



BeyondTrust

SecureAuth (Arculix) Push Notification for Privilege Management for Unix & Linux

Table of Contents

SecureAuth (Arculix) Push Notification for Privilege Management for Unix and Linux ..	3
Overview	3
Configure Privilege Management for Unix and Linux Policy	4
Create Group of Allowed Commands	4
Create a DatabaseAdmins Role	5
Assign test user to DBA Commands Group	5
Messages	5
Script Policy	6
User Experience	8

SecureAuth (Arculix) Push Notification for Privilege Management for Unix and Linux

This guide describes the steps to set up Arculix push notifications (to mobile) for PMUL managed elevation requests.

Overview

Arculix by SecureAuth allows BeyondTrust customers to deploy *passwordless continuous authentication* for Privilege Management for Unix and Linux (PMUL), while providing a flexible and frictionless user experience.

When Unix & Linux users require access to privileged commands or files, elevation requests can be approved with a mobile phone and the **Arculix It'sMe App** available for Android and Apple.

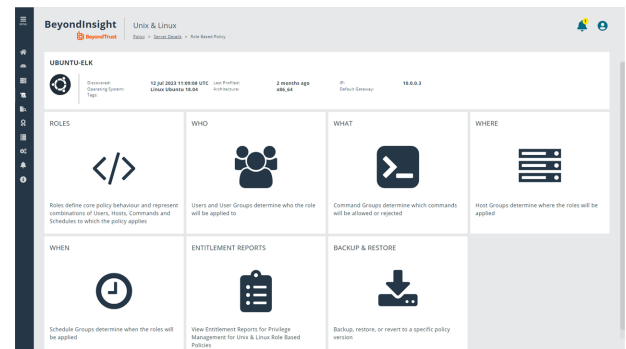
i You need a working test user with the Arculix mobile app to receive the Push notification. For more information, see the [Arculix Mobile app user guide](https://docs.secureauth.com/arculix/en/arculix-mobile-app-user-guide.html), at <https://docs.secureauth.com/arculix/en/arculix-mobile-app-user-guide.html>.

Configure Privilege Management for Unix and Linux Policy

A **Role-Based Policy** is required for the configuration described in this guide. However the Policy Script used could also be used for **Server-Based Policy**, but this is beyond the scope of this guide.

i Some of the steps listed in this section refer to our *BeyondInsight for Unix & Linux (BIUL)* application. For more information, see the *BeyondInsight for Unix & Linux User Guide*, at <https://www.beyondtrust.com/docs/privilege-management/console/beyondinsight-unix-linux/user/index.htm>.

You must first configure a Privilege Management for Unix and Linux (PMUL) Role-Based Policy, using the BeyondInsight for Unix & Linux (BIUL) application.

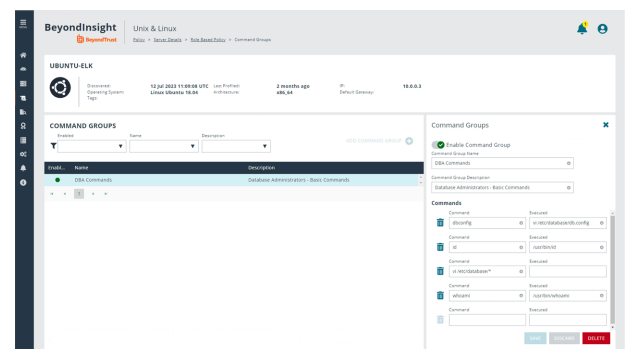


In the steps that follow, we will use a simplistic example based on a Database Admin Use Case to demonstrate how the integration works.

Create Group of Allowed Commands

We need to create a group of allowed commands. You will allow for the modification of a configuration file, and also some test commands. The **Command** and **Executed** parameter entries set in this example are as below:

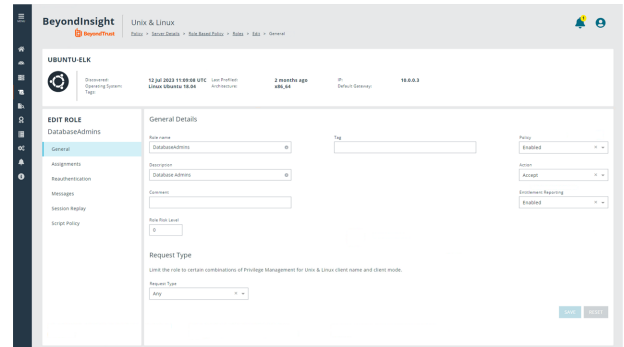
- dbconfig, vi /etc/database/db.config
- id, /usr/bin/id
- vi /etc/database/*, (leave blank)
- whoami, /usr/bin/whoami



Create a DatabaseAdmins Role

Create a role called **DatabaseAdmins**. The parameters set in the example are as follows:

- **Role Name:** DatabaseAdmins
- **Description:** Database Admins
- **Policy:** Enabled
- **Action:** Accept
- **Entitlement Reporting:** Enabled
- **Role Risk level:** 0
- **Request Type:** Any

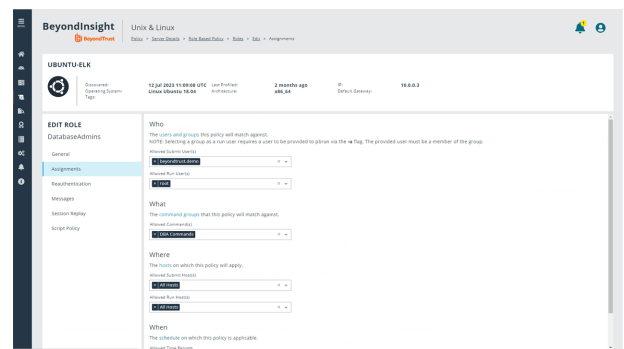


The screenshot shows the 'EDIT ROLE' configuration page for 'DatabaseAdmins' in the BeyondInsight console. The 'General' tab is selected, showing fields for Role Name (DatabaseAdmins), Description (Database Admins), Policy (Enabled), Action (Accept), Entitlement Reporting (Enabled), and Request Type (Any).

Assign test user to DBA Commands Group

Assign a working test user to the **DBA Commands** group that you created. The parameters set in the example are as follows:

- **Allowed Submit User(s):** beyondtrust.demo
- **Allowed Run User(s):** root
- **Allowed Commands:** DBA Commands
- **Allowed Submit Host(s):** All Hosts
- **Allowed Run Host(s):** All Hosts
- **Allowed Time Periods:** Any Time

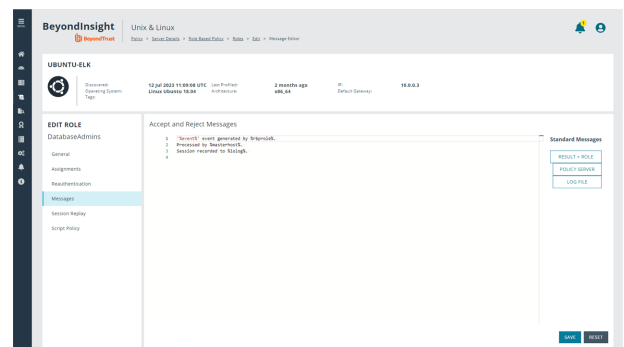


The screenshot shows the 'EDIT ROLE' configuration page for 'DatabaseAdmins' in the BeyondInsight console. The 'Assignments' tab is selected, showing the 'Who' (beyondtrust.demo), 'What' (DBA Commands), 'Where' (All Hosts), and 'When' (Any Time) fields.

Messages

Optionally, you can enable some messages. The **Accept and Reject Messages** script set in the example is as follows:

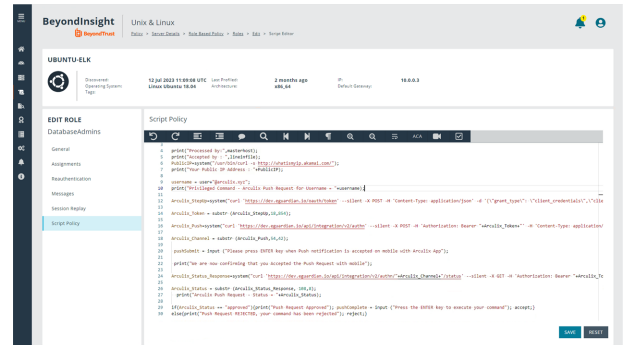
- 1 '%event%' event generated by %rbprole%.
- 2 Processed by %masterhost%.
- 3 Session recorded to %iolog%.



The screenshot shows the 'EDIT ROLE' configuration page for 'DatabaseAdmins' in the BeyondInsight console. The 'Messages' tab is selected, showing the 'Accept and Reject Messages' script set in the example.

Script Policy

You need to create a Script Policy to allow for the interaction with Arculix and the push notification on the user's mobile app. See the script policy example below.



```
# Arculix API service account - Oauth Client Credentials
client_id = "123456"
client_secret = "abc123"

print("Processed by:",masterhost);
print("Accepted by : ",lineinfile);
PublicIP=system("/usr/bin/curl -s http://whatismyip.akamai.com/");
print("Your Public IP Address : "+PublicIP);

username = user+"@arculix.xyz";
print("Privileged Command - Arculix Push Request for Username = "+username);

Arculix_StepUp=system("curl 'https://dev.eguardian.io/oauth/token' --silent -X POST -H 'Content-Type: application/json' -d '{\"grant_type\": \"client_credentials\", \"client_id\": \""+client_id+\"\", \"client_secret\": \""+client_secret+\"\", \"scope\": \"public\"}'");

Arculix_Token = substr (Arculix_StepUp,18,854);

Arculix_Push=system("curl 'https://dev.eguardian.io/api/integration/v2/authn' --silent -X POST -H 'Authorization: Bearer "+Arculix_Token+" -H 'Content-Type: application/json' -d '{\"credential_type\": \"password_less_login\", \"auth_credentials\": {\"username\": \""+username+"\"}, \"type\": \"Event Check\", \"message\": \"BeyondTrust Step Up Auth Request\", \"event\": \"Continuous-Auth\", \"auth_factor\": [\"push\"]}'");

Arculix_Channel = substr (Arculix_Push,54,42);

pushSubmit = input ("Please press ENTER key when Push notification is accepted on mobile with Arculix App");

print("We are now confirming that you Accepted the Push Request with mobile");

Arculix_Status_Response=system("curl 'https://dev.eguardian.io/api/integration/v2/authn/'+Arculix_Channel+'/status' --silent -X GET -H 'Authorization: Bearer "+Arculix_Token+"");

Arculix_Status = substr (Arculix_Status_Response, 108,8);
print("Arculix Push Request - Status = "+Arculix_Status);
```

```
if(Arculix_Status == "approved"){print("Push Request Approved"); pushComplete = input ("Press the  
ENTER key to execute your command"); accept;}  
else{print("Push Request REJECTED, your command has been rejected"); reject;}
```

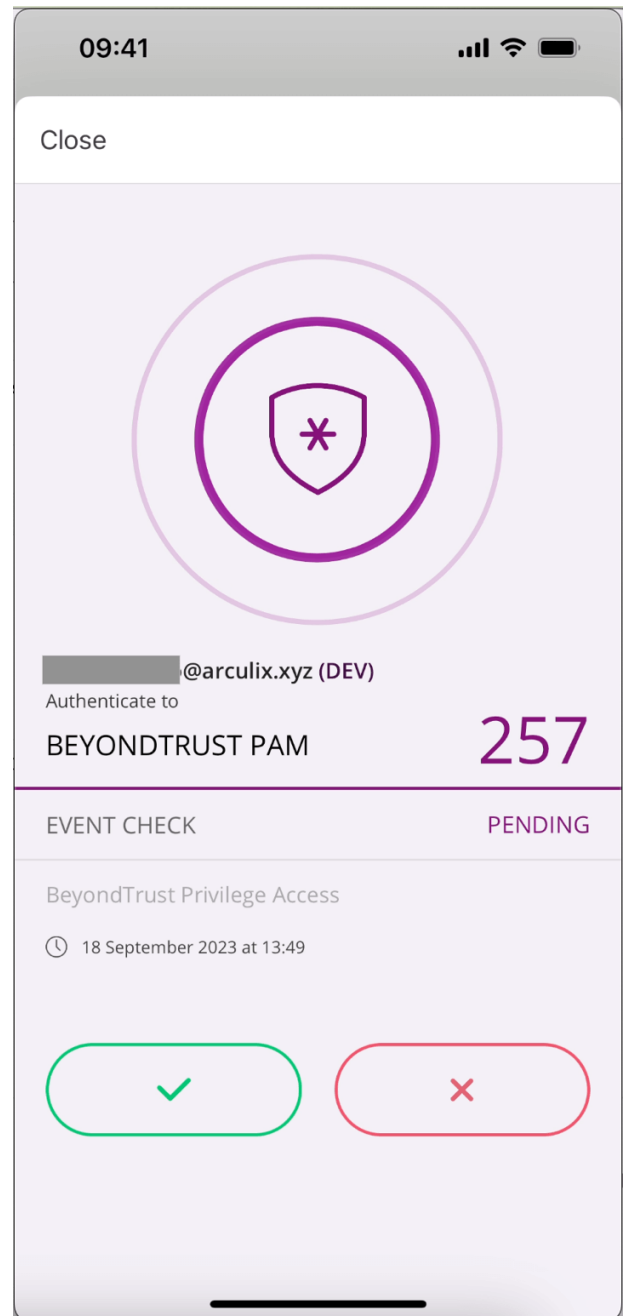
Here is a sample scenario of an attempt to modify a database configuration file.

```
ubuntu-elk1fab.blu.cloud - PuTTY
beyondtrust.demo@ubuntu-elk:/$ whoami
beyondtrust.demo
beyondtrust.demo@ubuntu-elk:/$ vi /etc/database/db.config
```

[illegible]

```
ubuntu-elk@ubuntu-cloud: ~$ sudo whoami
beyondtrust.demo@ubuntu-elk:/$ whoami
beyondtrust.demo
beyondtrust.demo@ubuntu-elk:/$ vi /etc/database/db.config
beyondtrust.demo@ubuntu-elk:/$ pburn vi /etc/database/db.config
"Accep" event generated by DatabaseAdmins.
Processed by ubuntu-elk.
Session recorded to .
Processed by : ubuntu-elk
Accepted by : Role Based Policy 'DatabaseAdmins'
Your Public IP Address : 206.193.148.51
Privileged Command - Arculix Push Request for Username = beyondtrust.demo@arcuili
x.xyz
Please press ENTER key when Push notification is accepted on mobile with Arculix
App
```


beyondtrust.demo receives a push notification on a mobile app similar to this one.



```
ubuntu-eltk.8ta.bta.cloud - PuTTY
beyondtrust,demo@ubuntu-eltk:~$ whoami
beyondtrust,demo
beyondtrust,demo@ubuntu-eltk:~$ vi /etc/database/db.config
beyondtrust,demo@ubuntu-eltk:~$ sburui vi /etc/database/db.config
'Accept' event generated by DatabaseAdmins.
Processed by ubuntu-eltk.
Session recorded to .
Processed by : ubuntu-eltk
Accepted by : Role Based Policy 'DatabaseAdmins'
Your Public IP Address : 206.198.149.51
Privileged Command - Arculix Push Request for Username = beyondtrust,demo@arculix.kxyz
Please press ENTER key when Push notification is accepted on mobile with Arculix App
We are now confirming that you Accepted the Push Request with mobile
Arculix Push Request - Status = approved
Push Request Approved
Press the ENTER key to execute your command
```

[illegible]

```

beyondtrust.dem@ubuntu-elk:~$ ssh ubuntu-elk
beyondtrust.dem@ubuntu-elk:~$ pbrun vi /etc/database/db.config
'Accept' event generated by DatabaseAdmins.
Processed by ubuntu-elk.
Session recorded to .
Processed by: ubuntu-elk
Accepted by : Role Based Policy 'DatabaseAdmins'
Your Public IP Address : 206.198.148.51
Privileged Command - Arculix Push Request for Username = beyondtrust.dem@arculix
.com
Please press ENTER key when Push notification is accepted on mobile with Arculix
App
We are now confirming that you Accepted the Push Request with mobile
Arculix Push Request - Status = rejected
Push Request REJECTED, your command has been rejected
beyondtrust.dem@ubuntu-elk:~$ pbrun23.1.0-12[29138]: Request rejected by pmmasterd on ubuntu-elk.
beyondtrust.dem@ubuntu-elk:~$

```