": "Planet
Orchestrate"}, {"uac_role_name": "user", "app_name": "Planet Orchestrate"}]}
}
```

9. Click **Execute**. You will see a response similar to the one below:

Sample Response:

```
{
     "description": "",
     "createdTime": "2020-04-08T13:24:07Z",
     "modifiedTime": "2020-04-17T07:05:26Z",
     "uuid": "0f8b5551-30df-4422-95f7-abe917b85de0",
     "name": "primary_config",
     "enabled": true,
     "serverIp": "ldap://172.16.0.105",
     "enableSsl": false,
     "sslLevel": "ALLOW",
     "domainSearchUser": "Administrator@AD",
     "enableReferrals": false,
     "baseDn": "dc=ad,dc=cyaninc,dc=com",
     "userNameAttribute": "sAMAccountName",
     "tenantAttribute": "description",
     "accessibleTenantsAttribute": "company",
     "groupNameAttribute": "cn",
     "groupObjectFilter": "(objectClass=Group)",
     "roleMap":
"{\"Administrators\":[{\"app_name\":\"BluePlanet\",\"uac_role_name\":\"Application
admin\"},{\"app_name\":\"UAC\",\"uac_role_name\":\"UAC sysadmin\"}]}"
}
```

10. Repeat Steps 8 and 9, replacing `primary_config` with `backup_config` if you are integrating with an external LDAP server with redundancy.

If you choose not to use the Swagger UAC API to configure your LDAP server settings, see the procedure below for an alternate example.

To configure your LDAP server using the Blue Planet API commands in your terminal window:

1. Create a token key for the Blue Planet server to communicate with the LDAP server.

```
curl -k -H "Content-Type:application/json" -d
'{"username":"admin","password":"adminpw", "tenant":"master"}' -X POST
https://10.206.30.107/tron/api/v1/tokens | python -m json.tool
```

A response containing text similar to the following displays:

```
{
"createdTime": "2016-05-11T17:30:05.812645Z",
"failedLoginAttempts": 0,
"lastSuccessIpAddress": "10.206.30.107",
"lastSuccessLogin": "2016-11-05 17:29:58+00:00",
"sessionId": "85f45768-7223-4bed-a9b6-92725208a3a5",
"timeout": 86400,
"token": "78b97d242d3bcac14b87",
"user": "5d5c14e2-96f4-45dd-9937-84d0d6cc9c40"
}
```

where the Blue Planet server with IP address 10.206.30.107 creates the token "78b97d242d3bcac14b87". You can also replace the IP address in the command with the fully-qualified domain name (FQDN).

2. Get the lapd-configs from the Blue Planet UAC app using the token you created in step 1:

```
curl -H "Content-Type: application/json" -k -H "Authorization: token
78b97d242d3bcac14b87" -X GET https://10.200.33.106/tron/api/v1/ldap-configs |
python -m json.tool
```

A response similar to the following displays:

```
{
  "count": 2,
  "previous": null,
  "results": [
  {
   "description": "",
   "createdTime": "2018-11-16T00:33:14Z",
   "modifiedTime": "2018-11-16T00:45:26Z",
   "uuid": "ff1fecd5-731c-4a4e-bed1-37e2b652712f",
   "name": "primary_config",
   "enabled": true,
   "serverIp": "ldap://10.200.33.106",
   "enableSsl": false,
   "sslLevel": "ALLOW",
   "domainSearchUser": "CIENA\\userxyz",
   "enableReferrals": false,
   "baseDn": "DC=ciena,DC=com",
   "userNameAttribute": "sAMAccountName",
   "tenantAttribute": "",
   "accessibleTenantsAttribute": "",
   "groupNameAttribute": "cn",
   "groupObjectFilter": "(objectClass=Group)",
   "roleMap": "{\"list.PQA
Team\":{\"app_name\":\"BluePlanet\",\"uac_role_name\":\"Application admin\"}}"
 },
    {
      "description": "",
      "createdTime": "2018-06-13T19:30:58Z",
      "modifiedTime": "2018-06-13T19:30:58Z",
      "uuid": "f23b974c-93dd-4df3-95e4-30a405ad94cf",
      "name": "backup_config",
      "enabled": false,
      "serverIp": "",
      "enableSsl": false,
      "sslLevel": "ALLOW",
      "domainSearchUser": "",
      "enableReferrals": false,
      "baseDn": "",
      "userNameAttribute": "sAMAccountName",
      "tenantAttribute": "",
      "accessibleTenantsAttribute": "companyA,companyB",
      "groupNameAttribute": "cn",
      "groupObjectFilter": "(objectClass=Group)",
      "roleMap": "{}"
    }
  ],
  "page": 1,
  "next": null
}
```

NOTE Many attributes in the *ldap-configs* are empty or contain a default value; for example, enabled is false and serverIp is empty.

3. Set the ldap config to match with those set on the LDAP server.

You can use one of the following three examples to help you determine the configuration that works best in your situation.

NOTE There is a space between the curl command and the URL. You can substitute details to create your command for the backup_config as needed.

Example 1.

This multi-tenant example uses *sAMAccountName* for the user_name_attribute; *Administrator@AD* and *My?Passwrd0* as the domain_search_user and domain_search_password for an LDAP search operation (which belongs to the LDAP server administrator); and the value of the description attribute on the user account to store tenant information on the Blue Planet server. The server names this user can access are listed in the accessible_tenants_attribute. The server_ip attribute must include the ldap:// prefix.

Create the JSON file, then run your PATCH command.

```
cat <<EOF > ldap-configs.json
{
"enabled":true,
"server_ip":"ldap://10.60.11.122",
"domain_search_user": "Administrator@AD",
"domain_search_password": " My?Passwrd0",
"base_dn": "dc=ad,dc=cyaninc,dc=com",
"user_name_attribute": "sAMAccountName",
"tenant_attribute":"description",
"accessible_tenants_attribute": "Company A,Company B,Company C",
"group_name_attribute": "cn",
"role_map": "{}"
}
EOF

curl -H "Content-Type: application/json" -k -H "Authorization: token
78b97d242d3bcac14b87" -X PATCH -d @ldap-configs.json
https://10.206.30.107/tron/api/v1/ldap-configs/88eebd1b-5496-4ba0-92c1-
fc68eea51ce2/ | python -m json.tool
```

A response similar to the following displays:

```
{
"baseDn": "dc=ad,dc=cyaninc,dc=com",
"createdTime": "2016-03-24T02:09:36Z",
"description": "",
"domainSearchUser": "Administrator@AD",
"enableSsl": false,
"enabled": true,
"groupNameAttribute": "cn",
"groupObjectFilter": "(objectClass=Group)",
"modifiedTime": "2016-03-24T02:09:36Z",
"name": "primary_config",
"roleMap": "{}",
"serverIp": "ldap://10.60.11.122",
"tenantAttribute": "description",
"userNameAttribute": "sAMAccountName",
//"accessibleTenantsAttribute": "Company A,Company B, Company C",
"uuid": "88eebd1b-5496-4ba0-92c1-fc68eea51ce2"
}
```

Example 2.

For a single host, use an existing regular LDAP user account for the LDAP search operation. Create the JSON file, then run your PATCH command.

```
cat <<EOF > ldap-configs.json
{
"enabled":true,
"server_ip":"ldap://10.60.11.122",
"domain_search_user": "user@ad.cyaninc.com",
"domain_search_password": "NewPassword@",
"base_dn": "dc=ad,dc=cyaninc,dc=com",
"user_name_attribute": "sAMAccountName",
"accessible_tenants_attribute": "",
"group_name enabled _attribute": "cn",
"role_map": "{}"
}
EOF

curl -H "Content-Type: application/json" -k -H "Authorization: token
78b97d242d3bcac14b87" -X PATCH -d @ldap-configs.json
https://10.206.30.107/tron/api/v1/ldap-configs/88eebd1b-5496-4ba0-92c1-
fc68eea51ce2/ | python -m json.tool
```

A response similar to the following displays:

```
{
"baseDn": "dc=ad,dc=cyaninc,dc=com",
"createdTime": "2016-03-24T02:09:36Z",
"description": "",
"domainSearchUser": "user1@ad.cyaninc.com",
"enableSsl": false,
"": true,
"groupNameAttribute": "cn",
"groupObjectFilter": "(objectClass=Group)",
"modifiedTime": "2016-03-24T02:09:36Z",
"name": "primary_config",
"roleMap": "{}",
"serverIp": "ldap://10.60.11.122",
"tenantAttribute": "",
"userNameAttribute": "sAMAccountName",
"accessibleTenantsAttribute": "",
"uuid": "88eebd1b-5496-4ba0-92c1-fc68eea51ce2"
}
```

Example 3.

For a multi-tenant system, use the user naming attribute, *uid*, the LDAP server is not an Active Directory server, and the bind-dn and bind-password for an LDAP search operation are *cn=manager,dc=ciena,dc=com* and *secret* and belong to the LDAP server administrator. Create the JSON file, then run your PATCH command.

```
{
"enabled":true,
"server_ip":"ldap://10.15.3.80",
"domain_search_user": "cn=manager,dc=ciena,dc=com",
"domain_search_password": "secret",
"base_dn": "dc=ciena,dc=com",
"user_name_attribute": "uid",
"tenant_attribute":"description",
"group_name_attribute": "cn",
"role_map": "{}"
}
EOF

curl -H "Content-Type: application/json" -k -H "Authorization: token
78b97d242d3bcac14b87" -X PATCH -d @ldap-configs.json
https://10.206.30.107/tron/api/v1/ldap-configs/88eebd1b-5496-4ba0-92c1-
fc68eea51ce2/| python -m json.tool
```

A response similar to the following displays:

```
{
"baseDn": "dc=ciena,dc=com",
"createdTime": "2016-03-24T02:09:36Z",
"description": "",
"domainSearchUser": "cn=manager,dc=ciena,dc=com",
"enableSsl": false,
"enabled": true,
"groupNameAttribute": "cn",
"groupObjectFilter": "(objectClass=Group)",
"modifiedTime": "2016-03-24T02:09:36Z",
"name": "primary_config",
"roleMap": "{}",
"serverIp": "ldap://10.15.3.80",
"tenantAttribute": "description",
"userNameAttribute": "uid",
"uuid": "88eebd1b-5496-4ba0-92c1-fc68eea51ce2"
}
```

4. Set the LDAP user account.

   a. Ensure the user account for the LDAP user has the following attributes set correctly.

      - An attribute with a “tenant” value in which the user belongs. For ldap-configs, our previous example used the attribute called `description`.
      - A unique email address. A non-unique email-address or empty email address causes issues.

      You may need to consult with your company LDAP administrator to set these attributes.

   b. Use the following LDAP user account examples to correctly set your attribute values:

      Example 1.
      Where the LDAP server is an Active Directory server and the administrator has created a tenant named *tenant3* on the Blue Planet server.

      ```
      dn: CN=user7,CN=Users,DC=ad,DC=cyaninc,DC=com
      objectClass: top
      objectClass: user
      description: tenant3
      sAMAccountName: user7
      mail: user7@example.com
      ```

      The user, *user7*, can log in to *tenant3* of Blue Planet Orchestration as long as there are no other users with the email address *user7@example.com* on *tenant3*.

Example 2.

Where the LDAP server is not an Active Directory server and the administrator has created a tenant named *tenant1* on the Blue Planet server.

```
dn: cn=Nelson User1,ou=people,dc=ciena,dc=com
objectClass: inetOrgPerson
objectClass: top
uid: user1
mail: user1@example.com
ou: Tester
description: tenant1
```

The user, *user1*, can log in to *tenant1* of Blue Planet Orchestration as long as there are no other users with the email address *user1@example.com* on *tenant1*.

5. To assign a UAC role to the user that logs in using an LDAP password, configure the LDAP user to be a member of a specific group on the LDAP server, and then use the `role_map` attribute of ldap-configs to map the group into a UAC role. For example, configure the LDAP user to be a member of Administrators group, and then map the Administrators group to the UAC role "admin" using the `role_map` attribute.

   Create the JSON file, then run your PATCH command.

   ```
   cat <<EOF > ldap-configs.json
   {
    "enabled":true, "server_ip":"ldap://10.60.11.122",
    "domain_search_user": "user1@ad.cyaninc.com", "domain_search_password":
   "NewPassword@", "base_dn": "dc=ad,dc=cyaninc,dc=com", "user_name_attribute":
   "sAMAccountName", "tenant_attribute":"description", "group_name_attribute":
   "cn", "role_map": "{\"Administrators\": {\"uac_role_name\":
   \"admin\",\"app_name\": \"UAC\"}}"
   }
   EOF
   ```

The following example command sets the ldap-configs `role_map` attribute (use snake_case format for “role_map”).

```
curl -H "Content-Type: application/json" -k -H "Authorization: token
efaba9a98a97f82d97dd" -X PATCH d @ldap-configs.json
https://10.206.30.107/tron/api/v1/ldap-configs/88eebd1b-5496-4ba0-92c1-
fc68eea51ce2/ | python -m json.tool
```

A response similar to the following displays:

```
{
  "baseDn": "dc=ciena,dc=com",
  "createdTime": "2016-03-24T02:09:36Z",
  "description": "",
  "domainSearchUser": "cn=manager,dc=ciena,dc=com",
  "enableSsl": false,
  "enabled": true,
  "groupNameAttribute": "cn",
  "groupObjectFilter": "(objectClass=Group)",
  "modifiedTime": "2016-03-24T02:09:36Z",
  "name": "primary_config",
  "roleMap": "{}",
  "serverIp": "ldap://10.15.3.80",
  "tenantAttribute": "description",
  "userNameAttribute": "uid",
  "uuid": "88eebd1b-5496-4ba0-92c1-fc68eea51ce2"
}
```

You must configure your user to be a member of the Administrators group in the LDAP system so UAC can assign the UAC role “admin”; for example:

```
dn: CN=user10,CN=Users,DC=ad,DC=cyaninc,DC=com
objectClass: top
objectClass: user
description: tenant3
sAMAccountName: user10
mail: user10@example.com
memberOf: CN=Administrators
```

6. To verify successful configuration, log in to Blue Planet using your LDAP user credentials.

7. To disable the LDAP server authentication, you must patch the *ldap-configs*. For example, to change the enabled command to **false** enter the following:

   ```
   curl -H "Content-Type: application/json" -k -H "Authorization: token
   efaba9a98a97f82d97dd" -X PATCH -d '{"enabled":false}'
   http://<bp_server_ip>/tron/api/v1/ldap-configs/88eebd1b-5496-4ba0-92c1-
   fc68eea51ce2/
   ```

   Ensure this command string appears as a continuous string with a space before the URL.

Associating LDAP groups with BP Product roles

To associate LDAP groups with BP product roles:

Assigning multiple roles with a single LDAP group

1. Since tron 11.0.9, you can map a single LDAP group to multiple UAC privileges using a list structure, for example:

   ```
   'role_map': {

     {"ldap_group": [{"uac_role_name": "non-existent-role111", "app_name": "Planet
   Orchestrate"}, {"uac_role_name": "user", "app_name": "Planet Orchestrate"}]}

   }
   ```

Undefined roles

2. If LDAP users with assigned undefined roles log in to BP, these roles are created implicitly on the fly, with no permissions assigned to the role.

Associating LDAP groups with BP Product Usergroups

To associate LDAP groups with BP product Usergroups, use the following guidelines:

- Create a user in LDAP and add this user to LDAP group.

To map LDAP users and usergroups using Swagger Ui:

1. Login to BP product as a user having **security admin** or **sysadmin** privileges.
2. From the App Bar UI, navigate to **System > Platform**.
3. Select **Swagger Ui**.

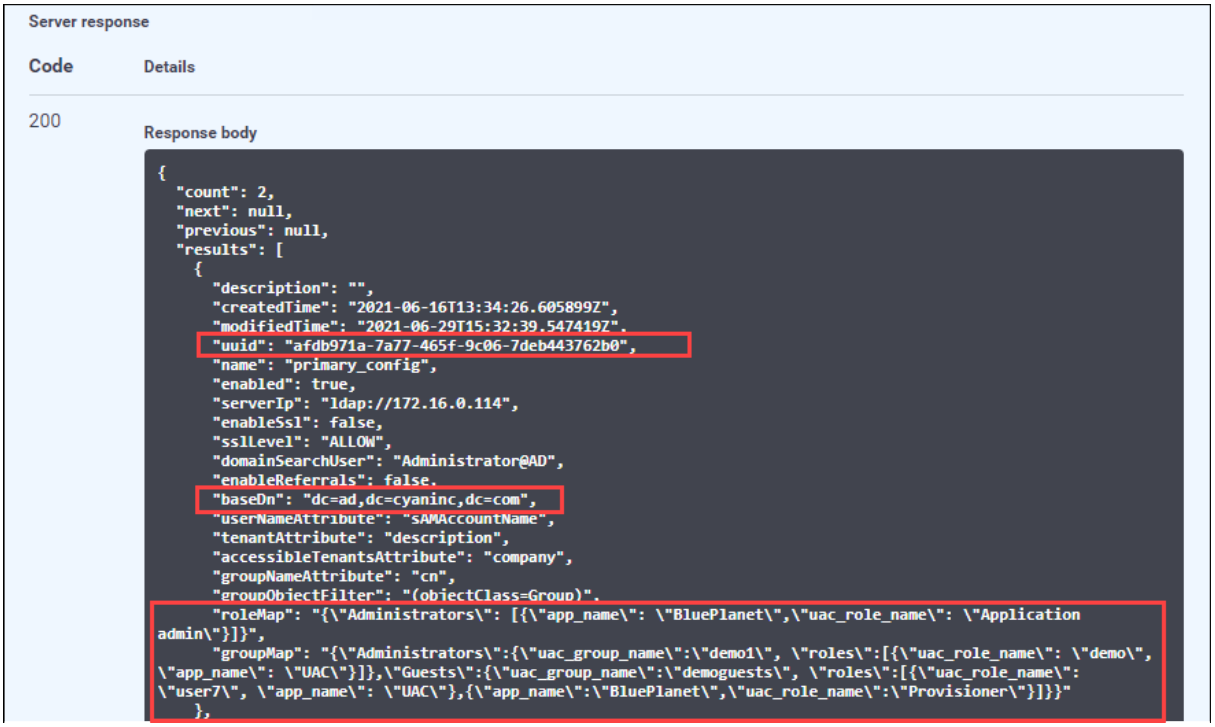
   The **Blue Planet APIs** page opens.

4. Click **UAC > ldap-configs v1 > GET /api/v1/ldap-configs List of all LDAP configurations.**.
5. Click **Try it out**.
6. Click **Execute**.
   The **roleMap** and **groupMap** fields appear in the response.

```
"roleMap": "{\"Administrators\": [{\"app_name\":
\"BluePlanet\",\"uac_role_name\": \"Application admin\"}]}",
    "groupMap": "{\"Administrators\":{\"uac_group_name\":\"demo1\",
\"roles\":[{\"uac_role_name\": \"demo\", \"app_name\":
\"UAC\"}]},\"Guests\":{\"uac_group_name\":\"demoguests\",
\"roles\":[{\"uac_role_name\": \"user7\", \"app_name\":
\"UAC\"},{\"app_name\":\"BluePlanet\",\"uac_role_name\":\"Provisioner\"}]}}"
  },
```

7. Copy the **baseDn** value from the response.
8. Copy the **uuid** value from the response.

NOTE In Administrators group, the configured role is 'admin' and application name is 'UAC', and LDAP users in Administrators group that automatically gets the admin role privileges. In Guests group, the configured role is 'user' and 'provisioner' so the LDAP users in the group Guests have permissions or privileges of 'user' and 'provisioner' roles.



```
{
  "count": 2,
  "next": null,
  "previous": null,
  "results": [
    {
      "description": "",
      "createdTime": "2021-06-16T13:34:26.605899Z",
      "modifiedTime": "2021-06-29T15:32:39.547419Z",
      "uuid": "afdb971a-7a77-465f-9c06-7deb443762b0",
      "name": "primary_config",
      "enabled": true,
      "serverIp": "ldap://172.16.0.114",
      "enableSsl": false,
      "sslLevel": "ALLOW",
      "domainSearchUser": "Administrator@AD",
      "enableReferrals": false,
      "baseDn": "dc=ad,dc=cyaninc,dc=com",
      "userNameAttribute": "sAMAccountName",
      "tenantAttribute": "description",
      "accessibleTenantsAttribute": "company",
      "groupNameAttribute": "cn",
      "groupObjectFilter": "(objectClass=Group)",
      "roleMap": "{\"Administrators\": [{\"app_name\": \"BluePlanet\",\"uac_role_name\": \"Application
admin\"}]}",
      "groupMap": "{\"Administrators\":{\"uac_group_name\":\"demo1\", \"roles\":[{\"uac_role_name\": \"demo\",
\"app_name\": \"UAC\"}]},\"Guests\":{\"uac_group_name\":\"demoguests\", \"roles\":[{\"uac_role_name\":
\"user7\", \"app_name\": \"UAC\"},{\"app_name\":\"BluePlanet\",\"uac_role_name\":\"Provisioner\"}]}}"
    },
```

9. Click **PATCH /api/v1/ldap-configs/{uuid} Update some values for the LDAP config with this UUID**.
10. Click **Try it out**.
11. In the **Parameters** section, in the **uuid** field, enter the unique id copied from step 8.
12. In the **description** field, enter the LDAP configuration description.

13. In the **enabled** field, select **true** value for activating LDAP configuration.
14. In the **server_ip** field, enter the server ip address.
    For example, **ldap://<ip of ldap server>**.
15. In the **group_map** field, enter the group map strings to map LDAP groups to tron groups.
16. Enter the other optional attributes as shown in the table below:

FIELD	DESCRIPTION
name	LDAP configuration name
enable_ssl	The enable_ssl value. The allowed values are "true" or "false".
ssl_level	SSL security level. The allowed values are "NEVER", "ALLOW", or "DEMAND".
domain_search_user	LDAP username to perform user lookups. Example: user@domain.com
domain_search_password	Password for domain search user
enable_referrals	Microsoft AD needs this to be disabled. Anonymous bind is not supported. The allowed values are "true" or "false".
base_dn	The base domain defined for the LDAP server
user_name_attribute	Attribute to load user name
tenant_attribute	Tenant attribute to filter user
accessible_tenants_attribute	Attribute to define multiple tenant access for this user
group_name_attribute	Attribute to load group name
group_object_filter	Filter when searching group
role_map	Dictionary string that maps LDAP groups or roles to a list of tron roles.

17. Click **Execute**.
    The roles that are added in the **group_map** field in step 15 are assigned to the usergroup and user on the BP Product-UI. You can validate the roles for users and usergroups by selecting **System > Platform > Swagger Ui**. For more information about user and usegroups, see the **User and system setup** chapter in respective *BP Administrator Guide*.

Configuring multi-tenant access for LDAP users

This section describes how to configure administrative rights for an individual LDAP user to enable access to a custom set of multiple tenants or sub-tenants. This capability allows users to temporarily change the admin rights of an LDAP user from one set of multiple tenants into another set of multiple tenants by updating a setting in the LDAP account of the associated employee.

For example, employee A has admin rights to Tenant1, Tenant2 and Tenant3; while employee B has admin rights to Tenant4 and Tenant5. These rights are set in each employee’s LDAP account. When employee B goes on vacation, an administrator can change employee A’s LDAP account temporarily to allow admin rights to Tenant4 and Tenant5 until employee B returns from vacation.

On the BP2 system, a minor setting is required in the UAC’s LDAP configuration. Ciena recommends you use the Swagger UI to perform this task.

This topic includes:

- Requirements for LDAP multi-tenant access
- Accessible tenant configuration guidelines
- Example of LDAP account with an attribute to store multi-tenant access information
- Configuring ldap-configs using Swagger
- Logging into the system after configuring LDAP accessibleTenantsAttribute

Requirements for LDAP multi-tenant access

- Set up the LDAP account on the LDAP server to support multi-tenancy. To perform the procedure to set up your LDAP server, see Configuring the LDAP server on Blue Planet.

  The LDAP account must have an attribute to store the information for the home tenant of the user, and this attribute is mapped into the `tenantAttribute` field of the UAC’s ldap-configs REST API.

- Add an attribute in the LDAP account to store the list of tenants that can be accessed by the user. The attribute name is configurable, so admins may use their discretion. If circumstances prevent updates to the LDAP schema, use any unused attribute in the LDAP user account for this purpose; for example, the `description` attribute. This attribute is used to work with (or map) the field `accessibleTenantAttribute` of UAC’s ldap-configs.
- Configure each LDAP account on the LDAP server to include the newly named tenant list attribute which contains the list the tenant names that can be accessed by the LDAP user.

Accessible tenant configuration guidelines

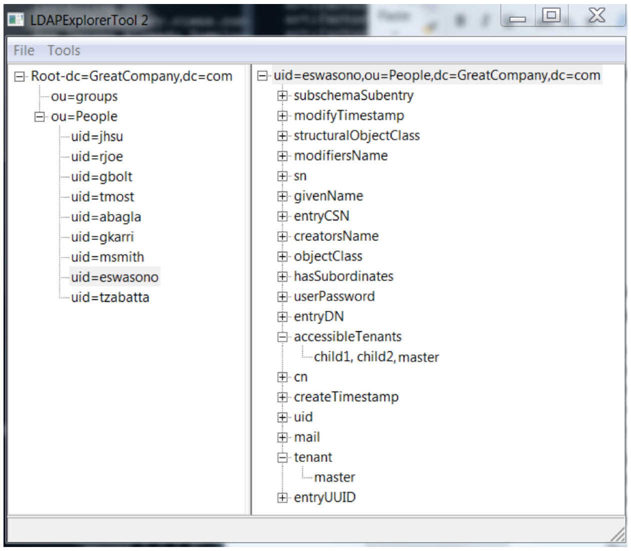
- Ensure that the attribute which stores the list of tenants (for example, accessibleTenants) for the LDAP account includes the home tenant, and the home tenant is accessible by the user.
- Any tenant, including same level tenants, parent tenant or even master tenant can be put in the LDAP account’s accessibleTenants or whatever attribute that is used for this purpose, regardless of the home tenant of the user.
- Even if the LDAP account of a user does not have the accessible Tenants named attribute, the behaviour of the tenant-context still works and the user continues to have access to the home tenant, and its sub-tenants if any. The same scenario exists if the ldap-configs’ accessibleTenantAttribute is left empty; all LDAP users will continue to be able to access their home tenant and its sub-tenants, if available.

Example of LDAP account with an attribute to store multi-tenant access information

The example depicts the online LDAP Explorer Tool displaying an LDAP account on an OpenLDAP server that has the `accessibleTenants` attribute. The `accessibleTenants` attribute stores the information for the list of tenants that can be accessed by the user. Blue Planet allows users to configure this attribute, so note that it is called "accessibleTenants" in this example. As depicted, the attribute contains a comma-separated tenants list that includes master, child1 and child2 tenants.

The LDAP account also includes a `tenant` attribute to store the information for the home tenant of the user. In this case, the home tenant of the user is master. You must configure the BP2 server to have the same structure of multi-tenancy in order to use this LDAP server. In other words, ensure that any tenant names are configured in the BP2 server; in this case, the child1 and child2 tenants.

LDAP configuration using the LDAPExplorerTool



```
⊟ Root-dc=GreatCompany,dc=com
    ou=groups
  ⊟ ou=People
      uid=jhsu
      uid=rjoe
      uid=gbolt
      uid=tmost
      uid=abagla
      uid=gkarri
      uid=msmith
      uid=eswasono
      uid=tzabatta
```

```
⊟ uid=eswasono,ou=People,dc=GreatCompany,dc=com
  ⊞ subschemaSubentry
  ⊞ modifyTimestamp
  ⊞ structuralObjectClass
  ⊞ modifiersName
  ⊞ sn
  ⊞ givenName
  ⊞ entryCSN
  ⊞ creatorsName
  ⊞ objectClass
  ⊞ hasSubordinates
  ⊞ userPassword
  ⊞ entryDN
  ⊟ accessibleTenants
      child1, child2, master
  ⊞ cn
  ⊞ createTimestamp
  ⊞ uid
  ⊞ mail
  ⊟ tenant
      master
  ⊞ entryUUID
```

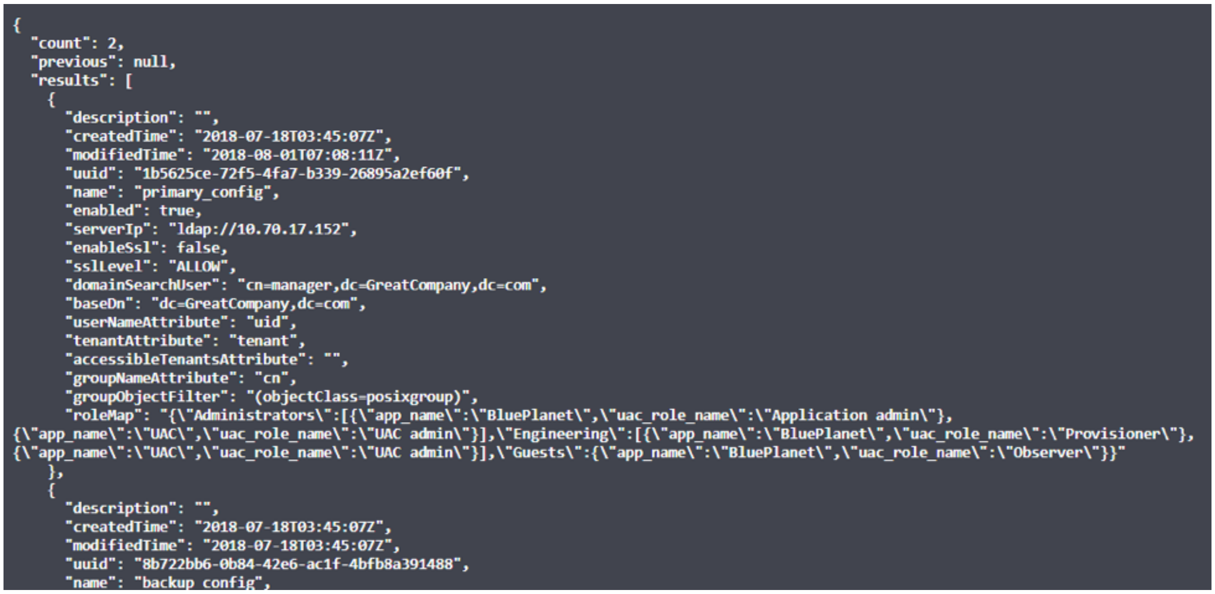
Configuring ldap-configs using Swagger

To configure multi-tenant access for LDAP user in Blue Planet:

1. Select **System > Platform > Swagger Ui** to access the Swagger APIs.
2. Select the **UAC** category and expand the **ldap-configs v1** section by clicking in that row.
3. To display a list of LDAP configurations, select **GET /api/v1/ldap-configs**.
4. Click **Try it out** then to execute the command, click **Execute**.

   Swagger displays two sets of ldap-configs; primary and backup. This section describes how to configure the primary config. You can configure the backup config in the same way. The following figure depicts a LDAP configuration for an OpenLDAP server without this feature configured.

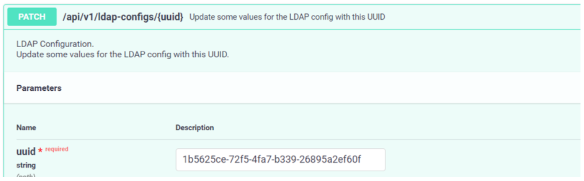
   LDAP configuration prior to accessibleTenants configuration

```
{
  "count": 2,
  "previous": null,
  "results": [
    {
      "description": "",
      "createdTime": "2018-07-18T03:45:07Z",
      "modifiedTime": "2018-08-01T07:08:11Z",
      "uuid": "1b5625ce-72f5-4fa7-b339-26895a2ef60f",
      "name": "primary_config",
      "enabled": true,
      "serverIp": "ldap://10.70.17.152",
      "enableSsl": false,
      "sslLevel": "ALLOW",
      "domainSearchUser": "cn=manager,dc=GreatCompany,dc=com",
      "baseDn": "dc=GreatCompany,dc=com",
      "userNameAttribute": "uid",
      "tenantAttribute": "tenant",
      "accessibleTenantsAttribute": "",
      "groupNameAttribute": "cn",
      "groupObjectFilter": "(objectClass=posixgroup)",
      "roleMap": "{\"Administrators\":[{\"app_name\":\"BluePlanet\",\"uac_role_name\":\"Application admin\"},
{\"app_name\":\"UAC\",\"uac_role_name\":\"UAC admin\"}],\"Engineering\":[{\"app_name\":\"BluePlanet\",\"uac_role_name\":\"Provisioner\"},
{\"app_name\":\"UAC\",\"uac_role_name\":\"UAC admin\"}],\"Guests\":{\"app_name\":\"BluePlanet\",\"uac_role_name\":\"Observer\"}}"
    },
    {
      "description": "",
      "createdTime": "2018-07-18T03:45:07Z",
      "modifiedTime": "2018-07-18T03:45:07Z",
      "uuid": "8b722bb6-0b84-42e6-ac1f-4bfb8a391488",
      "name": "backup_config",
```

   In the figure, the new accessibleTenantAttribute field is empty in the ldap-configs output. To configure, you must populate this field as follows. Note that the system is configured with multi-tenancy after you populate the tenantAttribute field with the word you previously configured in the LDAP account's in the LDAP server, in this example, tenant.

5. Scroll down, click **PATCH** and then click **Try it out**.
6. Populate the `uuid` field with the value of the uuid from your GET command output. For this example, the uuid from our example is 1b5625ce-72f5-4fa7-b339-26895a2ef60f. You can highlight the value and copy and paste it into the uuid field.



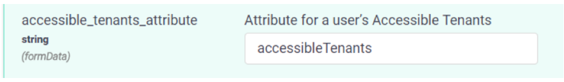
PATCH /api/v1/ldap-configs/{uuid} Update some values for the LDAP config with this UUID

LDAP Configuration.
Update some values for the LDAP config with this UUID.

Parameters

Name	Description
uuid * required string (path)	1b5625ce-72f5-4fa7-b339-26895a2ef60f

7. Populate the accessible_tenants_attribute field with the word accessibleTenants (from the LDAP account in the LDAP server in Figure 1). Notice that the PATCH section uses snake case format, while the GET section uses camelcase format for ldap-configs' fields, which is acceptable.



accessible_tenants_attribute string (formData)	Attribute for a user's Accessible Tenants accessibleTenants

8. Scroll down to click **Execute**.

   A successful response returns a value of 200, if no issues exist.

9. To verify the change, scroll up, and rerun the GET command as described in step 3.

Logging into the system after configuring LDAP accessibleTenantsAttribute

After you perform the configuration procedures for configuring accessible tenants, the LDAP user login can access the BP2 system on the home tenant to which the user belongs using credentials authenticated with the LDAP server. If authentication succeeds, the accessible tenants list gets transferred from the LDAP server. If the tenants have sub-tenants, UAC adds these sub-tenants to the list. This list is then displayed by the login UI and allows the user to select from this tenant-context list in the second login window.

NOTE When using the accessible Tenants attribute, make sure to include the home tenant in the list because it is not completely added.

TACACS configuration

TACACS configuration is same as the RADIUS server except for roles. The TACACS+ server does not support roles. Tron support for TACACS+ allows the administrator to configure default roles for new users. To do this, add the role names to the `role_map` field in the configuration re-code.

The following shows an example map:

Syntax

A json dictionary in this form:

```
{'default': { 'rolenames': [, , <etc...>]}}
```

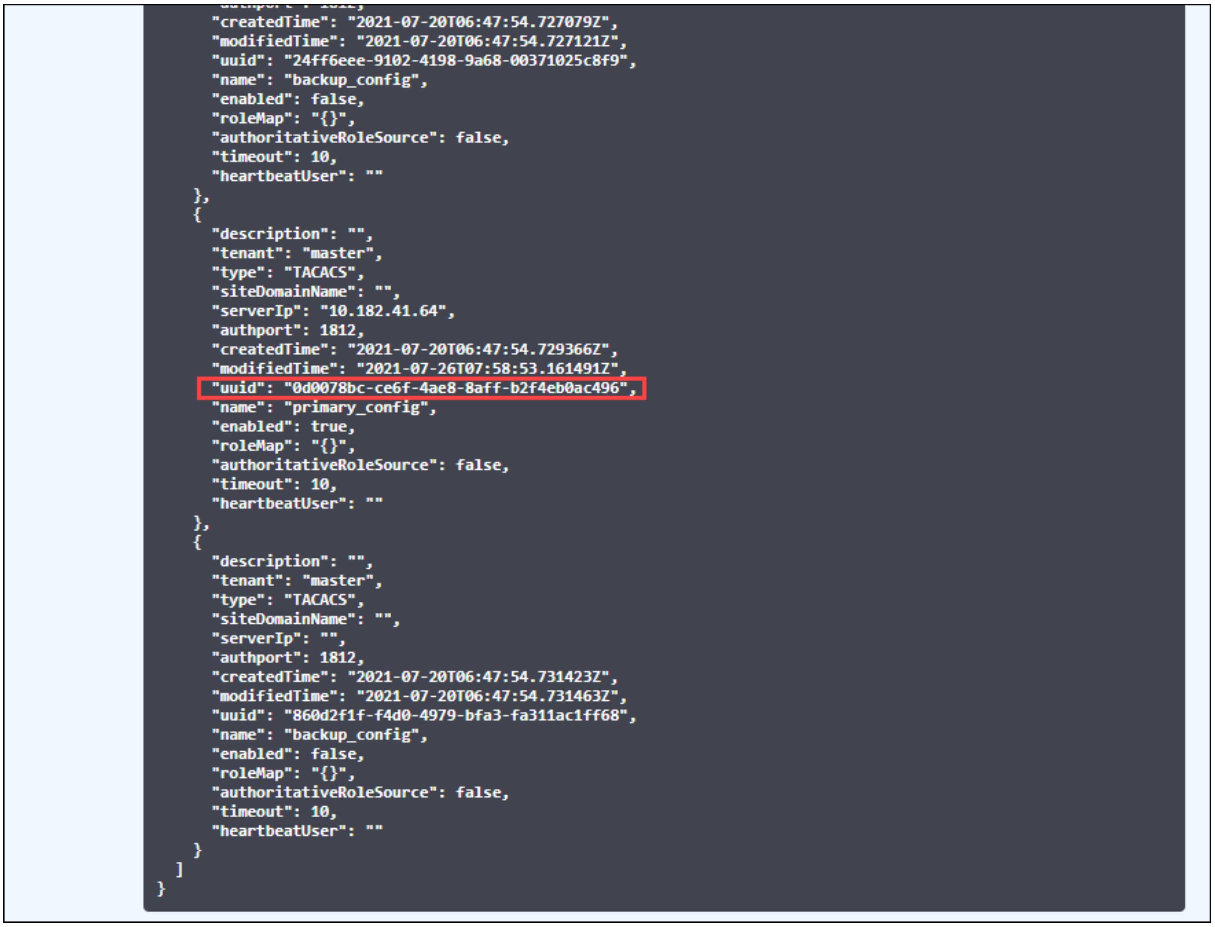
Example

```
 CURL  -s -k -H "Authorization: token $TOKEN" -X PATCH -d  '{
    "name": "primary_config",
    "description": "primary authconfig",
    "type": 'TACACS',
    "enabled": True,
    "server_ip": 'localhost',
    "server_secret": 'supersecret',
    "tenant": "master",
    "role_map": '{"default": {"rolenames": ["admin"]}}',
 }' https://localhost/tron/api/v1/auth-configs/1d1205fc-8a93-4088-b052-
ccb76860f97c/
```

Sample TACACS configuration

This section describes the procedure to configure TACACS:

1. Login to BP product as a user having **security admin** or **sysadmin** privileges.
2. From the App Bar UI, navigate to **System > Platform**.
3. Select **Swagger Ui**.
   The **Blue Planet APIs** page opens.
4. Click **UAC > auth-configs v1 > GET /api/v1/auth-configs List of all RADIUS configurations.**.
5. Click **Try it out**.
6. Click **Execute**.
   The configurations appear in the response.

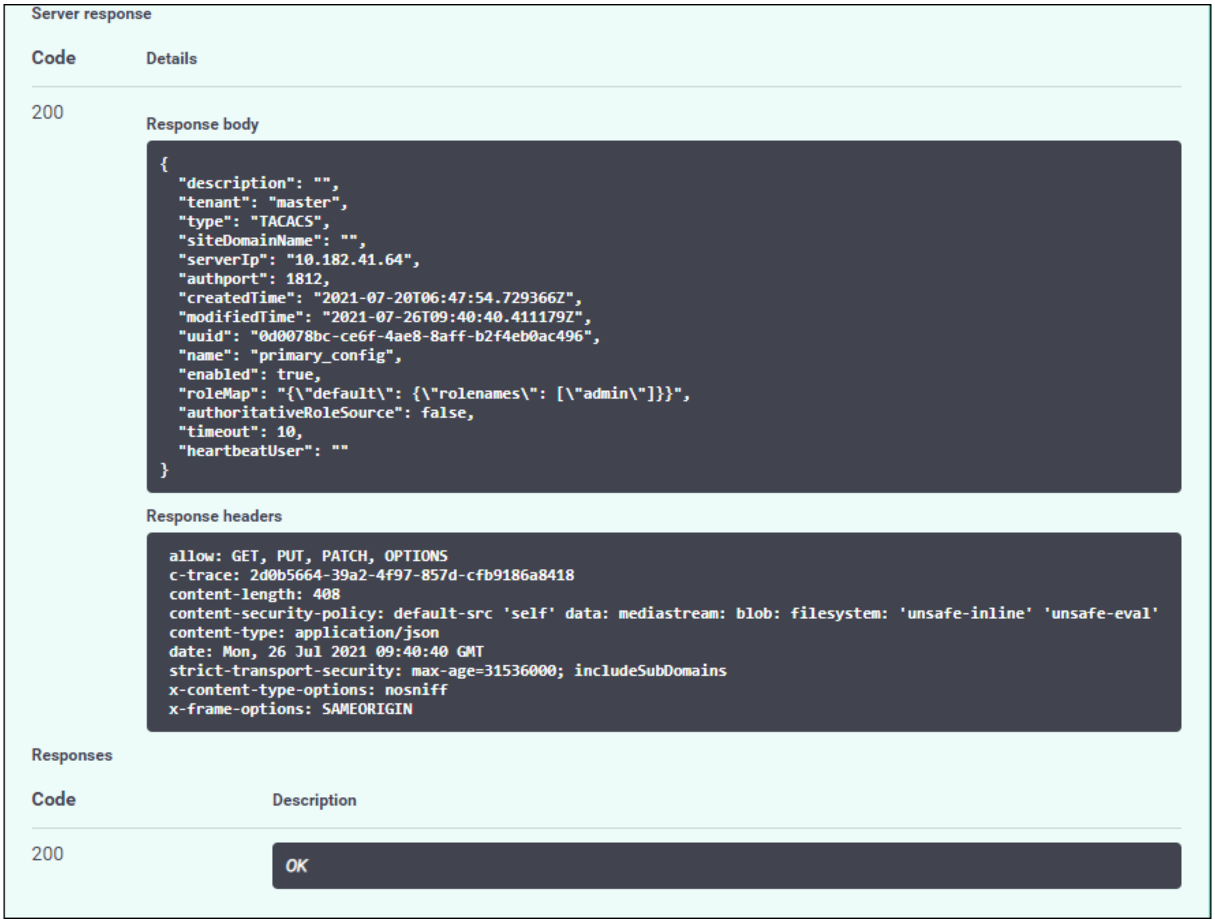


```
        "authport": 1812,
        "createdTime": "2021-07-20T06:47:54.727079Z",
        "modifiedTime": "2021-07-20T06:47:54.727121Z",
        "uuid": "24ff6eee-9102-4198-9a68-00371025c8f9",
        "name": "backup_config",
        "enabled": false,
        "roleMap": "{}",
        "authoritativeRoleSource": false,
        "timeout": 10,
        "heartbeatUser": ""
      },
      {
        "description": "",
        "tenant": "master",
        "type": "TACACS",
        "siteDomainName": "",
        "serverIp": "10.182.41.64",
        "authport": 1812,
        "createdTime": "2021-07-20T06:47:54.729366Z",
        "modifiedTime": "2021-07-26T07:58:53.161491Z",
        "uuid": "0d0078bc-ce6f-4ae8-8aff-b2f4eb0ac496",
        "name": "primary_config",
        "enabled": true,
        "roleMap": "{}",
        "authoritativeRoleSource": false,
        "timeout": 10,
        "heartbeatUser": ""
      },
      {
        "description": "",
        "tenant": "master",
        "type": "TACACS",
        "siteDomainName": "",
        "serverIp": "",
        "authport": 1812,
        "createdTime": "2021-07-20T06:47:54.731423Z",
        "modifiedTime": "2021-07-20T06:47:54.731463Z",
        "uuid": "860d2f1f-f4d0-4979-bfa3-fa311ac1ff68",
        "name": "backup_config",
        "enabled": false,
        "roleMap": "{}",
        "authoritativeRoleSource": false,
        "timeout": 10,
        "heartbeatUser": ""
      }
    ]
  }
```

7. Copy the TACACS uuid from the **Response Body**.
8. Click **PATCH /api/v1/auth-configs/{uuid} Update some values for the RADIUS config with this UUID.**.
9. Click **Try it out**.
10. In the **Parameters** section, in the **uuid** field, enter the unique id copied from step 7.
11. In the **description** field, enter the authenticator config description.
12. In the **tenant** field, type **master** for tenant.
13. In the **type** field, type **TACACS** for authenticator type.
14. In the **server_ip** field, enter the server ip of the configuration.
15. In the **authport** field, enter the auth port value.
16. In the **name** field, enter the configuration name.
17. In the **enabled** field, select **true** option to activate the configuration.
18. In the **role_map** field, enter the role map string that maps group or role to tron role.

    For example, **{"default": {"rolenames": ["admin"]}}**.
19. Click **Execute**.

    The configured TACACS configuration appears in the response.



Server response

Code Details

200

Response body

```
{
  "description": "",
  "tenant": "master",
  "type": "TACACS",
  "siteDomainName": "",
  "serverIp": "10.182.41.64",
  "authport": 1812,
  "createdTime": "2021-07-20T06:47:54.729366Z",
  "modifiedTime": "2021-07-26T09:40:40.411179Z",
  "uuid": "0d0078bc-ce6f-4ae8-8aff-b2f4eb0ac496",
  "name": "primary_config",
  "enabled": true,
  "roleMap": "{\"default\": {\"rolenames\": [\"admin\"]}}",
  "authoritativeRoleSource": false,
  "timeout": 10,
  "heartbeatUser": ""
}
```

Response headers

```
 allow: GET, PUT, PATCH, OPTIONS
 c-trace: 2d0b5664-39a2-4f97-857d-cfb9186a8418
 content-length: 408
 content-security-policy: default-src 'self' data: mediastream: blob: filesystem: 'unsafe-inline' 'unsafe-eval'
 content-type: application/json
 date: Mon, 26 Jul 2021 09:40:40 GMT
 strict-transport-security: max-age=31536000; includeSubDomains
 x-content-type-options: nosniff
 x-frame-options: SAMEORIGIN
```

Responses

Code	Description
200	*OK*

RADIUS, LDAP, and TACACS configuration

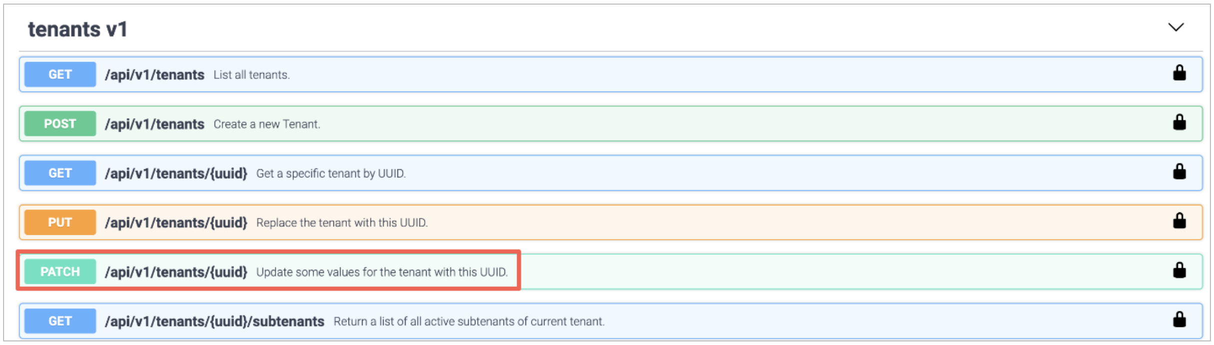
Enable the **fallback_to_local_auth** configuration at the tenant level in order to allow local (Blue Planet) authentication when external authentications (LDAP/RADIUS/TACACS) are configured.

NOTE Fallback to local authentication is applied to only the tenant for which fallback_to_local_auth is set to **true**. The same configuration is applied to sub-tenants. You must set fallback_to_local_auth as **true** to each sub-tenant.

To enable fallback to local (Blue Planet) authentication:

1. Login on to Blue Planet UI as **sysadmin** user
2. Navigate to **System > Platform > Swagger-UI**.
3. Select **UAC API's** and select the tenants v1 panel.

4. Click **PATCH** to run the Patch API:



tenants v1

GET /api/v1/tenants List all tenants.

POST /api/v1/tenants Create a new Tenant.

GET /api/v1/tenants/{uuid} Get a specific tenant by UUID.

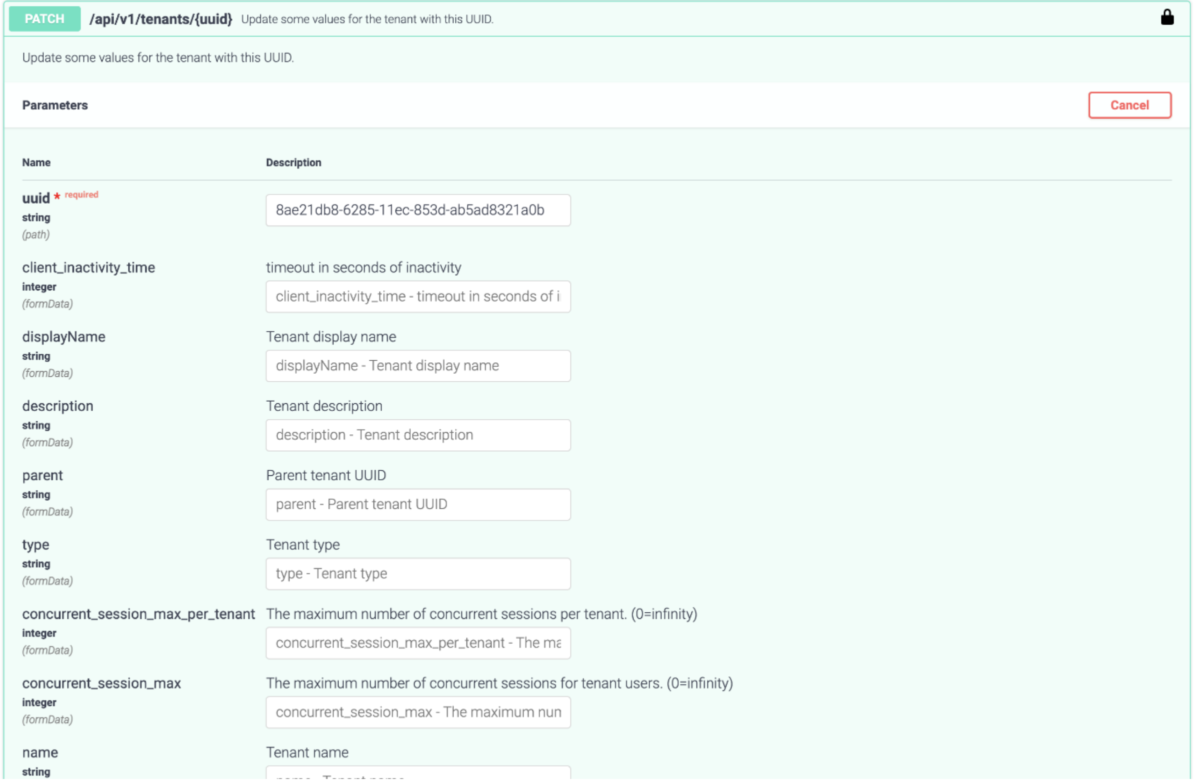
PUT /api/v1/tenants/{uuid} Replace the tenant with this UUID.

PATCH /api/v1/tenants/{uuid} Update some values for the tenant with this UUID.

GET /api/v1/tenants/{uuid}/subtenants Return a list of all active subtenants of current tenant.

5. Enter the uuid of tenant's fallback authentication that you want to enable or disable.

NOTE Run the GET API to see the uuid of the tenant.



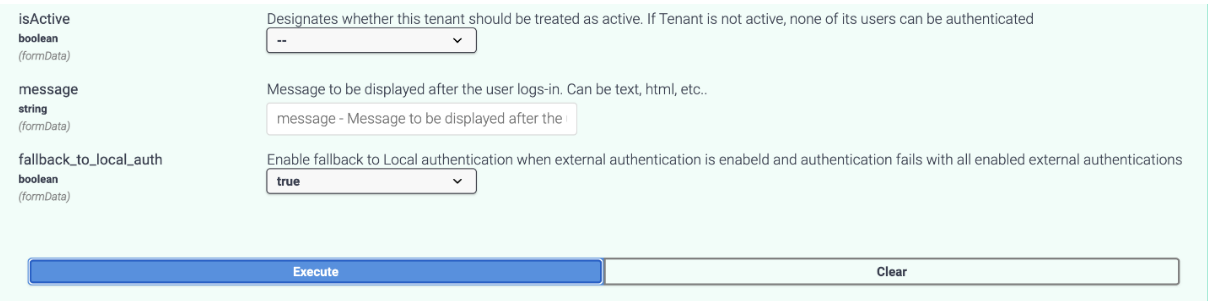
PATCH /api/v1/tenants/{uuid} Update some values for the tenant with this UUID.

Update some values for the tenant with this UUID.

Parameters

Name	Description
uuid * required string (path)	8ae21db8-6285-11ec-853d-ab5ad8321a0b
client_inactivity_time integer (formData)	timeout in seconds of inactivity client_inactivity_time - timeout in seconds of i
displayName string (formData)	Tenant display name displayName - Tenant display name
description string (formData)	Tenant description description - Tenant description
parent string (formData)	Parent tenant UUID parent - Parent tenant UUID
type string (formData)	Tenant type type - Tenant type
concurrent_session_max_per_tenant integer (formData)	The maximum number of concurrent sessions per tenant. (0=infinity) concurrent_session_max_per_tenant - The ma
concurrent_session_max integer (formData)	The maximum number of concurrent sessions for tenant users. (0=infinity) concurrent_session_max - The maximum nun
name string	Tenant name

6. For fallback_to_local_auth, select **true** or **false** to allow a local user to log on along with external users.



isActive
boolean
(formData)

Designates whether this tenant should be treated as active. If Tenant is not active, none of its users can be authenticated

--

message
string
(formData)

Message to be displayed after the user logs-in. Can be text, html, etc..

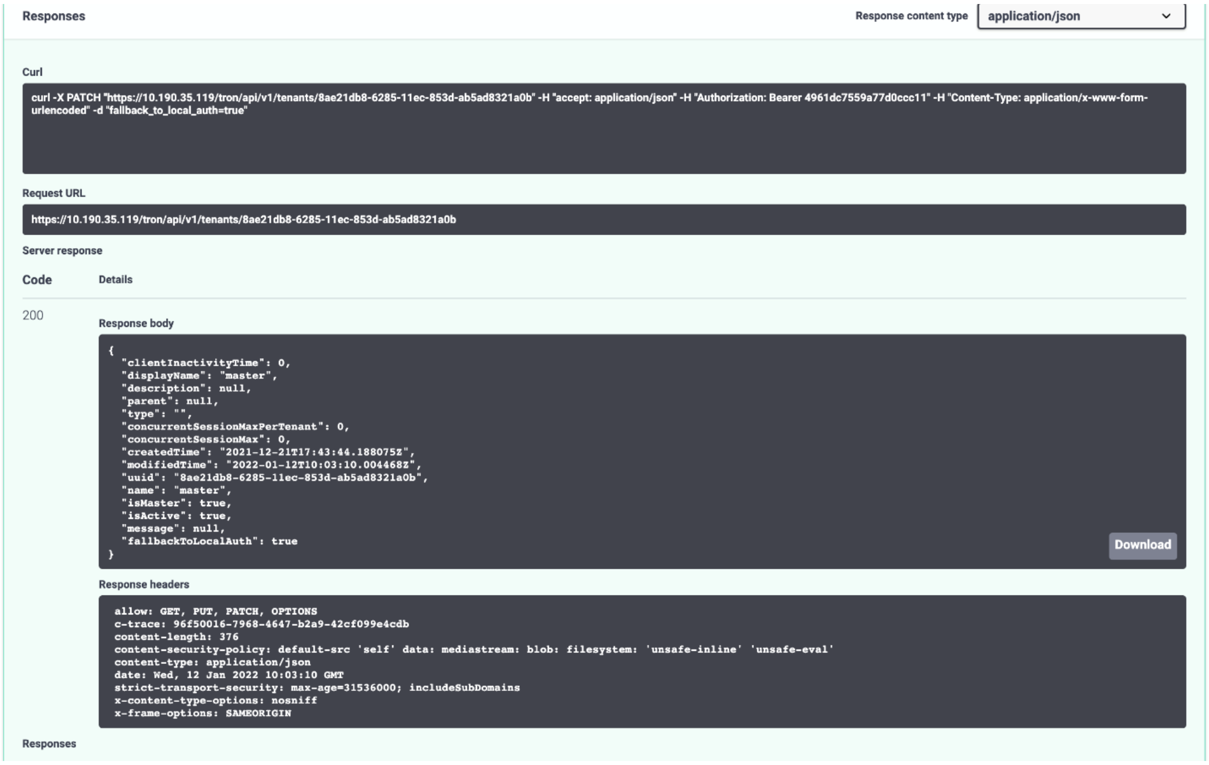
message - Message to be displayed after the

fallback_to_local_auth
boolean
(formData)

Enable fallback to Local authentication when external authentication is enabeld and authentication fails with all enabled external authentications

true

7. Click **Execute**. The following shows the API request:



Responses

Response content type: application/json

Curl

```
curl -X PATCH "https://10.190.35.119/tron/api/v1/tenants/8ae21db8-6285-11ec-853d-ab5ad8321a0b" -H "accept: application/json" -H "Authorization: Bearer 4961dc7559a77d0ccc11" -H "Content-Type: application/x-www-form-urlencoded" -d "fallback_to_local_auth=true"
```

Request URL

```
https://10.190.35.119/tron/api/v1/tenants/8ae21db8-6285-11ec-853d-ab5ad8321a0b
```

Server response

Code: 200

Response body

```
{
  "clientInactivityTime": 0,
  "displayName": "master",
  "description": null,
  "parent": null,
  "type": "",
  "concurrentSessionMaxPerTenant": 0,
  "concurrentSessionMax": 0,
  "createdTime": "2021-12-21T17:43:44.188075Z",
  "modifiedTime": "2022-01-12T10:03:10.004468Z",
  "uuid": "8ae21db8-6285-11ec-853d-ab5ad8321a0b",
  "name": "master",
  "isMaster": true,
  "isActive": true,
  "message": null,
  "fallbackToLocalAuth": true
}
```

Response headers

```
allow: GET, PUT, PATCH, OPTIONS
c-trace: 96f50016-7968-4647-b2a9-42cf099e4cdb
content-length: 376
content-security-policy: default-src 'self' data: mediastream: blob: filesystem: 'unsafe-inline' 'unsafe-eval'
content-type: application/json
date: Wed, 12 Jan 2022 10:03:10 GMT
strict-transport-security: max-age=31536000; includeSubDomains
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
```

Responses

```
curl -X PATCH "https://<site-ip>/tron/api/v1/tenants/<tenant_id>" -H "accept:
application/json" -H
"Authorization: Bearer <UAC token>" -H "Content-Type: application/x-www-form-
urlencoded" -d
"fallback_to_local_auth=true"
```

The Following shows the API response:

```
{
"clientInactivityTime": 0,
"displayName": "master",
"description": null,
"parent": null,
"type": "",
"concurrentSessionMaxPerTenant": 0,
"concurrentSessionMax": 0,
"createdTime": "2021-12-21T17:43:44.188075Z",
"modifiedTime": "2022-01-12T10:03:10.004468Z",
"uuid": "8ae21db8-6285-11ec-853d-ab5ad8321a0b",
"name": "master",
"isMaster": true,
"isActive": true,
"message": null,
"fallbackToLocalAuth": true
}
```

SAML configuration

NOTE This feature is not supported for UAA in the 23.08 release.

To accept a Security Assertion Markup Language (SAML) servers authorization, Tron requires an additional support. A newly configured API allows the administrator to control the interaction between Tron and the SAML server.

NOTE Tron does not authorize through SAML. It creates tokens for SAML response artifacts.

Terminology

The following table lists the terminologies for SAML configuration and their descriptions:

TERM	DESCRIPTION
SAML	It is an open standard for exchanging authentication and authorization data between parties.
idp	It is a system entity for SAML identity provider that issues authentication assertions in conjunction with a single sign-on (SSO) server.
sp	It is a system entity for SAML service provider that receives and accepts authentication assertions from a single sign-on (SSO) server.

TERM	DESCRIPTION
artifact	A SAML artifact or HTTP artifact is a binding in which a SAML request or response (or both) is transmitted by reference by using a unique identifier. In case of Tron, you should care only about the response artifact.

Example

- A dollar sign($) and a word indicates a variable.
  - $PREFIX - `PREFIX=https://localhost/tron`
  - $TOKEN = `TOKEN=$(curl --silent -k -H "Content-Type:application/json" -d '{"username":"admin","password":"adminpw"}' $PREFIX/api/v1/tokens | python -c "import sys, json; print json.load(sys.stdin)['token']")`
  - Greater Than or Less Than(<>) indicate that you need to supply the info indicated within the <>.
  - `| python -m json.tool` is added to the end of some calls to format the return data. It is not needed.

Requirements

You need to set up the SAML configuration to be enabled in Tron and update the IP address of your SAML server in Tron. Use the admin credentials in the SAML configuration API.

> **NOTE** A maximum of two SAML servers can be configured and must be named `primary_config` and `backup_config`. Otherwise, authentication breaks. Both the SAML servers are created by default.

Sample SAML Artifact

The SAML Artifact is required to be passed to the Tron API in `base64` encoding. (which is the normal encoding returned from the `idp`)

```
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48c2FtbDJwOlJlc3BvbnNlIHhtbG5zOnN
hbWwycD0idX
mJuO9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOnByb3RvY29sIiB4bWxuczp4cz0iaHR0cDovL3d3dy53My5vc
mcvMjAwMS9YTU
xTY2hlbWEiIENvbnNlbnQ9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDpjb25zZW50OnVuc3BlY2lma
WVkIiBEZXN0aW
5hdGlvbj0iaHR0cHM6Ly8xMC4yMDYuMzAuMTExL3Ryb24vYXBpL3YxL3Rva2Vucy1zYW1sLyIgSUQ9IlpPO
U9aQTJUNUg4Mj
BaNFlIVjJUQlJTOExPWEZITzVVWFlKVFEwWEsiIElzc3VlSW5zdGFudD0iMjAxOS0wNi0yNVQxNTo1Mzo0N
i42MTlaIiBWZX
JzaW9uPSIyLjAiPjxzYW1sMjpJc3N1ZXIgeG1sbnM6c2FtbDI9InVybjpvYXNpczpuYW1lczp0YzpTQU1MO
jIuMDphc3Nlcn
```

```
Rpb24iIEZvcm1hdD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOm5hbWVpZC1mb3JtYXQ6ZW50aXR5I
j5odHRwcy8vOj
EwLjIwNi4zMC4xMTE8L3NhbWwyOklzc3Vlcj48ZHM6U2lnbmF0dXJlIHhtbG5zOmRzPSJodHRwOi8vd3d3L
nczLm9yZy8yMD
AwLzA5L3htbGRzaWcjIj48ZHM6U2lnbmVkSW5mbz48ZHM6Q2Fub25pY2FsaXphdGlvbk1ldGhvZCBBbGdvc
ml0aG09Imh0dH
A6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjxkczpTaWduYXR1cmVNZXRob2QgQWxnb
3JpdGhtPSJodH
RwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNyc2Etc2hhMjU2Ii8+PGRzOlJlZmVyZW5jZ
SBVUkk9IiNaTz
lPWkEyVDVIODIwWjRZSFYyVEJSUzhMT1hGSE81VVhZSlRRMFhLIj48ZHM6VHJhbnNmb3Jtcz48ZHM6VHJhb
nNmb3JtIEFsZ2
9yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI2VudmVsb3BlZC1zaWduYXR1cmUiL
z48ZHM6VHJhbn
Nmb3JtIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIj48ZWM6S
W5jbHVzaXZlTm
FtZXNwYWNlcyB4bWxuczplYz0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIiBQc
mVmaXhMaXN0PS
J4cyIvPjwvZHM6VHJhbnNmb3JtPjwvZHM6VHJhbnNmb3Jtcz48ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRob
T0iaHR0cDovL3
d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+PGRzOkRpZ2VzdFZhbHVlPk1kc2ovZ2FxdmZhU
Gs2QWNrMnVyWG
9UeURRaEVJMXk4d1RLZnFweERXV0k9PC9kczpEaWdlc3RWYWx1ZT48L2RzOlJlZmVyZW5jZT48L2RzOlNpZ
25lZEluZm8+PG
RzOlNpZ25hdHVyZVZhbHVlPlE2ZTdkWS84MGtGOXlWK2RlN1QwdXVBVHlWcUdyWmRFTFZqd2ZGQk53dWZ6U
Fh3TGpPaDlwTE
NmU3ZpY1hvdFh5clNyUER5dXVreGNoeVNwOW5NblNRNE9EN3dxTlNROW1FdXl6ZS9iUVRNQkgrajFWb1dWZ
nE2S28xcVhPc1
Y4UnRLN2dPa2lCeTlIUG9NbVN1aGZxWjNEdXJLQ0RxeFJ6YjdCRmZBc1FBWUQySXRCeG5lY2ZBZmdnbVJaS
U1kMDVEaEhjY2
V5OVIrNyt5UGhsS1F0T1JzODNxNDdMUitWaFZ3dWtyV2diWG1HeEhEK2ZsVEYybkNLMkRRL3RKdlNMY1ZXR
1NnUURLZXBoeT
BXUnpEbVpYSE00a3ZWVlBrOS9LUnozVjAzVUx3U3JUSG11dTd4Vi9NK2JXcjF2NTY1QXdRbjZMR3h5Q1R1V
VdodVYyT3NOUT
09PC9kczpTaWduYXR1cmVWYWx1ZT48ZHM6S2V5SW5mbz48ZHM6WDUwOURhdGE+PGRzOlg1MDlDZXJ0aWZpY
2F0ZT5NSUlEam
pDQ0FuWUNDUURCcXN3M3JhcWZFekFOQmdrcWhraUc5dzBCQVFzRkFEQ0JpREVMTUFrR0ExVUVCaE1DVlZNe
EVUQVBCZ05WQk
FnTUNFMWhjbmxzWVc1a01SRXdEd1lEVlFRSERBaERiMngxYldKcFlURVRNQkVHQTFVRUNnd0tRbXgxWlZCc
1lXNWxkREVSTU
E4R0ExVUVDd3dJVUd4aGRHWnZjbTB4Q3pBSkJnTlZCQU1NQWtKUU1SNHdIQVlKS29aSWh2Y05BUWtCRmc5e
GVtaGhiMEJqYV
dWdVlTNWpiMjB3SGhjTk1Ua3dOakU1TVRjd01qSTNXaGNOTWpJd05qRTRNVGN3TWpJM1dqQ0JpREVMTUFrR
0ExVUVCaE1DVl
ZNeEVUQVBCZ05WQkFnTUNFMWhjbmxzWVc1a01SRXdEd1lEVlFRSERBaERiMngxYldKcFlURVRNQkVHQTFVR
UNnd0tRbXgxWl
ZCc1lXNWxkREVSTUE4R0ExVUVDd3dJVUd4aGRHWnZjbTB4Q3pBSkJnTlZCQU1NQWtKUU1SNHdIQVlKS29aS
Wh2Y05BUWtCRm
c5eGVtaGhiMEJqYVdWdVlTNWpiMjB3Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLQW9JQ
kFRQ3RoeVJ5NC
tJaDcvZjRHYWxQTkVrYzg1a3A0bW1MQjhZRjkrQXZiQkpweGtYeGtrTU5PR3NESEszTDJERmdpVTJ5Sm94U
GJkWkFjVDEwcF
owbHdyNStiZG1vd041b2dyMFZoc0dCcUk4eHcxWTZNdVdBQXZwb2RoU0ZaWml4bmhEWHRBeFZjWGprcWJpV
DVmVWthNVkzT1
Jhdm95UW1Ta0FjS1lheW5wSE1HQlR6eE9XenE5eUQ3azNvcVUrbUZnMUdyTkJPSUcwd253ZGtGUjl2cTRjN
m55QVRJemVnb2
U3NjI5VzJ6dTgwaEhaTG5TY1dNVCtHVGpCK25YN1NTVWxBdGxjeEovakFVb3MzTmFZWTZCV1dVdUJmaWdIM
nhidnZJbFFaS2
VUTzk2Z0xwUkJ5ek9kQVArTFZiNTF6d2ZBMmVVbWRRVWFyQzVWcGJBb1RQclIzQWdNQkFBRXdEUVlKS29aS
```

```
Wh2Y05BUUVMQl
FBRGdnRUJBSlNFS0Fmc1c4UldSNFJqTTd5SmRyczB6MFdRSFo2bTNCUm5RN0s3UnJZbHpQbTFpcC9JT2dwe
HZhOGozOGk1aF
M3T2JnSnFBS0hMUmNMb2w2VDZmZnJ2UUp2bE1tUk5DdkpKQUVvc214b1MrQm1SbzNpYldRQjVrTC84ajYrM
jY0dm1jczdrdW
ticGF4WFZ5K0VGMG1ORklvS2xRdmUxcmpzVE1YN3YxRVhING5OWWQ5eGZKTGhrTFZudUpZdFFXWGp2UXhOS
llON2FHT1c2Tm
l2UVZieHR2cnhQeFZVR2p1QmpUSi9wakNlU1F0djZBLzdRYXNESkErakF6ZWRDaFpEZmxRMU9GQnlsU0dFN
2IrSjU1Y0MrZl
FYTTlyOWdrR0cxNHNYU29kY3FKSEtVWnBod05DS2VZSDFjdHU4cGxvV2cvZlJGdlZJUDliU0pnNHltS3BZP
TwvZHM6WDUwOU
NlcnRpZmljYXRlPjwvZHM6WDUwOURhdGE+PC9kczpLZXlJbmZvPjwvZHM6U2lnbmF0dXJlPjxzYW1sMnA6U
3RhdHVzPjxzYW
1sMnA6U3RhdHVzQ29kZSBWYWx1ZT0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOnN0YXR1czpTdWNjZ
XNzIi8+PC9zYW
1sMnA6U3RhdHVzPjxzYW1sMjpBc3NlcnRpb24geG1sbnM6c2FtbDI9InVybjpvYXNpczpuYW1lczp0YzpTQ
U1MOjIuMDphc3
NlcnRpb24iIElEPSJTRUtUUlNLUlFVUEtYS0ZJQVNMTFExVlpMMUw3V1hYNUlIR1Q4RlFQIiBJc3N1ZUluc
3RhbnQ9IjIwMT
ktMDYtMjVUMTU6NTM6NDYuNjE5WiIgVmVyc2lvbj0iMi4wIj48c2FtbDI6SXNzdWVyIEZvcm1hdD0idXJuO
m9hc2lzOm5hbW
VzOnRjOlNBTUw6Mi4wOm5hbWVpZC1mb3JtYXQ6ZW50aXR5Ij5odHRwcy8vOjEwLjIwNi4zMC4xMTE8L3Nhb
WwyOklzc3Vlcj
48c2FtbDI6U3ViamVjdD48c2FtbDI6TmFtZUlEIEZvcm1hdD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6M
S4wOm5hbWVpZC
1mb3JtYXQ6dW5zcGVjaWZpZWQiPnRlc3Rlcjwvc2FtbDI6TmFtZUlEPjxzYW1sMjpTdWJqZWN0Q29uZmlyb
WF0aW9uIE1ldG
hvZD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmNtOmJlYXJlciI+PHNhbWwyOlN1YmplY3RDb25ma
XJtYXRpb25EYX
RhIE5vdE9uT3JBZnRlcj0iMjAxOS0wNi0yNVQxNTo1ODo0Ni42MTlaIiBSZWNpcGllbnQ9Imh0dHBzOi8vM
TAuMjA2LjMwLj
ExMS90cm9uL2FwaS92MS90b2tlbnMtc2FtbC8iLz48L3NhbWwyOlN1YmplY3RDb25maXJtYXRpb24+PC9zY
W1sMjpTdWJqZW
N0PjxzYW1sMjpDb25kaXRpb25zIE5vdEJlZm9yZT0iMjAxOS0wNi0yNVQxNTo0ODo0Ni42MTlaIiBOb3RPb
k9yQWZ0ZXI9Ij
IwMTktMDYtMjVUMTU6NTg6NDYuNjE5WiI+PHNhbWwyOkF1ZGllbmNlUmVzdHJpY3Rpb24+PHNhbWwyOkF1Z
GllbmNlPnRlc3
QtdHJvbi1zcDwvc2FtbDI6QXVkaWVuY2U+PC9zYW1sMjpBdWRpZW5jZVJlc3RyaWN0aW9uPjwvc2FtbDI6Q
29uZGl0aW9ucz
48c2FtbDI6QXR0cmlidXRlU3RhdGVtZW50PjxzYW1sMjpBdHRyaWJ1dGUgTmFtZT0iRW1haWwiIE5hbWVGb
3JtYXQ9InVybj
pvYXNpczpuYW1lczp0YzpTQU1MOjIuMDphdHRybmFtZS1mb3JtYXQ6dW5zcGVjaWZpZWQiPjxzYW1sMjpBd
HRyaWJ1dGVWYW
x1ZSB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3RhbmNlIiB4c2k6d
HlwZT0ieHM6c3
RyaW5nIj5kYnJvdW5zdEBibHVlcGxhbmV0LmNvbTwvc2FtbDI6QXR0cmlidXRlVmFsdWU+PC9zYW1sMjpBd
HRyaWJ1dGU+PH
NhbWwyOkF0dHJpYnV0ZSBOYW1lPSJGaXJzdE5hbWUiIE5hbWVGb3JtYXQ9InVybjpvYXNpczpuYW1lczp0Y
zpTQU1MOjIuMD
phdHRybmFtZS1mb3JtYXQ6dW5zcGVjaWZpZWQiPjxzYW1sMjpBdHRyaWJ1dGVWYWx1ZSB4bWxuczp4c2k9I
mh0dHA6Ly93d3
cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3RhbmNlIiB4c2k6dHlwZT0ieHM6c3RyaW5nIj5UZXN0ZXI8L
3NhbWwyOkF0dH
JpYnV0ZVZhbHVlPjwvc2FtbDI6QXR0cmlidXRlPjxzYW1sMjpBdHRyaWJ1dGUgTmFtZT0ibWVtYmVyT2YiI
E5hbWVGb3JtYX
Q9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDphdHRybmFtZS1mb3JtYXQ6dW5zcGVjaWZpZWQiPjxzY
W1sMjpBdHRyaW
J1dGVWYWx1ZSB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3RhbmNlI
iB4c2k6dHlwZT
```

```
0ieHM6c3RyaW5nIj5CUF9VU0VSPC9zYW1sMjpBdHRyaWJ1dGVWYWx1ZT48L3NhbWwyOkF0dHJpYnV0ZT48c
2FtbDI6QXR0cm
lidXRlIE5hbWU9Ikxhc3ROYW1lIiBOYW1lRm9ybWF0PSJ1cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6Y
XR0cm5hbWUtZm
9ybWF0OnVuc3BlY2lmaWVkIj48c2FtbDI6QXR0cmlidXRlVmFsdWUgeG1sbnM6eHNpPSJodHRwOi8vd3d3L
nczLm9yZy8yMD
AxL1hNTFNjaGVtYS1pbnN0YW5jZSIgeHNpOnR5cGU9InhzOnN0cmluZyI+Qmx1ZVBsYW5ldDwvc2FtbDI6Q
XR0cmlidXRlVm
FsdWU+PC9zYW1sMjpBdHRyaWJ1dGU+PC9zYW1sMjpBdHRyaWJ1dGVTdGF0ZW1lbnQ+PHNhbWwyOkF1dGhuU
3RhdGVtZW50IE
F1dGhuSW5zdGFudD0iMjAxOS0wNi0yNVQxNTo1Mzo0Ni42MTlaIj48c2FtbDI6QXV0aG5Db250ZXh0PjxzY
W1sMjpBdXRobk
NvbnRleHRDbGFzc1JlZj51cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6YWM6Y2xhc3NlczpQYXNzd29yZ
FByb3RlY3RlZF
RyYW5zcG9ydDwvc2FtbDI6QXV0aG5Db250ZXh0Q2xhc3NSZWY+PC9zYW1sMjpBdXRobkNvbnRleHQ+PC9zY
W1sMjpBdXRobl
N0YXRlbWVudD48L3NhbWwyOkFzc2VydGlvbj48L3NhbWwycDpSZXNwb25zZT4
```

The decoded version:

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2p:Response xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
Consent="urn:oasis:names:tc:SAML:2.0:consent:unspecified"
Destination="https://10.206.30.111/tron/api/v1/tokens-saml/"
ID="ZO9OZA2T5H820Z4YHV2TBRS8LOXFHO5UXYJTQ0XK" IssueInstant="2019-06-
25T15:53:46.619Z" Version="2.0">
   <saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https//:10.206.30.111</saml2:Issuer>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
         <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
         <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256" />
         <ds:Reference URI="#ZO9OZA2T5H820Z4YHV2TBRS8LOXFHO5UXYJTQ0XK">
            <ds:Transforms>
               <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
               <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                  <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-
exc-c14n#" PrefixList="xs" />
               </ds:Transform>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />

<ds:DigestValue>Mdsj/gaqvfaPk6Ack2urXoTyDQhEI1y8wTKfqpxDWWI=</ds:DigestValue>
         </ds:Reference>
      </ds:SignedInfo>

<ds:SignatureValue>y0WRzDmZXHM4kvVVPk9/KRz3V03ULwSrTHmuu7xV/M+bWr1v565AwQn6LGxyCTuU
WhuV2OsNQ==</ds:SignatureValue>
      <ds:KeyInfo>
         <ds:X509Data>

<ds:X509Certificate>MIIDjjCCAwggEKAoIBAQCthyRy4+Ih7/f4Gal1ctu8ploWg/fRFvVIP9bSJg4ym
```

```
KpY=</ds:X509Certificate>
         </ds:X509Data>
      </ds:KeyInfo>
   </ds:Signature>
   <saml2p:Status>
      <saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
   </saml2p:Status>
   <saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
ID="SEKTRSKRQUPKXKFIASLLQ1VZL1L7WXX5IHGT8FQP" IssueInstant="2019-06-
25T15:53:46.619Z" Version="2.0">
      <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https//:10.206.30.111</saml2:Issuer>
      <saml2:Subject>
         <saml2:NameID Format="urn:oasis:names:tc:SAML:1.0:nameid-
format:unspecified">tester</saml2:NameID>
         <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
            <saml2:SubjectConfirmationData NotOnOrAfter="2019-06-25T15:58:46.619Z"
Recipient="https://10.206.30.111/tron/api/v1/tokens-saml/" />
         </saml2:SubjectConfirmation>
      </saml2:Subject>
      <saml2:Conditions NotBefore="2019-06-25T15:48:46.619Z" NotOnOrAfter="2019-06-
25T15:58:46.619Z">
         <saml2:AudienceRestriction>
            <saml2:Audience>test-tron-sp</saml2:Audience>
         </saml2:AudienceRestriction>
      </saml2:Conditions>
      <saml2:AttributeStatement>
         <saml2:Attribute Name="Email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
            <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xs:string">dbrounst@blueplanet.com</saml2:AttributeValue>
         </saml2:Attribute>
         <saml2:Attribute Name="FirstName"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
            <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xs:string">Tester</saml2:AttributeValue>
         </saml2:Attribute>
         <saml2:Attribute Name="memberOf"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
            <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xs:string">BP_USER</saml2:AttributeValue>
         </saml2:Attribute>
         <saml2:Attribute Name="LastName"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
            <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xs:string">BluePlanet</saml2:AttributeValue>
         </saml2:Attribute>
      </saml2:AttributeStatement>
      <saml2:AuthnStatement AuthnInstant="2019-06-25T15:53:46.619Z">
         <saml2:AuthnContext>

<saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtecte
dTransport</saml2:AuthnContextClassRef>
         </saml2:AuthnContext>
      </saml2:AuthnStatement>
   </saml2:Assertion>
</saml2p:Response>
```

SAML Configuration Fields

The following table lists configuration fields for SAML and their descriptions:

NAME	DESCRIPTION OR NOTES
uuid	Unique identifier (Read Only).
name	Name of config `primary_config` and `backup_config` only. Unique per tenant description.
enabled	`True` to enable. default=`False`.
tenant	Tenant to filter users.
role_map	Optional dictionary string that maps group/role to tron role.
sso_url	The URL of the SAML server. default=`https://blueplaneturl.com/sp/ACS.saml2`
entity_id	SP entity id. default=`bluePlanetEntityId`
idp_issuer	idp entity id. default=`SAMLserverEntityID`
idp_issuer_uri	The URI of the issuer identity provider. default=`https://login-dev.ciena.com`
cert_file	Name and path of the certificate file. default=`/etc/bp2/tron/sso_cert.pem`
logout_url	The url where UI will redirect after logout.
show_logout_button	If enabled, logout button is displayed on UI.
recipient	ACS URL at SAML server.

Sample Tron SAML Configuration

Simple Configuration

The purpose of this endpoint is to provide minimal SAML configuration parameters to external parties so they can initiate SAML authentication (therefore it does not require authentication/authorization)

```
curl -k $PREFIX/api/v1/saml-configs-simple/
```

Full Configuration

```
curl -k -H "Authorization: token $TOKEN" $PREFIX/api/v1/saml-configs
```

Update or Change the SAML Configuration

```
CONFIG_UUID=$(curl \
   --silent -k \
   -H "Authorization: token $TOKEN" \
   $PREFIX/api/v1/saml-configs | python -c "import sys, json; print
json.load(sys.stdin)['results'][0]['uuid']")

curl -k -X PATCH
   -H "Content-Type:application/json"
   -H "Authorization: token $TOKEN"
   -d '{"enabled":True,"sso_url": "<idp url>","cert_file":"<Cert file path &
name>"}`
   $PREFIX/api/v1/saml-configs/$CONFIG_UUID
```

Sample Tron SAML Configuration Update Response

```
{
  "description": "primary SAML config",
  "tenant": "master",
  "createdTime": "2019-08-07T14:57:22.173581Z",
  "modifiedTime": "2019-08-07T14:57:22.173645Z",
  "uuid": "bd243a7d-4cf1-44f1-b06b-3b83fe0b09de",
  "name": "primary_config",
  "enabled": true,
  "roleMap": "{}",
  "useStrict": true,
  "useDebug": false,
  "ssoUrl": "https://sso.jumpcloud.com/saml2/tron",
  "entityId": "bluePlanetEntityId",
  "idpIssuerUri": "https://login-dev.ciena.com",
  "certFile": "/etc/bp2/tron/sso_cert.pem"
  "logoutUrl": "https://login-dev.ciena.com/logout",
  "showLogoutButton": false
  "recipient": "https://bpserverdns.com/tron/api/v1/tokens-saml"
}
```

Sample Tron SAML Token Request

```
curl -k -X POST
   -H "Content-Type:application/json"
   -H "Authorization: token $TOKEN"
   -d '{"tenant":"<tenant name>","SAMLResponse":"<saml base64 artifact>"}'
   $PREFIX/api/v1/tokens-saml
```

Sample Tron SAML Token Response

```
{
  "token": "c4413cf84ee1ca77cb2f",
  "login_detail": {
    "time": "2019-08-07T14:33:47.268531Z",
    "ip_address": "127.0.0.1",
    "user_agent": null,
    "session_id": "f6eed20f-096a-4926-bcab-66ecca43e9bd",
    "session_type": null
  },
  "createdTime": "2019-08-07T14:33:47.268531Z",
  "inactive_expiration_time": null,
  "is_successful": true,
  "timeout": 86400,
  "username": "tester",
  "user_tenant_uuid": "923a3a04-ba9b-4b41-bbe1-efd139abdd8d",
  "failedLoginAttempts": 0,
  "lastSuccessIpAddress": "None",
  "lastSuccessLogin": "None"
}
```

SAML Setup

NOTE If the SAML provider supports 2-factor authentication, then it gets support on the Tron.

To setup a SAML server for Tron:

1. Before starting SAML setup make sure that all Blue Planet roles (default/custom) are setup in the Roles UAC API. Note the names of each role you setup for use in step 12c.

   Example

   ```
   Admin, Application Admin
   ```

2. Go to the Blueplanet UI login page at:

   ```
   https://<IP Address>/login/?next=bp_local_login
   ```

   This is the administrator login page to bypass SSO login.

3. Login and go to the swagger API page:

   ```
   https://<IP Address>/swagger-ui/#/?swaggerUrl=%2Ftron%2Fdocs%2Fapi-docs%2F
   ```

4. Drop down the saml-configs.
5. Select **GET List of all SAML configurations**.
6. Select **Try it out** and then **Execute**.
7. Note the UUIDs for both `primary_config` and `backup_config`. For the remainder of this document we will call them <`UUID1`> and <`UUID2`> respectively.
8. Close the **GET List of all SAML configurations** section.
9. Select **PATCH Update some values for the SAML config with this UUID**.
10. Click **Try it out**.
11. Enter the following fields:
    a. uuid: Enter <`UUID1`> in the uuid field.
    b. enabled: Select `true` in the enabled field.
    c. role_map:

    The following table lists the role_map value and their descriptions:

ROLE_MAP VALUE	DESCRIPTION OR NOTES
empty	No role is assigned to SAML user, irrespective to any role that comes from SAML server.
{"default": {"rolenames":[<blueplanet role1>,<blueplanet role2>,<etc>]}}	Listed default roles are assigned to SAML user, irrespective to any role that comes from SAML server.
{ <Saml group name>: [{ "uac_role_name": "admin", "app_name": "UAC" }, { "uac_role_name": "Application admin", "app_name": "BluePlanet" }], "default": {"rolenames":["sysadmin"]} }	If SAML server provides user group name (<SAML group name>) as "admin", then the entry in the role-map value field should indicate the mapping between Saml group name and BluePlanet Roles else it recede to default roles.
{ <Saml group name>: [{ "uac_role_name": "admin", "app_name": "UAC" }, { "uac_role_name": "Application admin", "app_name": "BluePlanet" }] }	If SAML server provides user group name (<Saml group name>) as "admin" then the entry in the role-map value field should indicate the mapping between Saml group name and BluePlanet roles else no role is assigned.

Replace <rolename> placeholders with the BP rolenames for each user by default. In the example there are three roles, However you can enter any number of roles as per requirement, separated by commas.

> **NOTE** For no default roles enter `{}` or leave the field blank.

d. sso_url: (optional)

e. entity_id: Enter sp entity id.

f. idp_issuer_uri: Enter idp URL.

g. cert_file: The path and name of the certificate file. The public certificate needs to be entered into an accessible file on the BluePlanet server/Tron container. I suggest `/etc/bp2/tron/sso_cert.pem` which is the default.

h. logout_url (optional): The URL where UI will redirect after logout.

i. show_logout_button (optional): If enabled, logout button will be shown on UI.

j. idp_issuer: Enter idp entity id.

k. recipient: Enter `https://`**`<Blueplanet Server address>`**`/tron/api/v1/tokens-saml`.

l. user_strict: Select `true`.

12. Click **Execute**.

13. Repeat from Step 7, if you have SAML SSO server installed for "backup_config".

Host Operating System security

This section covers:

- Host OS Patching
- Host OS CIS hardening

Host OS Patching

The host OS should be kept updated for stability and security reasons. Before installing a BP product, ensure that the Ciena System Bundle is applied first before patching the system. After any BP product is installed, ensure that the product has been halted, refer to *Shutting down a Blue Planet cluster* in respective *Administrator Guide*.

For BluePlanet product installation with Geographical Redundancy, BluePlanet recommends applying OS patches to the backup site first and verifying whether the GR synchronization status is healthy before patching the primary site.

Host OS CIS hardening

Operating system (OS) hardening is a set of procedures or configuration settings applied to the system. This allows the system to comply with specific security policies restricting access to resources in the environment. The Center for Internet Security (CIS) provides benchmark documentation that lists the best practices to secure the system. Blue Planet uses these benchmarks to secure systems that are used for product testing and to ensure they will operate properly in a hardened environment.

Blue Planet products are tested against all CIS recommendations (Level 1 and 2, Scored and Unscored) from the following benchmarks:

- CIS Red Hat Enterprise Linux 7 Benchmark (v. 2.2.0)
- CIS CentOS Linux 7 Benchmark (v. 2.2.0)
- CIS Oracle Linux 7 Benchmark (v. 2.1.0)

Use these recommendations to apply any hardening on the host OS of any system running BP products. Hardening outside of these recommendations will be considered unsupported.

The following sections will detail the steps to follow for a hardened system as well as exceptions to the recommendations in the benchmarks:

- Hardening the OS
- Special considerations
- Enabling the CIS firewall profile

Hardening the OS

Below is the steps for installing BP product in a hardened environment is the following:

1. Install the OS.
2. Install the System Bundle.
3. Harden the OS.
4. Reboot the system.

NOTE Although the system requires a reboot after the Bundle installation and hardening step, the system reboot can also be done after completing both the steps together.

5. Install the BP product

NOTE Do not apply hardening after the product has been installed without consulting BP Support. The System Bundle and the BP Installer make modifications to the hardening to all the products for proper functioning

NOTE Use of a pre-hardened image (eg. AWS CIS AMI) is supported.

Special Considerations

Some CIS recommendations require special consideration when applied to Blue Planet products.

This section covers:

- Ensure IP Forwarding is disabled
- Firewall configuration
- Ensure SSH root login is disabled
- Ensure SSH Idle Timeout Interval is configured
- Ensure SSH access is limited

- Ensure no world writable files exist
- Ensure no unowned files or directories exist
- Ensure no ungrouped files or directories exist
- Performing maintenance activities as bpmaint on CIS system

Ensure IP forwarding is disabled

When BP products are installed as a cluster, this option cannot be disabled as the host OS acts as a router. The recommendation states “The following network parameters are intended for use if the system is to act as a host only.”

Firewall configuration

BP products provide its own mechanism for updating the firewall configuration (`bpfirewall`). Because of this, Blue Planet has provided a file to enable the CIS firewall recommendations.

Ensure SSH root login is disabled

In a clustered environment, allowing SSH root login between nodes simplifies administrative tasks. During BP product installation, the `/etc/ssh/sshd_config` file updates to allow root login between nodes in the cluster, using both the public IP and private network.

The added code will be similar as below:

```
# ## BP2 ## {"tag_name": "BP_ROOT_SSH_WHITELIST_V1", "kind": "START"}
Match Address 10.186.0.149,10.186.0.90,10.186.1.104,172.16.0.0/24,172.16.1.0/24,
172.16.2.0/24
    PermitRootLogin without-password
# ## BP2 ## {"kind": "END", "tag_name": "BP_ROOT_SSH_WHITELIST_V1"}
```

NOTE Do not remove this block if the general case for `PermitRootLogin` is set to no.

Ensure SSH Idle Timeout Interval is configured

This recommendation cannot be applied at this time.

Ensure SSH access is limited

If SSH is configured to specifically allow or deny users or groups, the following users should be explicitly allowed: `root, bpuser, bpadmin, bpmaint`.

Ensure no world writable files exist

Due to a limitation of the docker version embedded in BP product, this recommendation cannot be fully met. World writable files are created in the `/opt/Ciena/data/docker/containers` directory. This directory should be excluded from any audit of this recommendation.

Ensure no unowned files or directories exist

Due to a limitation of the docker version embedded in BP product, this recommendation requires a workaround. This workaround is required to prevent an audit finding against this CIS recommendation. It is not required for BP product operation.

Add the following three lines to the `/etc/passwd` file. The UIDs and GIDs match those for `galera`, `glusterfs`, and `graphite` respectively:

```
galera:x:105:107::/opt/ciena/bp2/:/sbin/nologin

glusterfs:x:2001:2001::/opt/ciena/bp2:/sbin/nologin

graphite:x:33:33::/opt/ciena/bp2:/sbin/nologin
```

NOTE Ensure that the UIDs do not conflict with any other users on the system.

Ensure no ungrouped files or directories exist

Due to a limitation of the docker version embedded in BP product, this recommendation requires a workaround. This workaround is required to prevent an audit finding against this CIS recommendation. It is not required for BP product operation.

Adding the following three lines to the /etc/group file. The GIDs match those for `galera`, `glusterfs`, and `graphite` respectively:

```
galera:x:107:galera

glusterfs:x:2001:glusterfs

www-data:x:33:
```

NOTE The GID for graphite falls within the reserved range; therefore, we will use the standard group (`www-data`). Also, ensure that the GIDs do not conflict with any other groups on the system.

Performing maintenance activities as bpmaint on CIS system

The bpmaint user is designed to have limited access privileges. To perform maintenance activities using the bpmaint user account on hardened systems like CIS, special privileges must be provided.

By default, the bpmaint user on the CIS system can collect the system logs and various statistics and can use the functionality as described in the non-CIS systems (Performing maintenance activities using bpmaint section in bp admin guide). The bpmaint user however cannot collect the application logs. Special privileges are required to collect the application log files, for which the bpmaint user must be a part of the docker group.

To add the bpmaint user to a docker group:

1. Login as bpadmin user on the target host system.
2. Run the following command:

   ```
   sudo usermod -a -G docker bpmaint
   ```

3. Perform the required maintenance activity by using the commands described in the non-CIS section.

   NOTE Usage must be restricted since the docker group is a part of the root group.

Enabling the CIS firewall profile

A pre-configured CIS profile has been provided for use with `bpfirewall`. Follow the below procedure to enable the CIS firewall profile:

NOTE If you have installed your product with GR, this procedure must be followed for each site.

1. Log in to the BP product hosts as `bpadmin`.
2. Change to the directory: `cd /etc/bp2/site`.
3. Using vi (or `sudo vi` if not root), edit the `bpfirewall.conf` file by appending the following lines to the end of the file:

   ```
   filtering_output: true
   filtering_forward: true
   ```

   NOTE Do not change any existing content in the file. If the file is empty, you must remove the **curled parenthesis {}** before proceeding with appending the command string as stated above.

4. Replace the default profile with the CIS profile:

```
sudo cp -p bpfirewall.cis bpfirewall
sudo chown bpuser:docker bpfirewall
```

NOTE If any custom rules had already been applied, they should be copied and applied again to the new file

5. Synchronize the site:

```
sudo bp2-site sync-site-config
```

6. Enable and restart the firewall:

```
sudo bpssh systemctl enable bpfirewall
sudo bpssh systemctl restart bpfirewall
```

Updating the security patches of operating system environment

BP product supports the option to install and operate on an RHEL/OL or CentOS operating system environment, which is regularly updated (via yum update) with the latest security patches. The host OS should be kept updated for stability and security reasons. Apply the Blue Planet system bundle before patching the system, and ensure that BP product is halted. Refer to respective *User Guide* for more details. For BP product with GR, Blue Planet recommends that you apply OS patches to the backup site (one site at a time), and verify that the GR synchronization status is healthy before patching the primary site.

Both of the following scenarios are supported:

- Updating RHEL/OL/CentOS 7.x OS in advance of installing BP Product
- Updating RHEL/OL/CentOS 7.x OS on installed BP2 sites

Updating RHEL/CentOS/OL 7.x OS in advance of installing BP Product

When you update the OS with latest security patches in advance of installing BP Product, you must do this for each host that is targeted to be part of an BP2 site. For details on how to update the OS on one specific host, see the system bundle installation instructions documentation available when downloading the system bundles from the Ciena software portal.

Updating RHEL/CentOS/OL 7.x OS on installed BP2 sites

Prerequisites

1. For BP product with Geographical Redundancy, Ciena recommends that you perform this procedure on the Backup site first and verify that the GR syncronization status is healthy before you perform this procedure on the Primary site.
2. This procedure requires root privileges on the specific hosts where the OS is updated

CAUTION You must stop the BP2 solution on the target BP2 site before you apply the OS updates. Refer to *Administration Guide*.

Workflow (for a specific BP2 site)

1. Log in as root user to Host 0 of the target site
2. Fully stop docker services

```
systemctl stop docker
```

3. Apply the OS update. For details on how to apply OS updates on a host, see the system bundle installation instructions documentation available when downloading the system bundles from the Ciena software portal.

   CAUTION For multi-host systems, do not perform the final reboot step at the end of the OS update until the OS update is first applied to all of the hosts of the BP2 site.

4. Reboot all of the hosts of the BP2 site; it is important not to stagger the reboots between the hosts. This step will automatically restart docker services and BP2 solution.

Co-existence of 3rd Party Software with BP Products

As a general consideration 3rd party tools performing any of the following functions are typically conflicting with BP products and therefore not approved to co-residence with BP products:

- Real time local file system scanning
- Anti-virus / anti-malware software with real-time scan active
- Tools performing real time collection of metrics
- Tools that alter the configuration of the OS where BP products is running, install additional drivers, and so on
- Tools attempting to establish a direct local connection with the docker engine component of BP products or that they attempt to deploy 3rd party images into the local docker engine repository of BP products

Contacting Blue Planet

Blue Planet Division Headquarters	7035 Ridge Road Hanover, MD 21076 +1 800-921-1144
Blue Planet Support	https://www.blueplanet.com/support
Sales and General Information	https://www.blueplanet.com/contact
Training	https://www.blueplanet.com/learning

For additional information, please visit https://www.blueplanet.com.

LEGAL NOTICES

THIS DOCUMENT CONTAINS CONFIDENTIAL AND TRADE SECRET INFORMATION OF CIENA CORPORATION, INCLUDING ITS SUBSIDIARY, BLUE PLANET SOFTWARE, INC., AND ITS RECEIPT OR POSSESSION DOES NOT CONVEY ANY RIGHTS TO REPRODUCE OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING THAT IT MAY DESCRIBE. REPRODUCTION, DISCLOSURE, OR USE IN WHOLE OR IN PART WITHOUT THE SPECIFIC WRITTEN AUTHORIZATION OF CIENA CORPORATION OR BLUE PLANET SOFTWARE, INC IS STRICTLY FORBIDDEN. EVERY EFFORT HAS BEEN MADE TO ENSURE THAT THE INFORMATION IN THIS DOCUMENT IS COMPLETE AND ACCURATE AT THE TIME OF PUBLISHING; HOWEVER, THE INFORMATION CONTAINED IN THIS DOCUMENT IS SUBJECT TO CHANGE. While the information in this document is believed to be accurate and reliable, except as otherwise expressly agreed to in writing, BLUE PLANET PROVIDES THIS DOCUMENT “AS IS” WITHOUT WARRANTY OR CONDITION OF ANY KIND, EITHER EXPRESS OR IMPLIED. The information and/or products described in this document are subject to change without notice. For the most up-to-date technical publications, visit https://my.ciena.com.

Security

Ciena® cannot be responsible for unauthorized use of equipment and will not make allowance or credit for unauthorized use or access.