RTCaaS Middleware Installation and Configuration

Introduction

The RTCaaS Middleware translates secure authenticated REST API calls from external components such as the SecureX Orchestrator to calls against ISE over pxGrid or the Active Directory over LDAP.

- Broker between SecureX RTC and ISE (and AD)
- REST API
  - rtcGetUserByIP. (retrieves username and MAC from IP from Cisco ISE)
  - rtcGetUserInfo (retrieves user info from AD: userPrincipalName, AD groups, Lastlogon...)
  - rtcANCapply

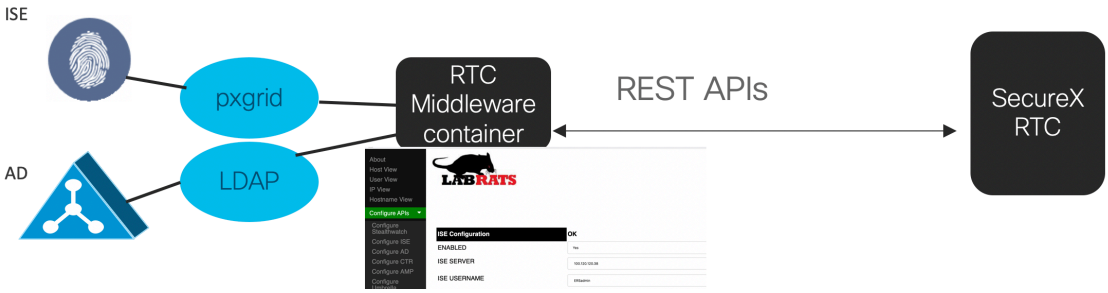


Figure 1 RTCaaS Middleware - Overview

Currently the following calls are available

- Retrieve username and MAC address from IP address
- Retrieve information about user such as group membership, last login etc from the Active Directory, given a username
- Apply or clear ISE ANC policy for an endpoint given IP or MAC

1. Run docker-compose up

On a machine with docker and docker-compose installed, download the docker-compose.yml file and run docker compose.

```
ubuntu@ip-100-125-2-91:~/RTC$ ls
docker-compose.yml
ubuntu@ip-100-125-2-91:~/RTC$ docker-compose up
Creating network "rtc_my-net" with driver "bridge"
Pulling db (mongo:)...
latest: Pulling from library/mongo
```

Figure 2 Installation with docker-compose

2. Login to RTCaaS Middleware Web GUI

Login to the middleware by browsing to https://<your ip address>. You will have to accept certificate warning.

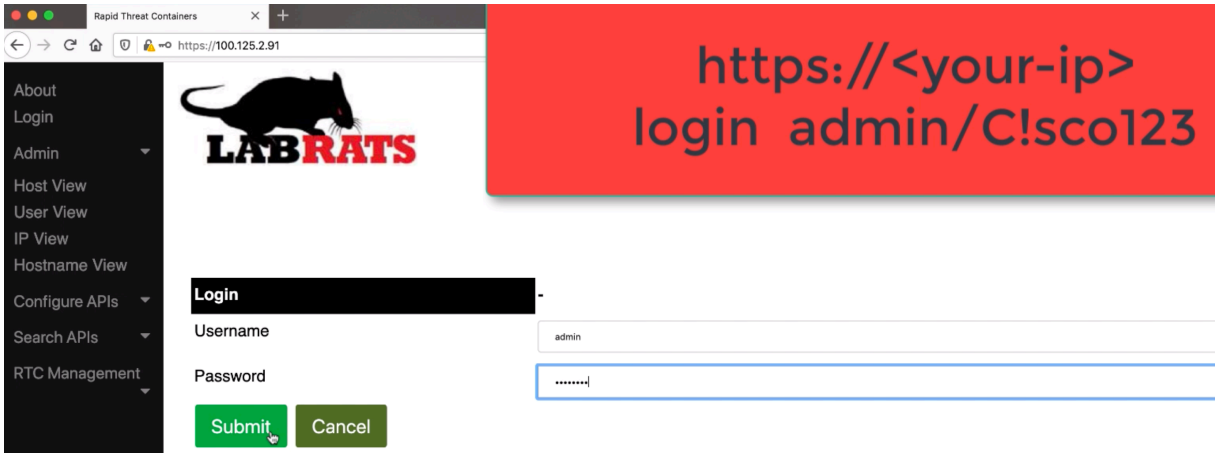


Figure 3 Logging in to the Web UI

Username admin.
Password C!sco123.

3. Hardening: Configure Trusted Networks.

By default, all IP addresses can connect to the middleware. Change this by
- Add a trusted network for your admins (e.g. bastion hosts)
- Add a trusted network for your SXO
- Delete the default network

You can find out the public IP of the SXO by creating a trivial workflow that connects to a REST API such as ipify.com

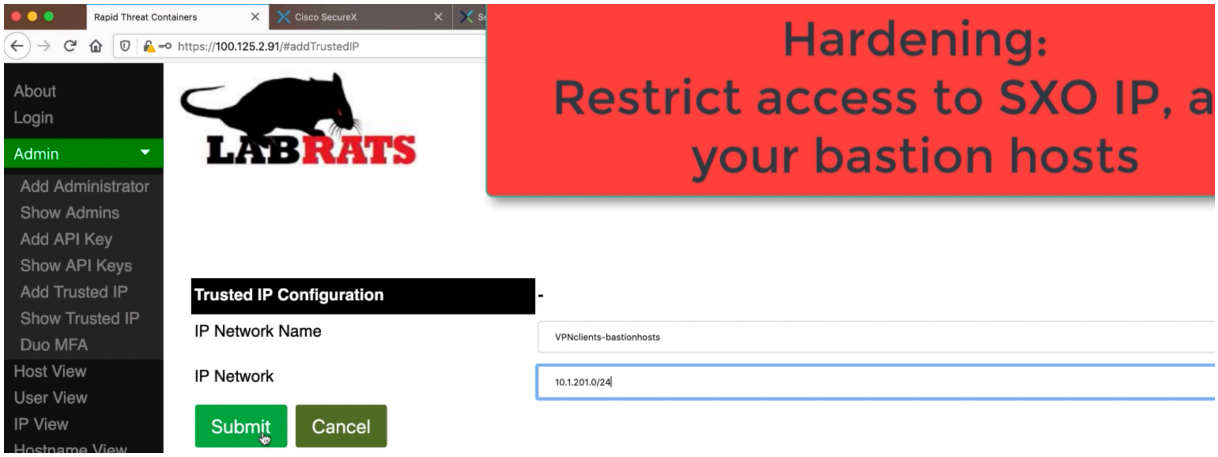


Figure 4 Adding a Trusted IP

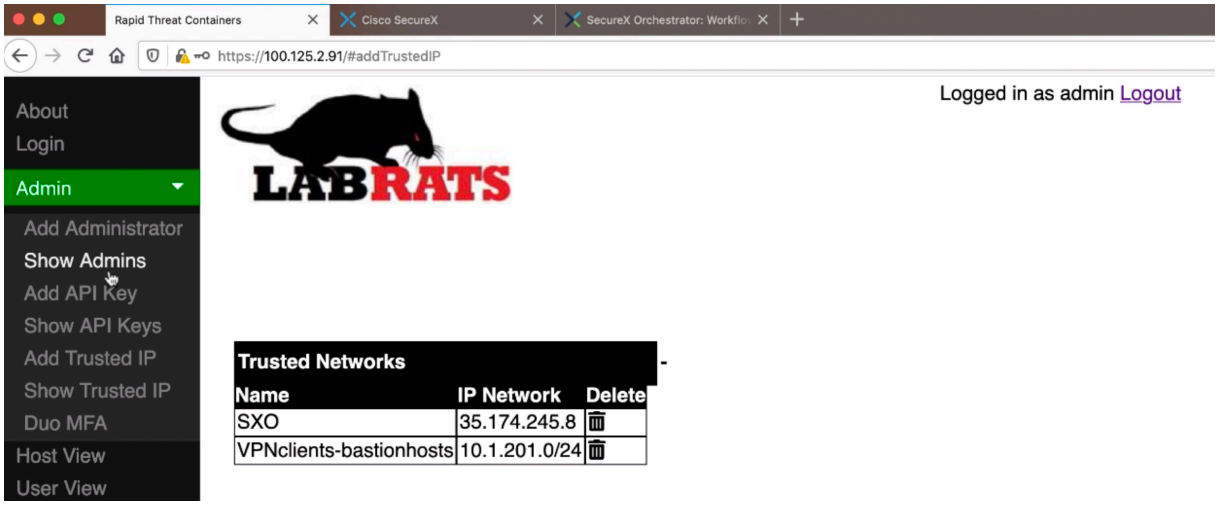


Figure 5 After adding trusted IPs and deleting default

4. Hardening: Add New Admin User and default admin

Add a new admin user with a password.

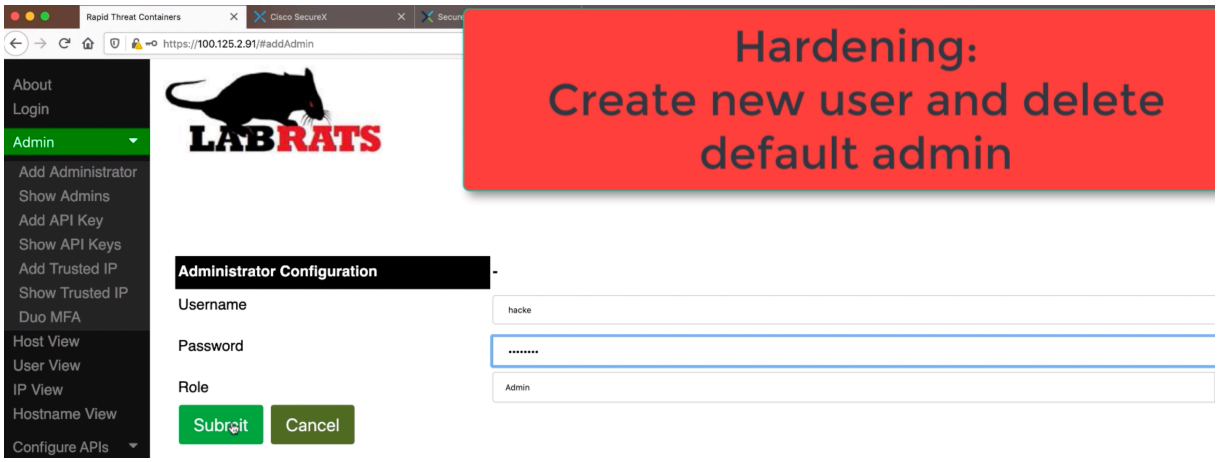


Figure 6 Adding a new admin user

Delete the old admin user (admin).

It is advisable to first logout and test login as the newly created user, before deleting admin account. If you want to apply Duo MFA as in step 5, it is best to wait until after Duo MFA has been implemented and tested before deleting the admin user.

5. Hardening: Configure Duo Protect to enable MFA for users.

   The middleware supports MFA with Duo.

   Enter the duo host, I-KEY and S-KEY (from the Duo Portal configuration).

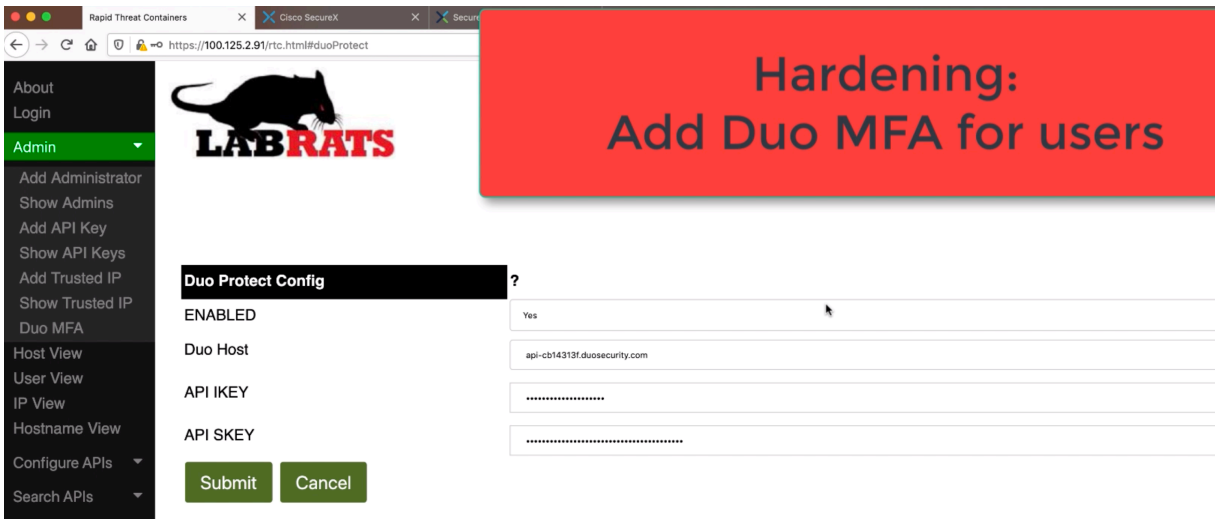


Figure 7 Duo Protect Configuration

After configuring Duo Protect, all users will require to use MFA except any user called “admin”. You can test by logging out and in again with any username except “admin”.

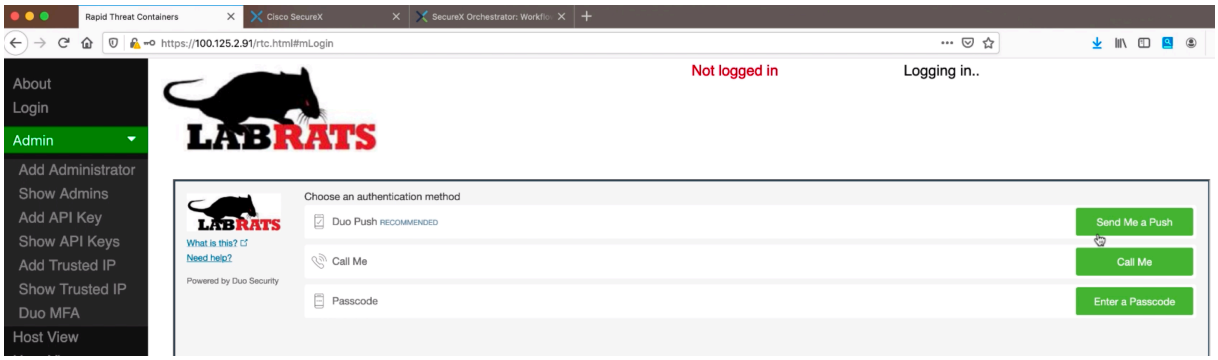


Figure 8 Logging in with Duo MFA

6. Configure and Test the Active Directory Interface

Configure the Active Directory interface with
- Server ip
- Base DN
- The username and password of a user that can browse the Active Directory

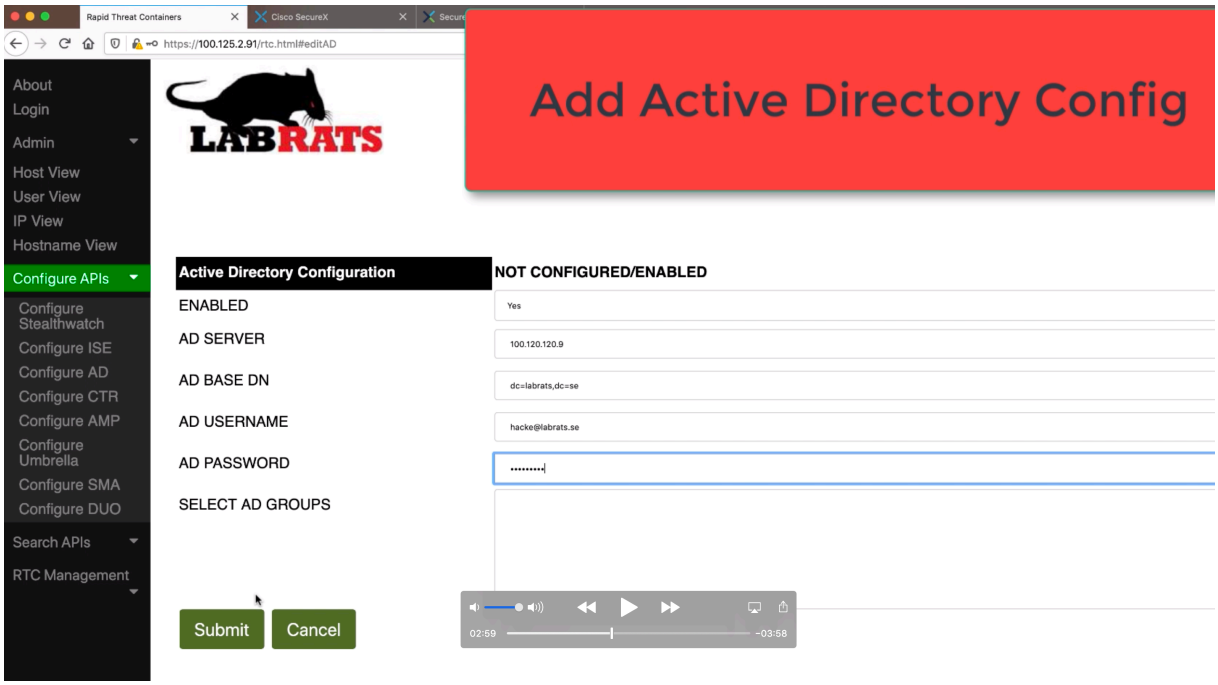


Figure 9 Configuring Active Directory

You can test the Active Directory Interface by searching for a user that exists in thee active directory.

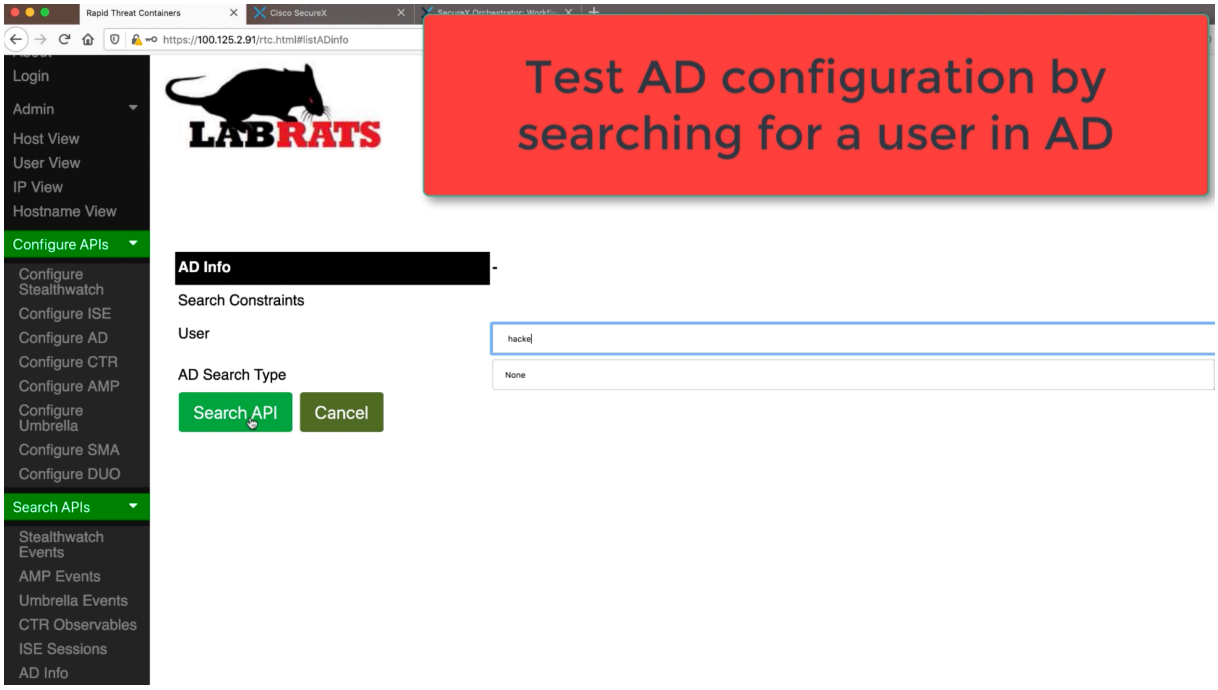


Figure 10 Test AD configuration

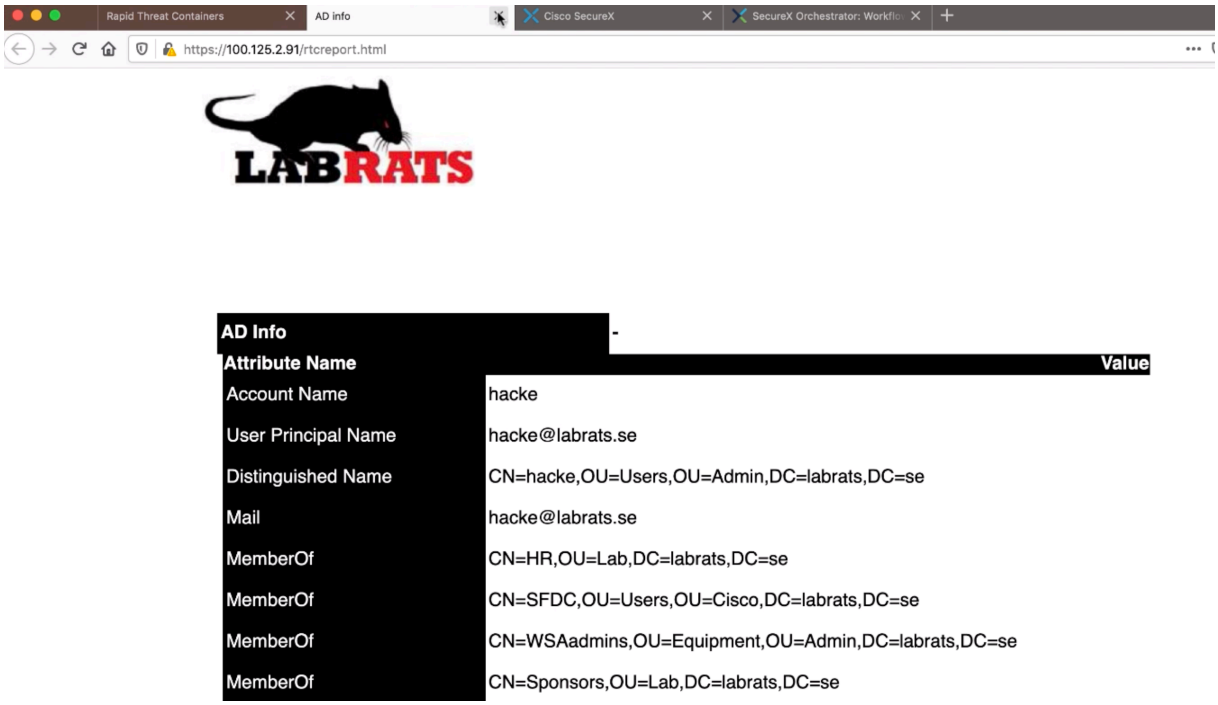


Figure 11 Test AD result (displaying groups and other info)

7. Configure and Test the ISE pxGrid Interface

The middleware currently only supports authentication with (dynamically generated) pre-shared key to the pxGrid bus. Manual Approval in the ISE GUI is therefore required.

Prepare by opening a separate tab in your browser and login to ISE, browsing to the page with pxGrid clients (ISE : Administration/pxGrid Services)

In the middleware, configure the ISE API by specifying
- IP address of ISE server
- Username of ERS user (should be ERSadmin in ISE)
- Password of ERS user
- Nodename of pxGrid client, e.g. the middleware (must be unique)

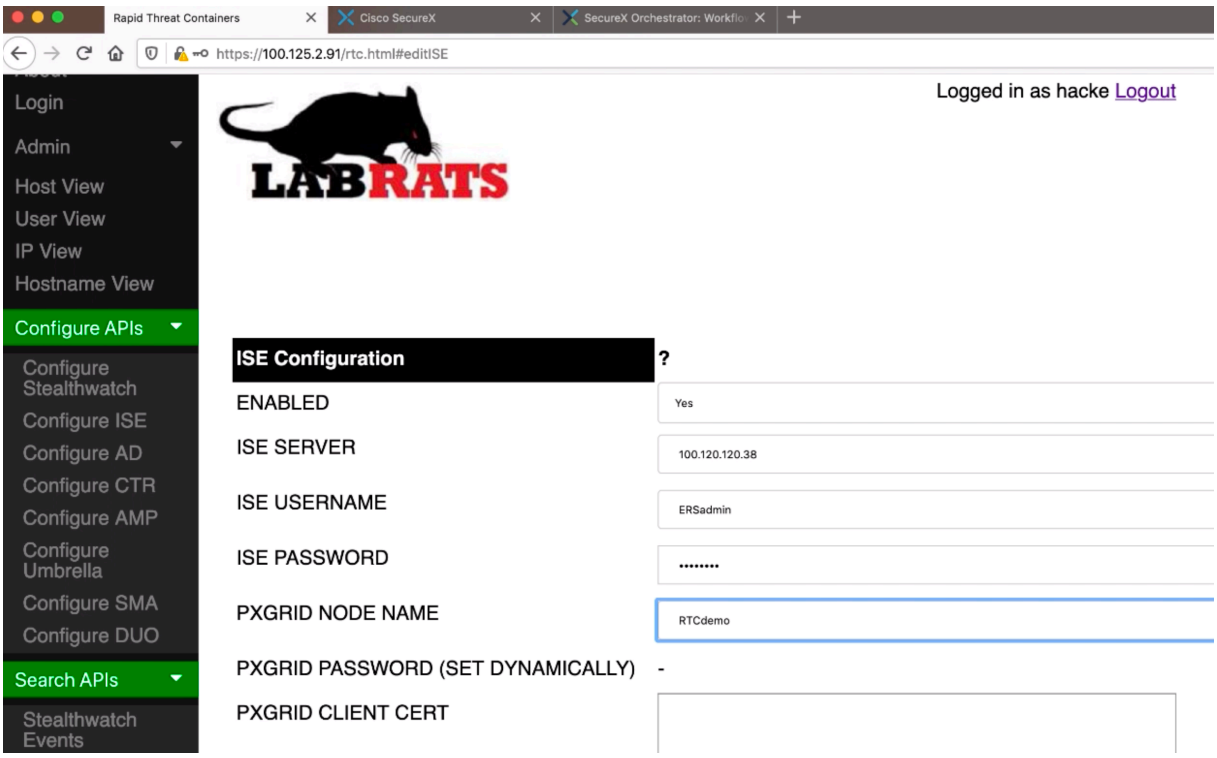


Figure 12 Configure ISE

After submitting the ISE configuration, you must approve the client in the ISE GUI. This has to be done within 4 minutes from configuring ISE in the middleware.

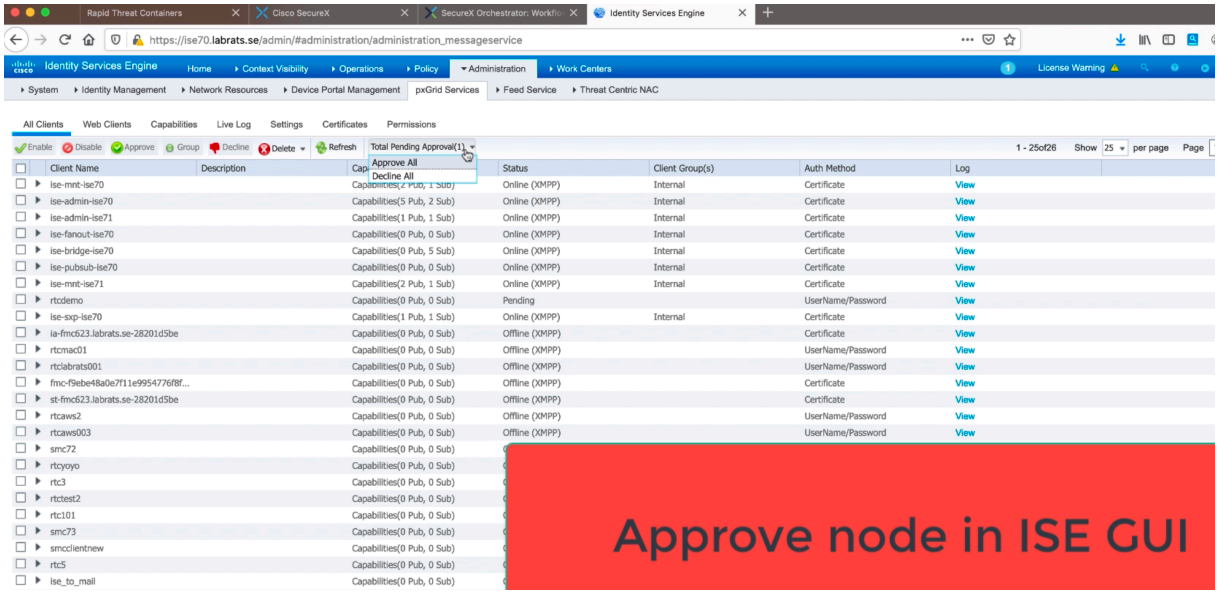


Figure 13 Approve pxGrid node in ISE GUI

You can test the ISE pxGrid connection by searching ISE API, which should show all connected sessions if it works. In Middleware GUI, Search APIs -> ISE Sessions.

ISE Sessions -

User Name	MAC	IP	SGT	OS	Profile	NAS	Port
CN=mordiac,OU=WiredDot1X,OU=Lab,DC=labrats,DC=se	00:0C:29:15:69:FA	10.1.33.13	Employees	Windows 7 Enterprise	Windows7-Workstation	10.1.40.253	GigabitEthe
CN=mephisto,OU=WiredDot1X,OU=Lab,DC=labrats,DC=se	00:0C:29:76:99:F8	10.1.33.32	Employees	Windows 7 Professional	Windows7-Workstation	10.1.40.253	GigabitEthe
CN=VMRAT31,OU=WiredDot1X,OU=Lab,DC=labrats,DC=se	00:0C:29:CF:3F:15	169.254.87.251	Employees	Windows 10 Enterprise	Windows7-Workstation	10.1.40.253	GigabitEthe
CN=garfield,OU=WiredDot1X,OU=Lab,DC=labrats,DC=se	00:50:56:8B:95:4A	10.1.33.33	Employees	Windows 10 Enterprise	Windows10-Workstation	10.1.40.253	GigabitEthe

Figure 14 Output from Search API / ISE Sessions

8. Generate API key

The API calls are authenticated with an API key. Generate the API key and copy it to notepad or similar.

Figure 15 Add API key for SecureX Orchestrator.

The API key has to be present in all calls in a custom header RTCAuthz.

9. SXO – Create Secure String Variable with API Key

In SXO, create a Secure String Variable with the API key from previous step.

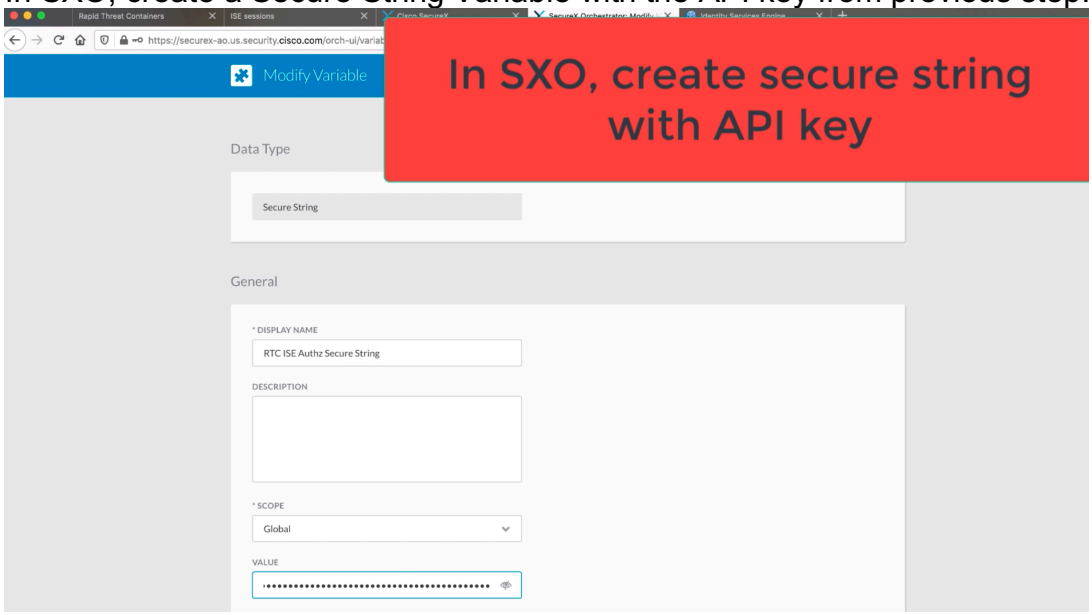


Figure 16 Create Secure String Variable with API key

10. SXO – Create Target

In SecureX Orchestration, create a Target that points to your RTC Middleware.

- Protocol HTTPS
- Specify hostname/IP of your Middleware
- Path /cgi-bin
- Check Disable Server Certificate Validation

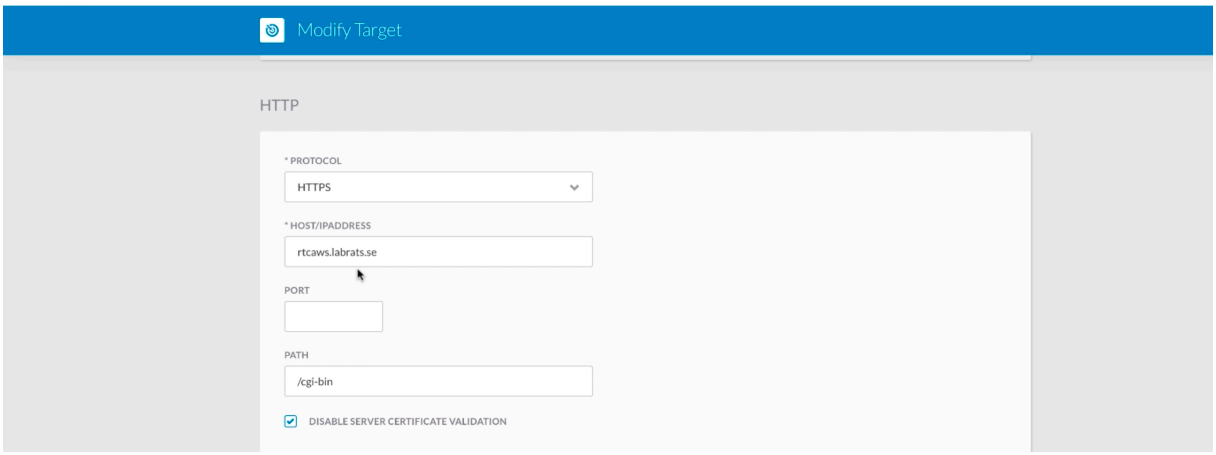


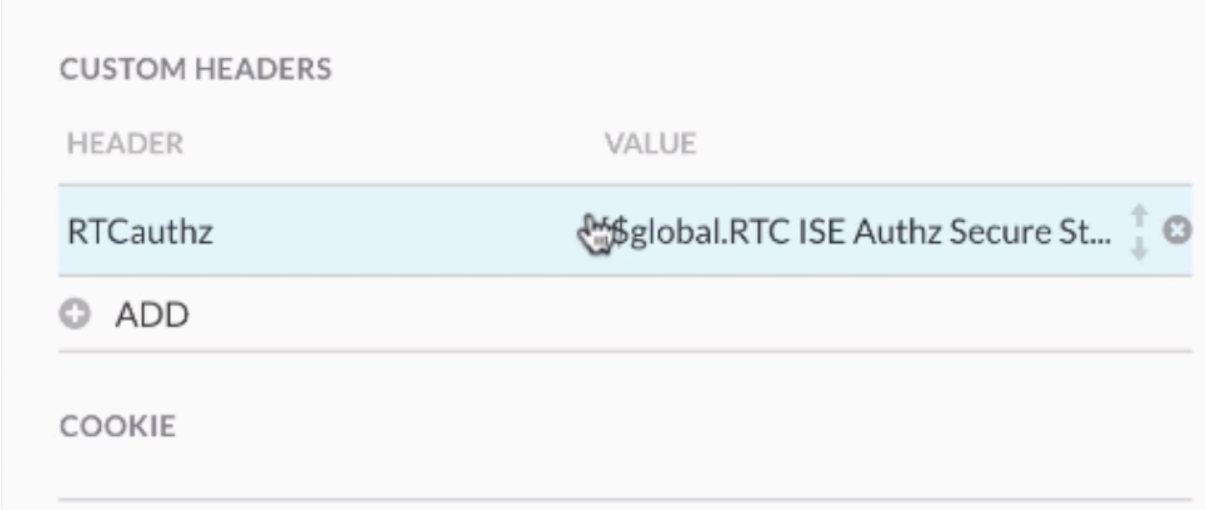
Figure 17 SXO - specify target

11. SXO - Creating web requests to the RTC Middleware

In SXO when you create a workflow with a web request to the RTC middleware, ensure that

Target is set to the target defined in previous step

You add a custom header named RTCAuthz with the value of the Secure String from previous step.



Ensure that the request relative url and body are set according to the API request.

API call	Relative URL	Method	Body
Get User By IP	/rtcAPIgetUserByIP.py	POST	{“ip”: “10.1.33.33”}
Get User Info from Username	/rtcAPIgetUserInfo.py	POST	{“user”:”hacke”}
Apply ANC	/rtcAPIancApply.py	POST	{“ip”:”10.1.33.33”}
Clear ANC	/rtcAPIancClear.py	POST	{“ip”: “10.1.33.33”}