IBM Security

QRadar SOAR Bootcamp Training

IBM QRadar SOAR

IBM

QRADAR SOAR TRAINING

Labs plan

Lab 1 : Prepare the environment
Lab 2 : Incident Creation & Editing
Lab 3 : Working with the Privacy Module and Breach
Lab 4 : Reports & Dashboards
Lab 5 : Field & Tab Customization
Lab 6 : Incident Types, Phases, and Tasks
Lab 7 : Scripts
Lab 8 : Rules
Lab 9 : Additional script and rule
Lab 10 : Inbound Email
Lab 11 : Playbooks
Lab 12 : Install the Integration Server
Lab 13 : Installing QRadar SOAR Circuits as a Service
Lab 14 : install the apphost package
Lab 15 : install your first app
Lab 16 : Install fn_utilities app
Lab 17 : Create new Custom Palybook that uses our Shell Command Function
Lab 18 : Install nmap in AppHost and remote control to VM
Lab 19 : Install the Components File for App Host

Specific last half day for Partners & IBMers: Link your QRadar to SOAR
Lab 20 : Reserve a QRadar for demo on TechZone – Must Have a Partner IBMiD or be IBMers for the Labs
Lab 21 : Configure your QRadar to send Alerts to SOAR
Lab 22 : Configure your SOAR to Query QRadar with 2 apps
Lab 23 : Create offense and see them in SOAR

QRADAR SOAR TRAINING

CEST Agenda

EMEA & APAC

Day 1:

8h30 – 10:00 SOAR presentation and positioning – Demo of a successful automated playbook

10:00 – 10:15 Break

10:15 – 12:00 Labs 1-2-3-4 + Q&A

Day 2:

8h30 – 10:00 Labs 5-6-7-8

10:00 – 10:15 Break

10:15 – 12:00 Labs 9-10 + Q&A

Day 3:

8h30 – 10:00 Playbook Design – Lab 11

10:00 – 10:15 Break

10:15 – 12:00 Labs 12-13-14-15 + Q&A

Day 4:

8h30 – 10:00 Labs 16-17

10:00 – 10:15 Break

10:15 – 12:00 18-19 + Q&A

Day 5:

8h30 – 10:00 Labs 20-21

10:00 – 10:15 Break

10:15 – 12:00 22-23 + Q&A

QRADAR SOAR TRAINING

EST Agenda

US & CANADA

Day 1:

9h30 – 11:00 SOAR presentation and positioning – Demo of a successful automated playbook

11:00 – 11:15 Break

11:15 – 13:00 Labs 1-2-3-4 + Q&A

Day 2:

9h30 – 11:00 Labs 5-6-7-8

11:00 – 11:15 Break

11:15 – 13:00 Labs 9-10 + Q&A

Day 3:

9h30 – 11:00 Playbook Design – Lab 11

11:00 – 11:15 Break

11:15 – 13:00 Labs 12-13-14-15 + Q&A

Day 4:

9h30 – 11:00 Labs 16-17

11:00 – 11:15 Break

11:15 – 13:00 Labs 18-19 + Q&A

Day 5:

9h30 – 11:00 Labs 20-21

11:00 – 11:15 Break

11:15 – 13:00 Labs 22-23 + Q&A

QRADAR SOAR TRAINING

Lab 1

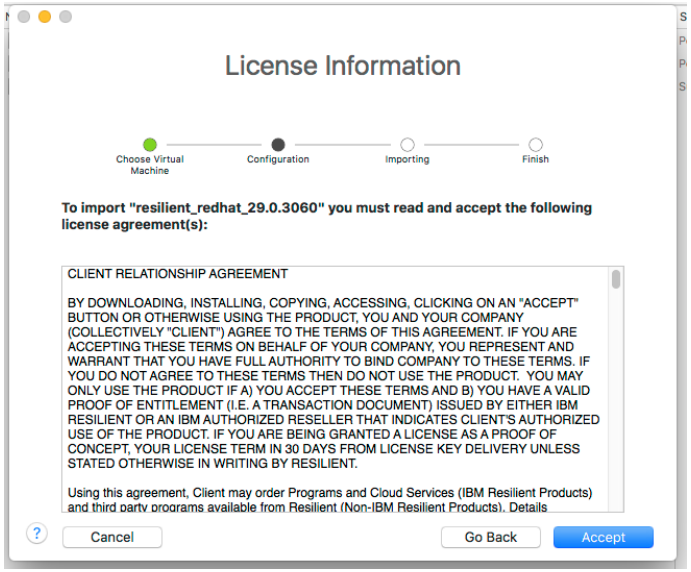
If you are working under a remote Saas Lab like ***emea-training***, most of this lab has been done to setup the remote saas environement. Please check each page if some action need to be done

Goal :

- Initialize a working environment
- Initialize a prepared Virtual Machine

Import the OVA To VMware

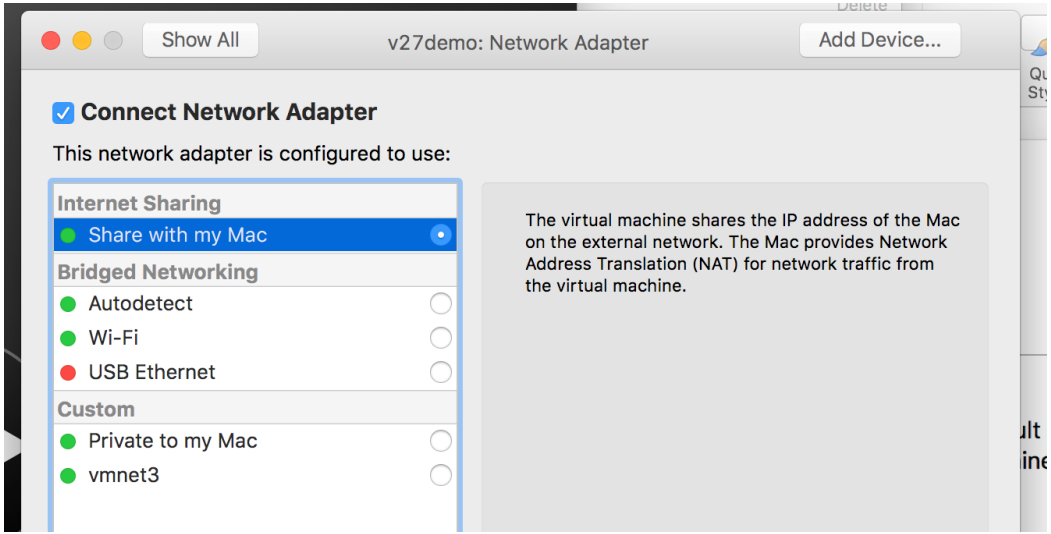
Not Able on Remote Saas ***emea-training*** Lab



Import to VMware or VirtualBox

We have found that at different locations, we have had to change the network settings from Bridge to NAT.

Start with NAT and if you have problems later on, please try changing to Bridge.



Create your passwords

Not Applicable on Remote Saas ***emea-training*** Lab

```
Changing password for user root.
New password:
BAD PASSWORD: The password contains less than 4 character classes
Retype new password:
passwd: all authentication tokens updated successfully.

Changing password for user appadmin.
New password:
BAD PASSWORD: The password contains less than 4 character classes
Retype new password:
passwd: all authentication tokens updated successfully.

Please note, SSH is disabled for root.
In the future, please use appadmin for SSH to this system.
```

Create a password for the “**root**” and “**appadmin**” account

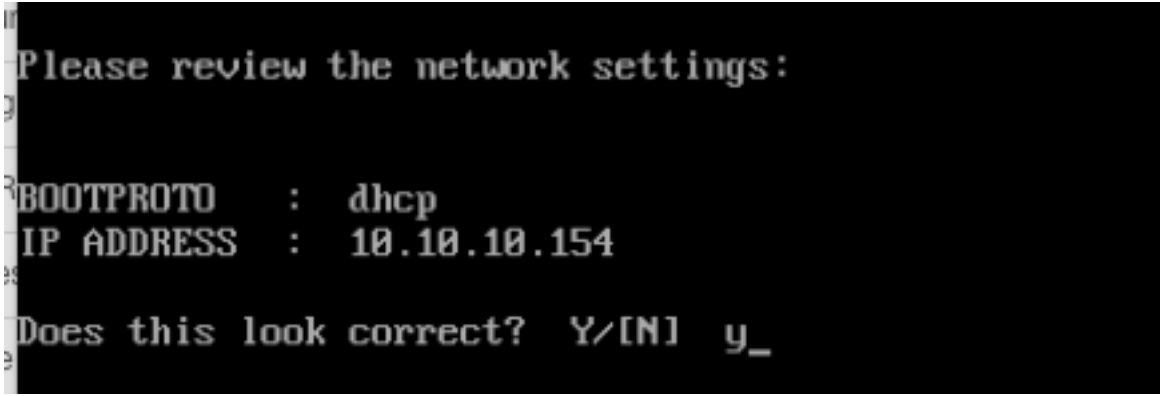
*Tips: the password “**resilient**” works well for demo on any keyboards*

Get the IP for QRadar SOAR

Not Applicable on Remote Saas ***emea-training*** Lab

Choose dhcp and check that QRadar SOAR found an IP.

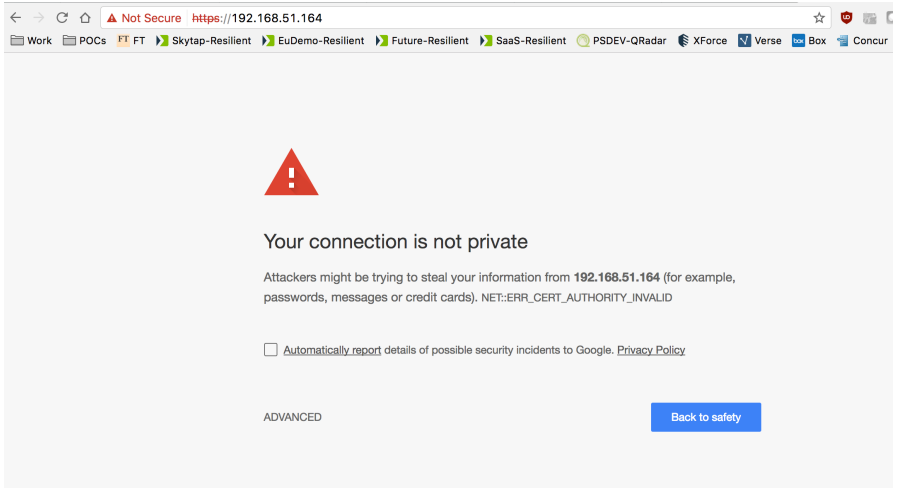
If it has not, please try changing your network card for the virtual machine, then restart.
As a last resort, or if required, you can set a static IP.



The system will then reboot.

Access QRadar SOAR GUI

Not Applicable on Remote Saas ***emea-training*** Lab
Please use:
https://emea-training.poc.resilientsystems.com/



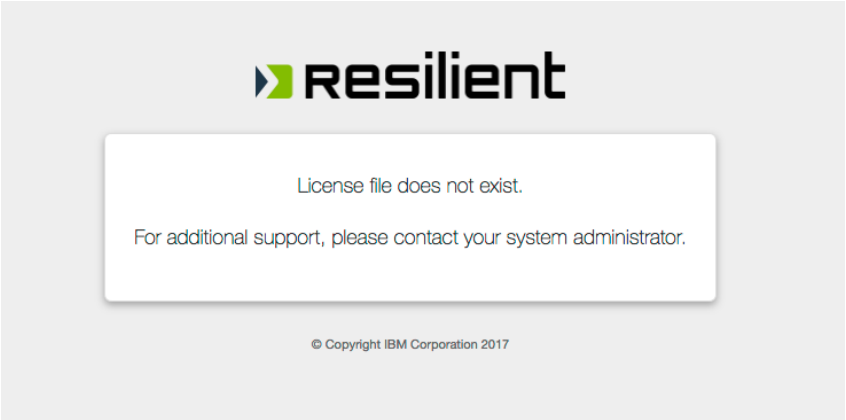
Go to https://<resilient_ip>

You need to replace <resilient_ip> with the ip of your machine.
If you did not get an ip in the dchp settings section, you can use this command to find it:

ifconfig

You will get a self-signed certificate warning, click through this.

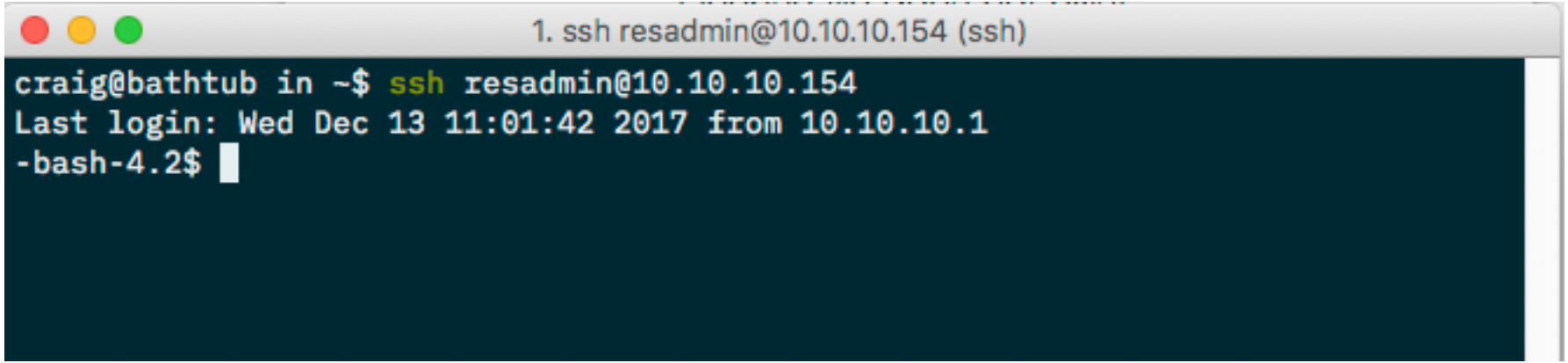
You will see a warning screen that you have no license installed.



Login to QRadar SOAR CLI

Not Applicable on Remote Saas ***emea-training*** Lab

Login with appadmin via SSH using terminal (Mac) or putty (Windows).



You can use ‘sudo su –’ to gain root or use sudo for root requiring commands

Import license

Not Applicable on Remote Saas ***emea-training*** Lab

```
-bash-4.2$ sudo license-import
Please enter or paste in the license:
-----BEGIN LICENSE-----
AB+LCAAAAAAAAACFkMtqwzAQRX8lzNqtH7IcWxBooS6F0i5aAt2O5DE1sS2jB00o/feOyS
rZRAIN3EHnzp1fcDh3dgIFXfpyfPo4VF9Nu4METPTBTuTecSLufrabu83+lZ/2rX3kvicT
3RBO7Yx6pA5UcJESoOMyOAyDnUHlslxvna0ngejJzWeaoR5NJfJaYiH6Sme1kSLXhegyWU
udM39B73+sYzDkjSh1gZq2ZdNtRa+l7mWhC90YiSQRzmwPao7jmICjxe7dyD+/Q1i8StNb
fuqWxcPKvDdW+JPnMqmyFOmqrZtAZOseR8/50azZ/bPjoDz+4Xo9kS4FHvxSMHgl4ILmQv
r7B4Zs/3u1AQAA:MCwCFCjzDQLi8Sw4kggESwcWWG6ux3UpAhRBx5+kRHCIG+xS/qevUMd
l+Kzefg==
-----END LICENSE-----
Successfully imported license
Customer name:  SE - UK - EMEA
Expiration:  22-Dec-2018
US regulators enabled:  true
CA regulators enabled:  true
EU regulators enabled:  true
APAC regulators enabled:  true
Security module enabled:  true
Actions framework enabled:  true
Users:  Unlimited

-bash-4.2$
```

Run:
sudo license-import

Paste in the license from the file provided by your instructor. Don't forgot to include the top and bottom dashed lines.

Press Enter

Create first Administrator User

Not Applicable on Remote Saas ***emea-training*** Lab
you already have been invited as Masteradmin to the Saas location

Run the resutil newuser command to create a User account.
You can use the commands text file in the Bootcamp folder so you do not have to write the commands out.

Your first name Your surname

```
-bash-4.2$ sudo resutil newuser -createorg -email YourEmail@Company.com -first "FirstName" -last "LastName" -org "YourOrg"
```

This should be an actual email address, you will use it to log into the platform.

This can be the name of your company, or you can set it to IBM if you like.

Example (Please do not use the values from the example) :

```
-bash-4.2$ sudo resutil newuser -createorg -email jessica.c@uk.ibm.com -first "Jessica" -last "Cholerton" -org "IBM"
Enter the password for the user:
Confirm the password for the user:
Creating a new user Jessica Cholerton <jessica.c@uk.ibm.com>
Creating a new organization IBM
Adding the user Jessica Cholerton <jessica.c@uk.ibm.com> to the organization IBM
Assigning the following roles to user jessica.c@uk.ibm.com: Master Administrator
-bash-4.2$
```

Do not use " " unless you are using more than one word and using a space to separate them.

Setup Email Technical Account used by QRadar SOAR

You will have been asked to have an email account ready to use for QRadar SOAR. You can use any SMTP account.

(We have a mail-in-a-box account you can use if required.)

Run:

sudo resutil smtpedit -email resilientaccount@gmail.com -name "Resilient Platform" -host smtp.gmail.com -user resilientaccount@gmail.com -port 587

y@mail.poc.resilientsystems.com

Technical Account email address

Example (Please do not use the values from the example) :

```
-bash-4.2$ sudo resutil smtpedit -email resilient.jess@gmail.com -name "Resilient Platform" -host smtp.gmail.com -user resilient.jess@gmail.com -port 587
Enter the password for the user:
Confirm the password for the user:
Successfully edited the SMTP configuration
SMTP Host: smtp.gmail.com
SMTP Port: 587
SMTP User: resilient.jess@gmail.com
SMTP Password: hidden
SMTP From Email: resilient.jess@gmail.com
SMTP From Name: Resilient Platform
SMTP StartTLS Enabled: true
SMTP Whitelist Hostnames:
-bash-4.2$
```

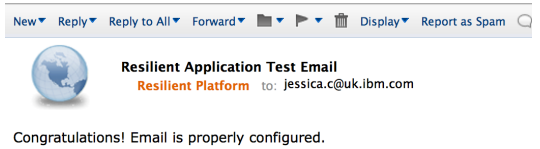
This password needs to be the password you log in to the email with

Not Applicable on Saas **emea-training** Lab you will receive emails from donotreply@mail.poc.resilientsystems.com

Now we need to test that it is working.

Use any email address that you can check, e.g. your work email

```
-bash-4.2$ sudo resutil smtptest -email jessica.c@uk.ibm.com
Successfully sent the test email to jessica.c@uk.ibm.com
-bash-4.2$
```




```
sudo resutil smtptest -email <your_work_email>
```

Setup Email Technical Account Help

Not Applicable on Remote Saas ***emea-training*** Lab

- **If you are using gmail, you will need to enable low security for apps, otherwise the email will not work, access the settings using this link:**

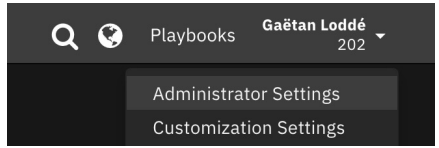
  (`https://myaccount.google.com/lesssecureapps`)

- If you get the following error message, it means that you haven't put your email address password in correctly. It needs to be the password that you log into your emails with.

```
-bash-4.2$ sudo resutil smtptest -email jessica.c@uk.ibm.com
[sudo] password for resadmin:
An error occurred while running the command line utility: Sending the email to the following server failed : smtp.gmail.com:25
Sending the email to the following server failed : smtp.gmail.com:25
  534-5.7.9 Please log in with your web browser and then try again. Learn more at
534 5.7.9  https://support.google.com/mail/?p=WebLoginRequired i66sm22589636wmd.0 - gsmtp
```

Add more users and team

To reach Administrator Settings click on your user name at the top right corner of the screen



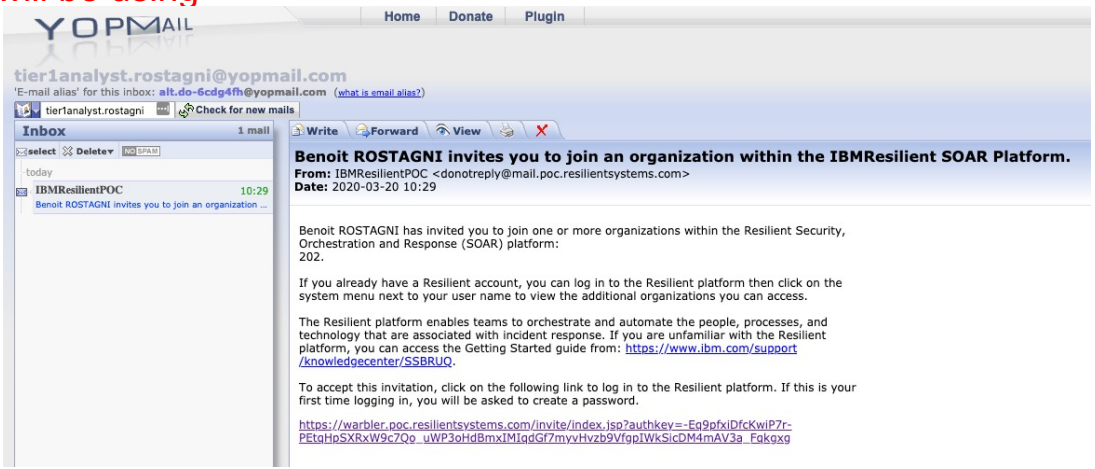
Go to Administrator Settings > Users > Invite Users > and invite the following users by their emails (replace YOURNAME by your Real Name ! – You may use another free email platform)

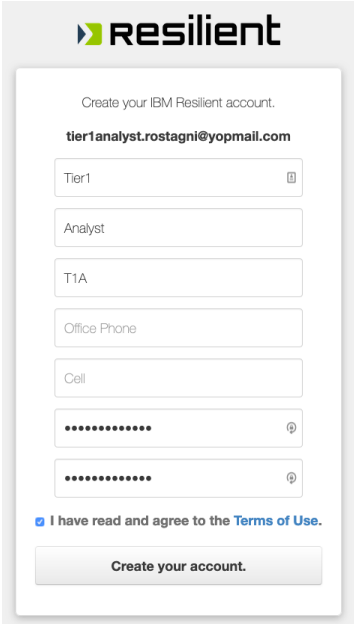
First Name	Last Name	Email	Permissions
Tier1	Analyst	Tier1Analyst.YOURNAME@yopmail.com	None
Tier2	Analyst	Tier2Analyst.YOURNAME@yopmail.com	None
Legal	Counsel	LegalCounsel.YOURNAME@yopmail.com	None

Go to https://yopmail.com and validate each user

Note: If Yopmail doesn't work, you can use any temporary mailbox service

Just be sure to use the mail generated by the temporary mailbox service you will be using



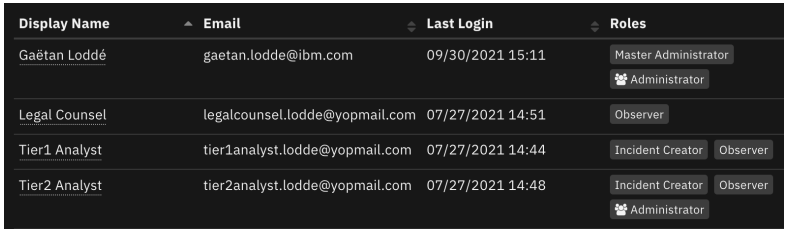


Create groups and add users

To access the User Settings in platform:

Click on Username (upper right)→ Administrator Settings→ Users

Click on each of the 3 Users you just created and edit permissions to match the roles in the following table



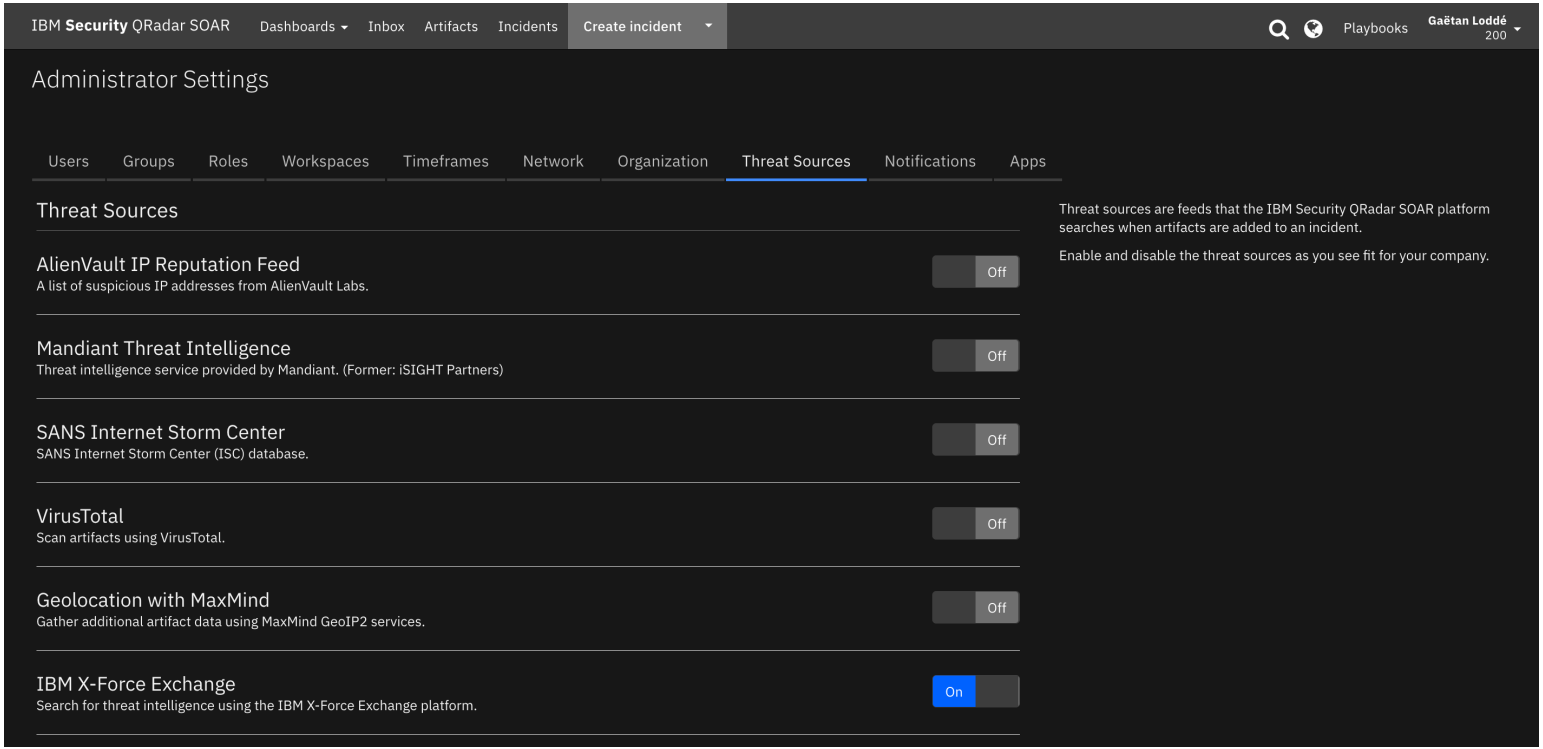
Display Name	Email	Last Login	Roles
Gaëtan Loddé	gaetan.lodde@ibm.com	09/30/2021 15:11	Master Administrator, Administrator
Legal Counsel	legalcounsel.lodde@yopmail.com	07/27/2021 14:51	Observer
Tier1 Analyst	tier1analyst.lodde@yopmail.com	07/27/2021 14:44	Incident Creator, Observer
Tier2 Analyst	tier2analyst.lodde@yopmail.com	07/27/2021 14:48	Incident Creator, Observer, Administrator

Now, click on the “Groups” tab and create the following Groups:

Group Name	Members	Permission
CSIRT	Your User Account, Tier2 Analyst	Administrator
Analysts	Tier1 Analyst, Tier2 Analyst	Incident Creator
Legal Team	Legal Counsel	Observer

Enable Threat Intelligence

Click on Username (upper right)→ Administrator Settings→ Threat Sources



You *should* register for API keys for free for X-Force.

- Get your IBM Xforce API Key @ https://exchange.xforce.ibmcloud.com/settings/api

QRADAR SOAR TRAINING

Lab 2

Goal:

- Understand the notion of incident and case
- Understand the notion of user, be part of a group, associated rights and roles, visibility of incident’s tasks
- Understand the notion of artifact, IOC, and enrichment by Threat Intelligence
- Understand the notion of teamwork on a task and group communication (war room) on this task

Lab 2: The Basics – Incident Creation & Editing

- Create a series of incidents using the New Incident creation wizard. This wizard will walk you through a new incident creation. Use the following table to populate your incidents:
  - NOTE: Artifacts can only be added to the incident after it has already been created.

Incident Type	Members	Owner	Artifacts
Malware	CSIRT	(will auto assign to you)	DNS: atmape.ru IP: 8.8.8.8
Phishing	CSIRT	(will auto assign to you)	URL: http://fatwallet.com/res?25242
Denial Of Service	CSIRT	(will auto assign to you)	DNS: poneytelecom.eu IP: 8.8.8.8

- Self Discovery: Have a play with:
  - Members Tab - add new members to an incident
  - Tasks Tab - Assign Tasks (individually and/or in bulk), note how users who are not members cannot be assigned tasks
  - Artifacts Tab - Look at the Graph view within Artifacts. Do this for each of the newly created incident, note how the different relationships are shown based on their common artifacts
  - Details Tab - Add additional Incident Types to an incident along with other field entries, observe new tasks being added to the incident
  - Notes Tab – Try sending a note to another user, using the @ functionality

Lab 2 Review: The Basics – Incident Creation & Editing Part

- The QRadar SOAR IRP, unlike a typical ITSM/Ticketing solution is purpose built for Incident Response.
- Task playbooks based on Incident Types ensure repetitive and consistent responses.
- Task individualization and ownership can be managed under the context of a single Incident, moreover the task ownership allows for accountability as well.
- Users must be added as members in order to see information. This maintains Op Sec sensitivity.
- Artifacts are automatically enriched via activated Threat Intelligence (where enabled and applicable).
- Artifact visualization allows for a quick understanding of the pervasiveness of the managed IOC, while the related incidents allow users to leverage lessons learned from previous similar incidents.
- Incident Details can be customized to include/exclude fields as necessary minimizing confusion of what information needs to be provided.
- Directing notes allows for better collaboration between multi-faceted teams that may be involved with an incident.

QRADAR SOAR TRAINING

Lab 3

Goal:

- Define threats linked to personal data depending on incidents
- Understand the different issues during a security breach depending on the geographic area

Lab 3: Working with the Privacy Module and Breach

- What QRadar SOAR provides is information surrounding Breach Notification and does not focus on privacy generalities.
- Go to the homepage by clicking on QRadar SOAR in the top task bar
  - Click on Resource Library on the right
  - NOTE: This is used from a purely research perspective.

You may get an error 'Unable to find Object ID 14'.
This happens when you select certain regulations for the first time.
Simply refresh the page and start again.

- Demoing Breach
  - Choose the Incident typed "Phishing"
  - Click on the Breach tab→ Edit
  - We will choose a scenario. For example, select:
    - Was personal information or personal data involved? : Yes
    - Data Encrypted: No
    - Data Format: Electronic
    - Data Types: Choose all the main headers (Contact Information, Personal Information, Credit Card, etc)
    - Applicable US Federal and/or Trade Org options
    - Pertinent localities/countries
    - Save
    - Note that one task is added to 'Assess the Risk'. You will now see the GDPR Assessment form in the Breach Tab. If you fill in that form, depending on your answers, you will see additional tasks being added.
- Self Discovery: Enable other countries for breach they may be applicable to you. Click on some of the links to understand what researchers have to work through in order to find relevant information.

Lab 3 Review: Working with the Privacy Module and Breach

- What QRadar SOAR provides is information surrounding Breach Notification and does not focus on privacy generalities.
- The Privacy Library allows for research to be conducted quickly and effectively. QRadar SOAR provides an overview of the topic pertinent to the Legislative and/or Regulatory statutes available globally. While also providing links to the source of the information should additional research be warranted.
- When a Breach happens: A quicker understanding of notification obligations can be obtained simply by identifying the nature of the breach, data types, and applicable legislative territory and/or trade organization during an ongoing incident.

- Summary:
  - Who you need to notify – Which Regulator/Authority
  - What you need to send to notify – What information is required, some notification templates
  - How you need to notify - Provides email/physical addresses of who to send the notification to
  - Why you need to notify – The Regulation/Legislation behind the need to notify

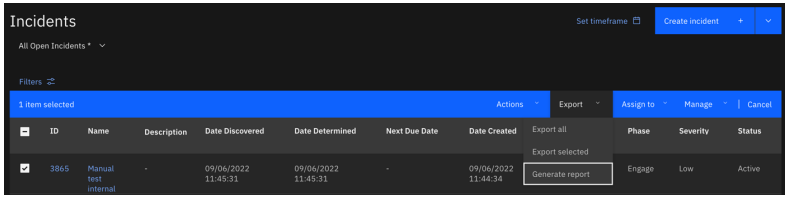
QRADAR SOAR TRAINING

Lab 4

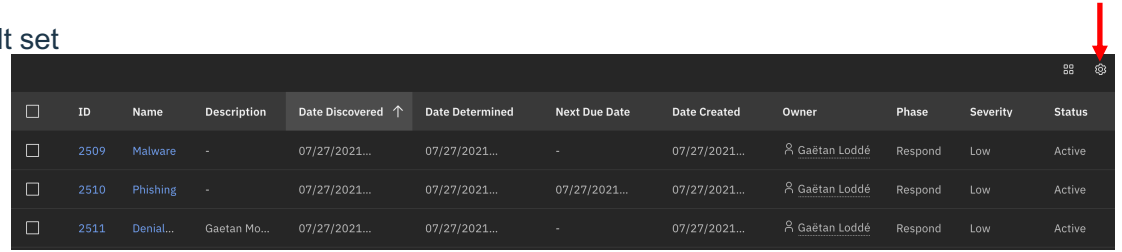
Goal:

- Understand the notion of report on an incident containing different targets
- Understand the notion of dashboard, KPI, and data segregation from users accessing to those dashboards

Lab 4: Reports & Dashboards



- Incident specific reporting:
  - Select an incident with the checkbox
  - Select the Export option
  - Click on Generate Report
  - Select any of the templates provided or “Customize”
    - Explore the output provided by each selection
    - Hint: Deselecting--> Reselecting puts the item back to the top. This can be rearranged via drag/drop
- List Incidents:
  - Click on Incidents (Grey Bar)
  - Select Fields to display as columns in the result set
  - Rearrange the order of the columns
  - Add filters based on field data (Click Filters)
  - Select All or a sub-set of Incidents
  - Click on Export to export or print



- Analytics Dashboard
  - Click Dashboard→ Analytics Dashboard
  - + Add Widget
    - Add widgets Open Incident by Type / Owner
    - Name and Save
  - Drag/Drop on screen and Save As
- Self Discovery: Create your own reports. Test the two different export options of List Incidents.

Lab 4 Review: Reports & Dashboards

- Reporting is simple

- Options:
  - Incident Specific Reporting (via Incident case – right side)
  - Reporting in Mass (via List Incidents)
  - Analytics Dashboards (via Dashboards)

- Exporting and/or Printing

QRADAR SOAR TRAINING

Lab 5

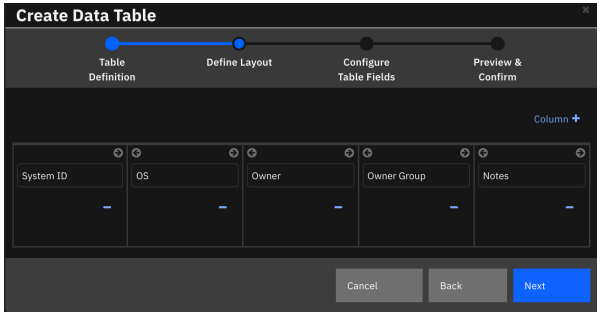
Goal:

- Know how to enrich the interface to gather interesting data
- Master the different data gathering fields

Lab 5: Field & Tab Customization – Part 1/3

- Click on your username (upper right)→ Customization Settings
- Click on Incident Tabs→ Details
- Add new fields:
  - Label: “Executive Involved”, Type: Boolean
  - Label: “Systems Infected”, Type: Boolean
  - Explore other field creation

Be careful to make sure you copy the table values exactly. If you miss capital letters or add accidental spaces, this will cause problems in later labs.



- Create a Data Table using the following matrix for columns and data types per column
  - Data Table Name: ”Infected Systems”

Column Name (API Access Name)	System ID (system_id)	OS (os)	Owner (owner)	Owner Group (owner_group)	Notes (notes)	Remediation State (remediation_state)
Field Type	Text	Select	Text	Select	Text Area	Select
Values		Android		Development		Pending
		Linux		Exec		Confirmed
		Mac		Finance		Escalated
		IOS		GA		Remediated
		Windows		HR		
				Marketing		
				Sales		

- Self Discovery: Create new fields and Data Tables of your own

Lab 5: Field & Tab Customization (Placement) – Part 2/3

- While still in the Details tab:
  NOTE: Save each time you place fields where you want them
  - Drag Drop the “Executive Involved” field from the list of fields and place just below Severity
  - Drag Drop the “Systems Infected” field just below “Executive Involved”
  - Drag Drop the “Infected Systems” data table just above the ”Basic Details” header.
    - NOTE: You can see how easy it is to keep adding fields and place them onto the form. But what about fields that are not immediately relevant?

  - On the right-hand side of the screen, scroll down to Blocks
  - Drag a Section over to below the recently placed “Infected Systems” data table
  - Drag the existing “Infected Systems” data table on form into the new section
  - Click on the section spanner icon and add a condition to the section that allows it to appear only when the “Systems Infected” field is “Yes”
  - Save
- Explore your incidents, and edit within the Details Tab by toggling Yes/No for “Systems Infected” What happens a bit below this?
- Self Discovery: Try adding fields to another tab. Are they useful there?

Lab 5 - cont: Field & Tab Customization (Custom Tabs) – Part 3/3

- Navigate back to Incident Tabs→ Manage Tabs:
  - Click the “+” at the end of the listed tabs.
  - Tab Text = “Infected Systems”
    - Add a condition to the section that allows it to appear only when the “Systems Infected” field is “Yes”.
- Now navigate to Incident Tabs→ Infected Systems.
  - Drag Drop the “Infected Systems” data table into the tab space.
  - Save
- Explore your incidents, and edit within the Details Tab by toggling Yes/No for “Systems Infected” What happens to available tabs?
- Is this a better option than adding to the Details Tab? Remove the one of the options.
- Self Discovery:
  - Add the “Executive Involved” field to the Summary Tab.
  - Have a play with fields and Tab/Section conditions. You’ll find that multiple conditions imply an AND operator.

Lab 5 Review: Field & Tab Customization

- QRadar SOAR allows users to configure the system to include all their field entries
- Fields can be of various types:
  - Text, Text Area, Date, Date/Time, Number, Boolean, Select, Multi-Select
- The API Access Names are automatically generated, but can be edited
- Fields can be mandatory, mandatory on close, or optional (default)
- Users can also create mini spreadsheets/data tables within the solution
  - This is especially useful for incidents that may involve several systems
- Placement of the fields is as easy as drag drop once created
- Fields in the IRP can appear/disappear based on need
  - This is based on the condition(s) provided to sections or tabs

QRADAR SOAR TRAINING

Lab 6

Goal:

- Understand the notion of phases in the solving steps of an incident
- Understand the customization of descriptive to do tasks elements, elements to produce to accomplish it, and expected elements to produce a report

Lab 6: Incident Types, Phases, and Tasks

- Navigate to Incident Types
  - Add a Type: ”Manual Incident”
  - Parent: [Blank], Hidden: [No]
    - NOTE: Adding a parent allows for inheritance of the parent incident’s tasks
- Navigate to Phases & Tasks
  - Create Phase:
    - Phase Name: Final Phase
  - Create Task:
    - Task Name: Legal Standing
    - Phase: Final Phase
    - Instructions: Confirm that all legal obligations have been fulfilled.
    - Incident Fields, drag and drop from the list:
      - Criminal Activity
      - Data Encrypted
- Self discovery:
  - Create additional Incident Types and Tasks
  - Drag Drop tasks within a phase to reset the order

Lab 6 Review: Incident Types, Phases, and Tasks

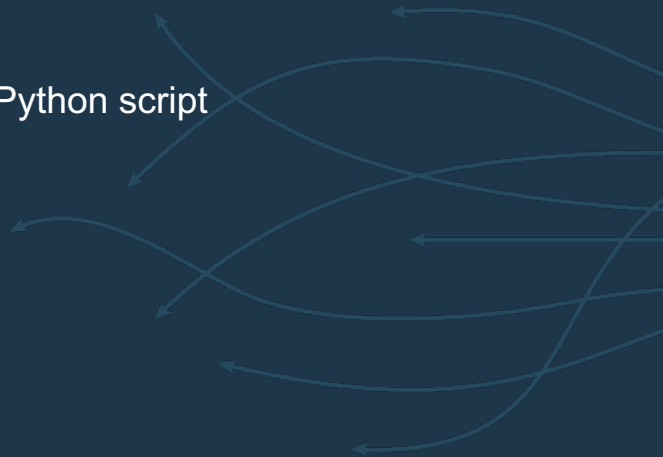
- Creating Incident Types is quite easy. They are simply identifiers.
- Phases are also easy. They are just gatherings of tasks.
- Task creation is also easy. This flexibility allows for a gathering of tasks, much like you would gather ingredients for a recipe.
  - You'll also find that these tasks can be reused. That's coming up.
- The order of tasks within a phase dictates the order they will appear in an active incident
- Hyperlinks within task instructions are useful for providing quick access to external resources
- Why hide an incident type ?
  - Its ruleset is still under construction.
  - Hiding a parent forces the initiator to be more specific, and not generalize to a parent that may be vague.

QRADAR SOAR TRAINING

Lab 7

Goal:

- Create/Use a Python script



Lab 7: Scripts – Part 1/2 , Part 2 is the lab 8

- We’re not here to learn Python scripting!

- However… Python scripts can be used for advanced function within QRadar SOAR
- For now, simply copy the contents of the “CBHuntIOCs.txt” file into a new script
- Here are the parameters for the script to be created:
  - Name: ”CB Hunt IOCs”
  - Object Type: Artifact
  - Script: [paste the contents into the script space]
    - It should start with: “incident = artifact.getParentObject()”
    - And end on row 42 with: “row[‘remediation_state’] = ‘Pending’
- NOTES: When using a script to populate fields, be sure that the entry corresponds with the available options if the Field/DT Column is of type: Select, Multi-Select, Boolean, etc.

QRADAR SOAR TRAINING

Lab 8

Goal:

- Understand the notion of Playbook and organize a set of rules to run this playbook
- Understand the notion of manual or automatic actions

Lab 8: And now… Rules – Part 1/3

- The rules allow for massive flexibility when dealing with incidents, and the creation of dynamic playbooks.

- Go to the Rules tab in Customization Settings

- Let's create our first rule. Use the following matrix to create it.

Rule Type	Display Name	Object Type	Condition(s)	Activities
Automatic	Manual Incident	Incident	Incident Type: Is equal to: Manual Incident	Add Tasks: [Select 5-10 tasks]

Lab 8: Rules – Part 2/3

- Create a series of rules using the settings below: Use the AND operator for the last.

Rule Type	Display Name	Object Type	Condition(s)	Activities
Menu Item	CB Hunt IOC	Artifact	Value: Has a value	Run Script: CB Hunt IOCs
Automatic	Set Executive Involved	Data Table Infected Systems	Owner Group: Is equal to: Exec	Set Field: “Incident: Executive Involved”: Yes
Automatic	Executive Involved	Incident	Executive Involved: Is equal to: Yes	Set Field: “Incident: Severity”: High
				Set Field: “Incident-Owner”: Tier 2 Analysts
				Set Field: “Incident: Members”: CSIRT, Legal Team, Tier2 Analysts
				Add Tasks: Determine if illegal activity is involved
Automatic	Assign to Legal	Task	Name: Contains: Legal	Set Field: “Task-Owner”: Legal Counsel
			Executive Involved: Is equal to: Yes	

Lab 8: Rules Demonstration – Part 3/3

- Take note of the following fields:
  - Severity
  - Executive Involved
  - Incident Owner
  - Members
  - Infected Systems (Data Table)
- Take note of your tasks.
- In an incident that has an artifact, click on the ellipse […] next to an artifact.
  - Mimic the “demonstration” of running an integrated Carbon Black scan for the identified IOC.
- What happened to the fields & tasks?

Lab 8 Review: Rules

- Rules allow for dynamic playbooks to be created.
- This allows for not only playbook creation based on incident type(s), but also the dynamic capability to adapt to any incident based on business rules and new data.
- The drivers of these rules can be multi-conditional.
- Automatic rules are automatically enacted given the right conditions.
- Automatic rules are processed top down. However unlike a firewall rule that may stop at the first hit, all rules are processed. As such, any clashes are subject to the effects of the last applicable rule run.

QRADAR SOAR TRAINING

Lab 9: Additional script and rule

Goal:

- Understand the notion of “click” simplification asked by automatisms
- Understand the notion of Milestone

Lab 9: Additional script and rule work 1 of 2

Please, create script & rule that will auto assign task to user connected when the task owner is empty, and the task status is changed to close.

Hint: the python code to assign user connected to the task ownership is below.

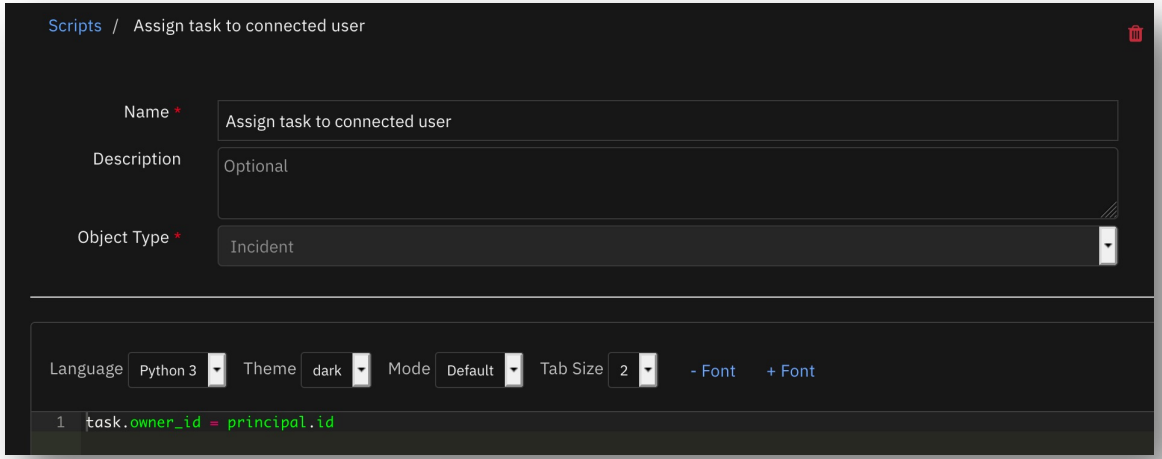
task.owner_id = principal.id

Test it with a user owner or member of an incident.

Test it also with a user that has an Administration permission, but who is NOT owner or member of the incident.

What happen ?

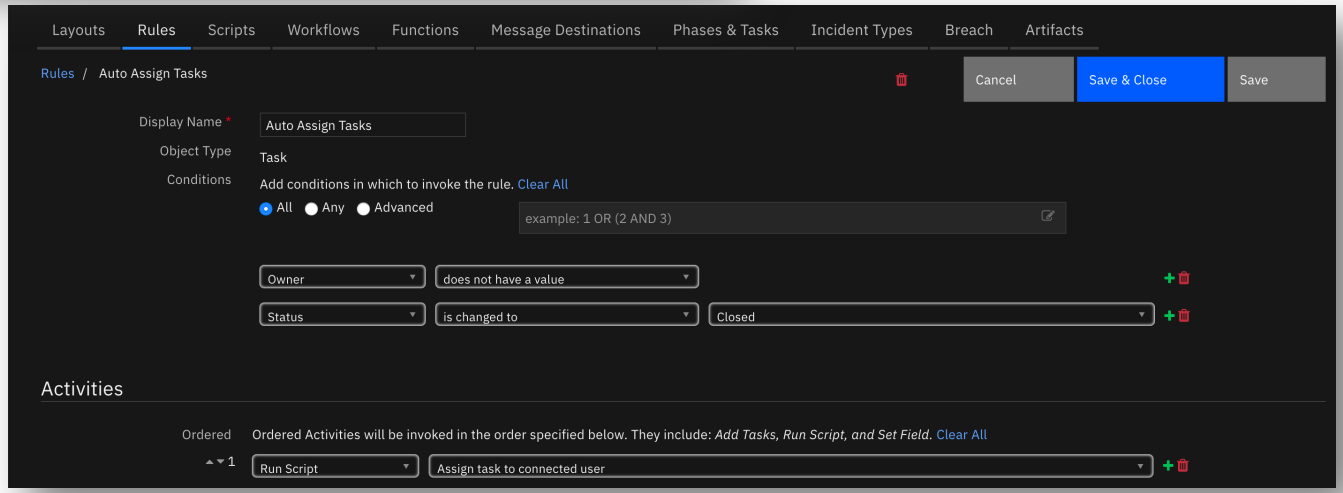
Auto-assign task to user connected



SOLUTION Slide !!!

Please do it by yourself before looking at the solution

Note : There is a error in this slide, for you to find !!!



Yes but, what if it does not work? (Auto-assign task to user connected)

When the user is not listed in the incident membership, The script is a little bit more complex :

```
# Check if the connected user that maybe viewer or full admin  is  a member of this incident
inside = False
log.debug("Inside: {}".format(inside))
log.info("Principal Name: {}".format(principal.name))
log.info("Owner ID: {}".format(incident.owner_id))
log.info("Members: {}".format(incident.members))

# Check Incident Owner Level
if incident.owner_id == principal.name:
  inside = True
# Check Incident Owner if Group
elif groups.findByName(incident.owner_id):
  log.debug("In incident owner group")
  inside = True
else:
  log.debug("Not in incident owner group")

# Check Members Level
for member in incident.members:
  log.debug("Looking at Member: {}".format(member))
  if member == principal.name:
    inside = True
  elif groups.findByName(member):
    log.debug("Inside group Member: {}".format(member))
    inside = True

# if 1, the connected user is a member
log.debug("Last Inside Value (0= False/1= True): {}".format(inside))

# Add current connected user as a member
if principal.type == "apikey":
  donothing = "If you have no Orchestration user or add it below as member"
  incident.members = list(incident.members) + ["resilient.action@rostagni.com"]
  task.owner_id = "resilient.action@rostagni.com"
else:
  if not inside:
    incident.members = list(incident.members) + [principal.id]
# Add current connected user as Task Owner, as this full script was made to add connected user as task owner.
  task.owner_id = principal.id
```

SOLUTION Slide !!!

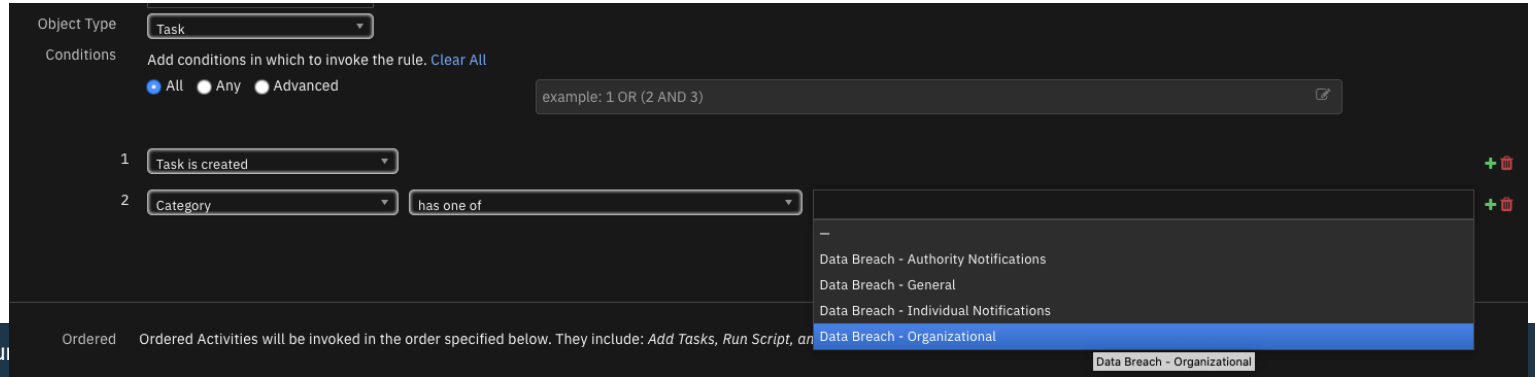
Beware of Cut & Paste quote errors or identation errors from PDF, use the txt file on Lab 9 folder

Lab 9: Additional script and rule work 2 of 3

Please, create scripts & rules that will add a milestone when Legal Team is added to the incident, and will assign all privacy tasks to a legal user of this team.

Hint: the python code to ADD a user to the members list is below.

```
from datetime import datetime
if "Legal Team" not in incident.members:
  incident.members = list(incident.members) + ["Legal Team"]
  incident.addMilestone('Add Legal Group', 'Legal Group is added to incident as Privacy Analysis is needed', datetime.now())
```



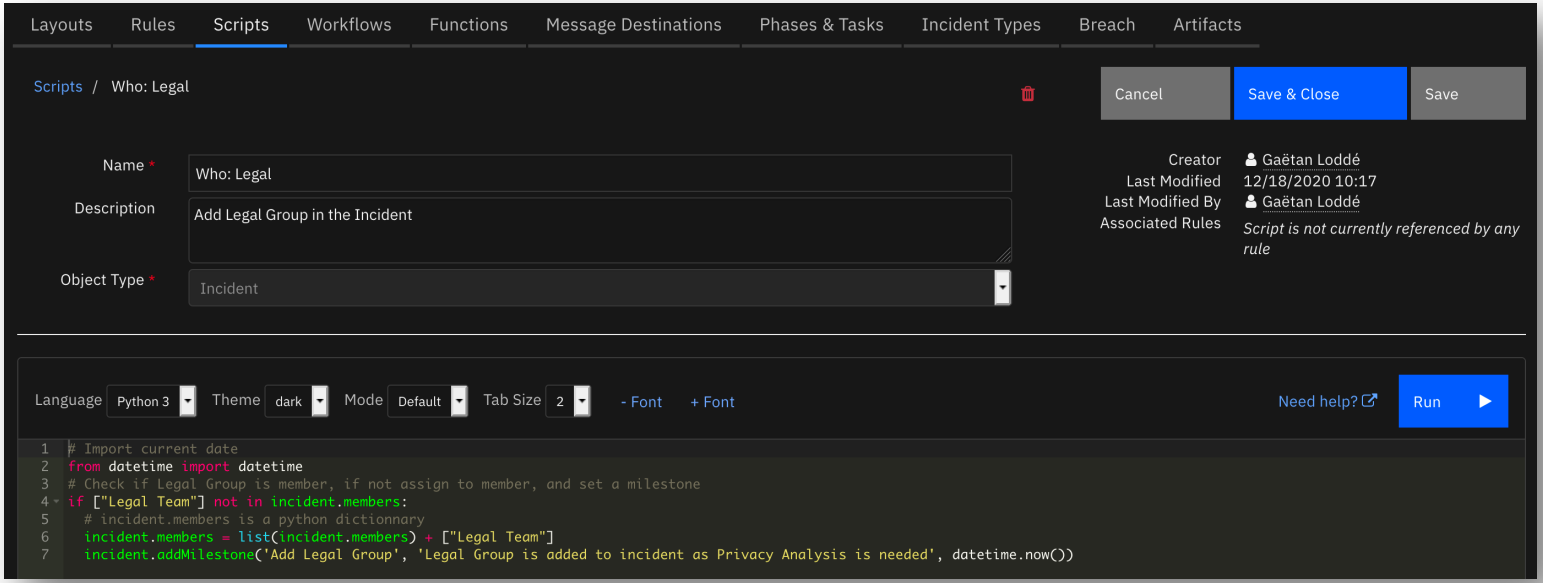
Milestone when a task is assigned to Legal

1/2 Creation of the script

SOLUTION Slide !!!

Please do it by yourself before looking at the solution

```
from datetime import datetime
if "Legal Team" not in incident.members:
    incident.members = list(incident.members) + ["Legal Team"]
    incident.addMilestone('Add Legal Group', 'Legal Group is added to incident as Privacy Analysis is needed', datetime.now())
```

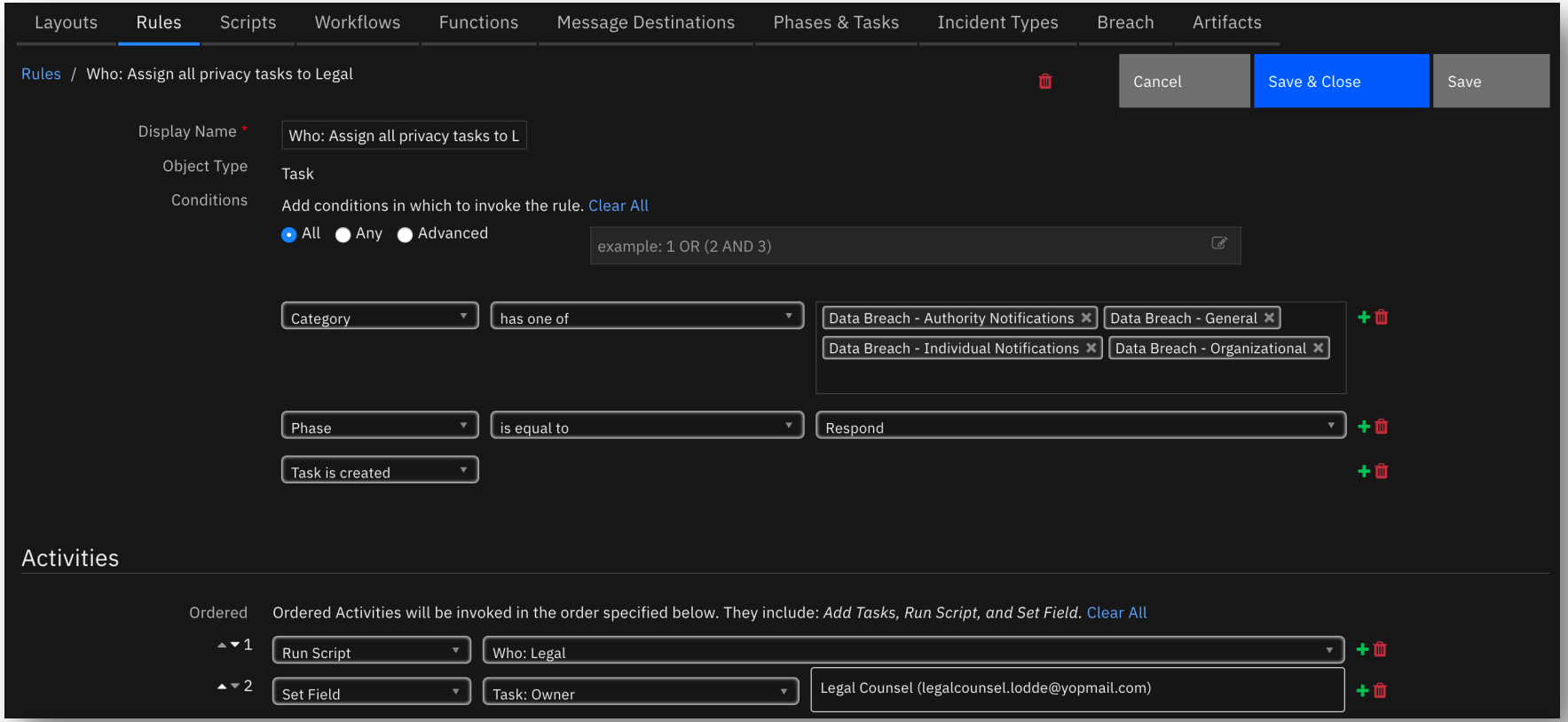


Milestone when a task is assigned to Legal

2/2 Creation of the rule (automatic)

SOLUTION Slide !!!

Please do it by yourself before looking at the solution



QRADAR SOAR TRAINING

Lab 10: Inbound Email

Goal:

- Create a mail analyzer on an automatic mail collect

Setup Email Technical Account used by QRadar SOAR

You will have been asked to have an email account ready to use for QRadar SOAR. You can use any SMTP account.

(We have a mail-in-a-box account you can use if required.)

Run:

sudo resutil smtpedit -email resilientaccount@gmail.com -name "Resilient Platform" -host smtp.gmail.com -user resilientaccount@gmail.com -port 587

y@mail.poc.resilientsystems.com

Technical Account email address

Example (Please do not use the values from the example) :

```
-bash-4.2$ sudo resutil smtpedit -email resilient.jess@gmail.com -name "Resilient Platform" -host smtp.gmail.com -user resilient.jess@gmail.com -port 587
Enter the password for the user:
Confirm the password for the user:
Successfully edited the SMTP configuration
SMTP Host: smtp.gmail.com
SMTP Port: 587
SMTP User: resilient.jess@gmail.com
SMTP Password: hidden
SMTP From Email: resilient.jess@gmail.com
SMTP From Name: Resilient Platform
SMTP StartTLS Enabled: true
SMTP Whitelist Hostnames:
-bash-4.2$
```

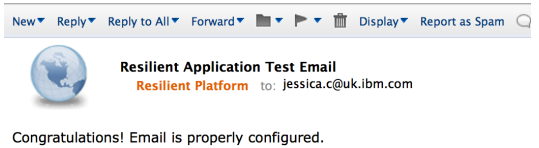
This password needs to be the password you log in to the email with

Not Applicable on Saas Lab you will receive emails from donotreply@mail.poc.resilientsystems.com

Now we need to test that it is working.

Use any email address that you can check, e.g. your work email

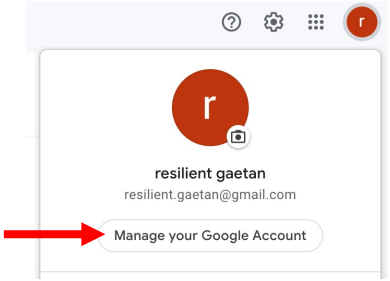
```
-bash-4.2$ sudo resutil smtptest -email jessica.c@uk.ibm.com
Successfully sent the test email to jessica.c@uk.ibm.com
-bash-4.2$
```



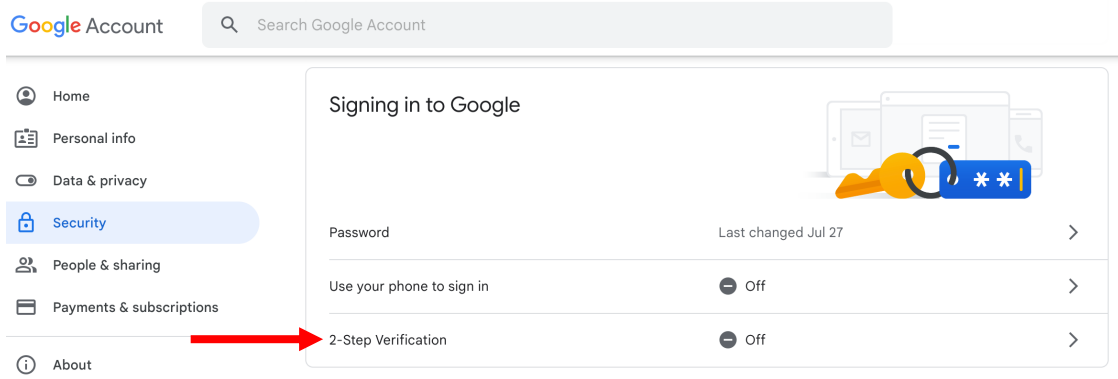

```
sudo resutil smtptest -email <your_work_email>
```

Setup Email Technical Account used by QRadar SOAR

- **Create GMAIL account named resilient.YOURNAME@gmail.com**
  Do NOT USE your regular Gmail account
  QRadar SOAR will monitor all email send to this box to analyze them.
- **For gmail account, you will need to create an app password to connect your mail account to the SOAR platform, otherwise the email will not work**
- **To do so, first go to the "Manage your Google Account" setting from your google profile:**

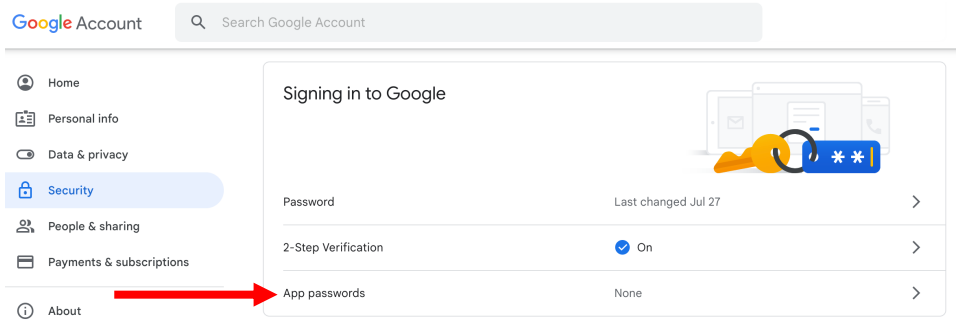


- Then select « Security » and enable the 2-step Verification with your phone number and the option « App passwords » will appear when going back to the 2-step Verification option at the bottom of the web page



Setup Email Technical Account used by QRadar SOAR

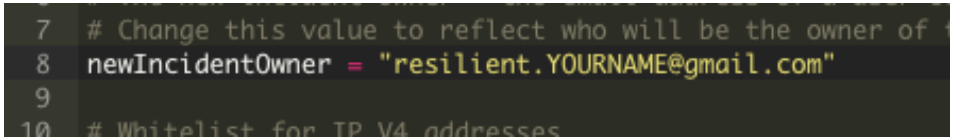
- After the 2-Step Verification is enabled the « App passwords » option will be enabled in your security tab



- Click on the “App passwords” option and in the “select app” field select “other (custom name)” and fill with “Gmail on SOAR” and click on “Generate”
- An app password will be created, copy and paste it in a text file to use it later
- In the gmail settings, go to the POP/IMAP tab and enable IMAP access.
- If you get the following error message, it means that you haven’t put your email address password in correctly. It needs to be the password that you log into your emails with.
- When the Gmail account is ready, invite **resilient.YOURNAME@gmail.com** into QRadar SOAR as Master Admin, go to GMAIL and validate the invitation.

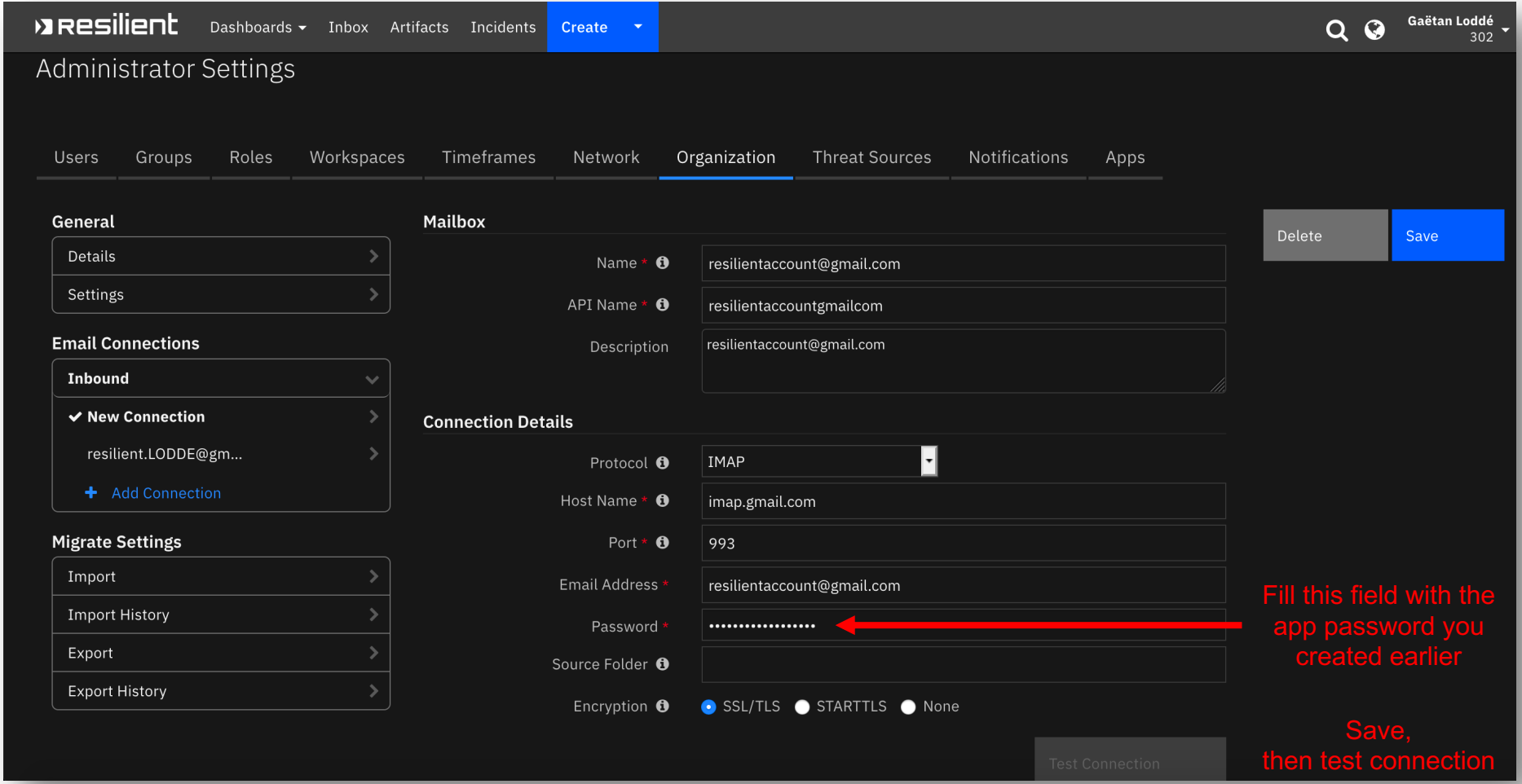
Lab 10: Inbound Email

- Connect your email server and technical account that will receive all emails, to create automatically new incidents.
- Add a rule that will run the script « Sample script: process inbound email (v49) » when a new email is received
- Change in the « Sample script: process inbound email » the value of the owner of the incident when the incident is created (line 8)

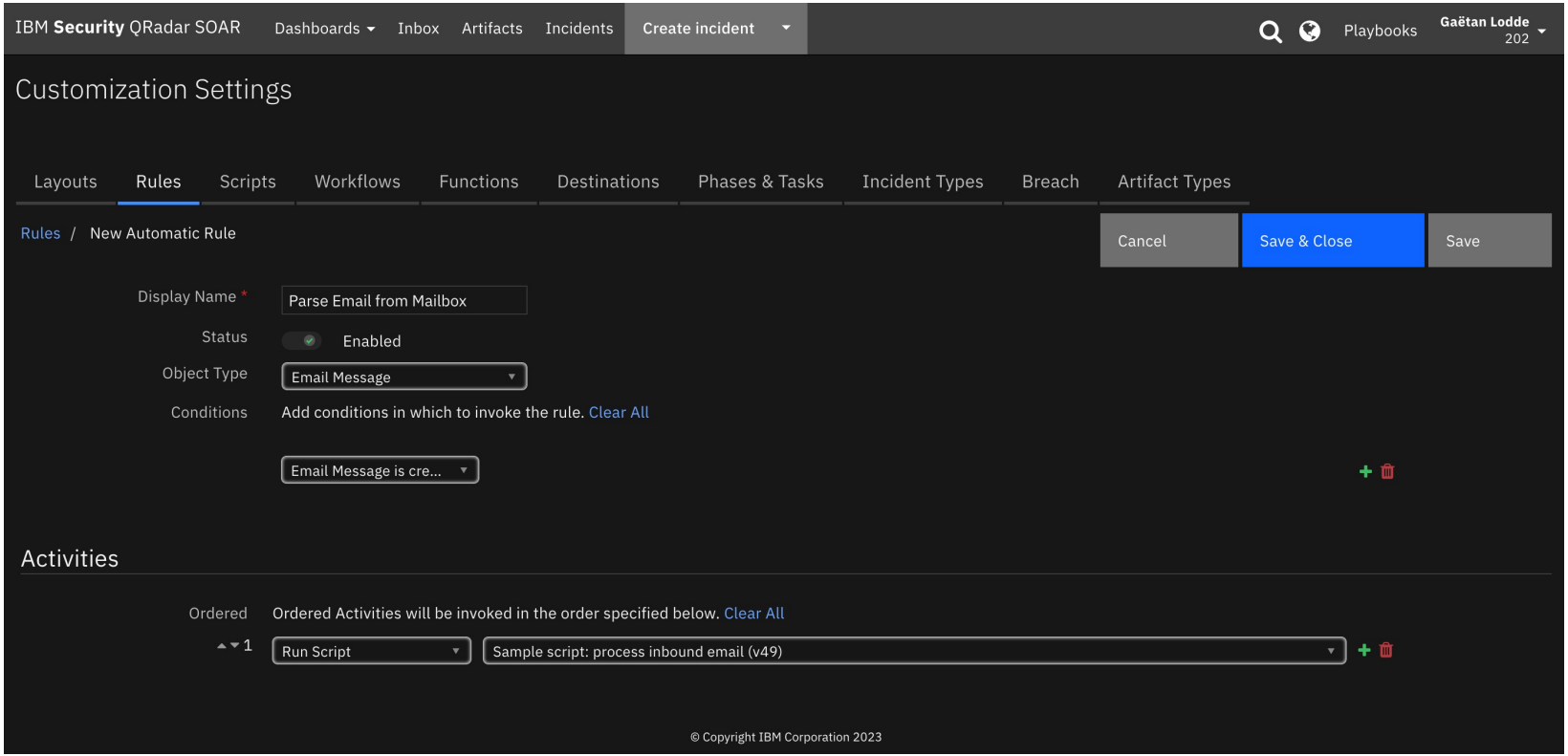


- Note: the script can be modified to add new pattern matching, or other fields assignments please look at the script documentation and the latest version of this script at App Exchange “**Generic Email Parsing Script”** app: https://exchange.xforce.ibmcloud.com/hub/extension/4ba70106b6f2dfa77cb1e3c921db7ff5
- You can also look at the "**QRadar SOAR IRP Playbook Designer Guide.pdf**” for all objects to use in a script.

Define the connection to the email server

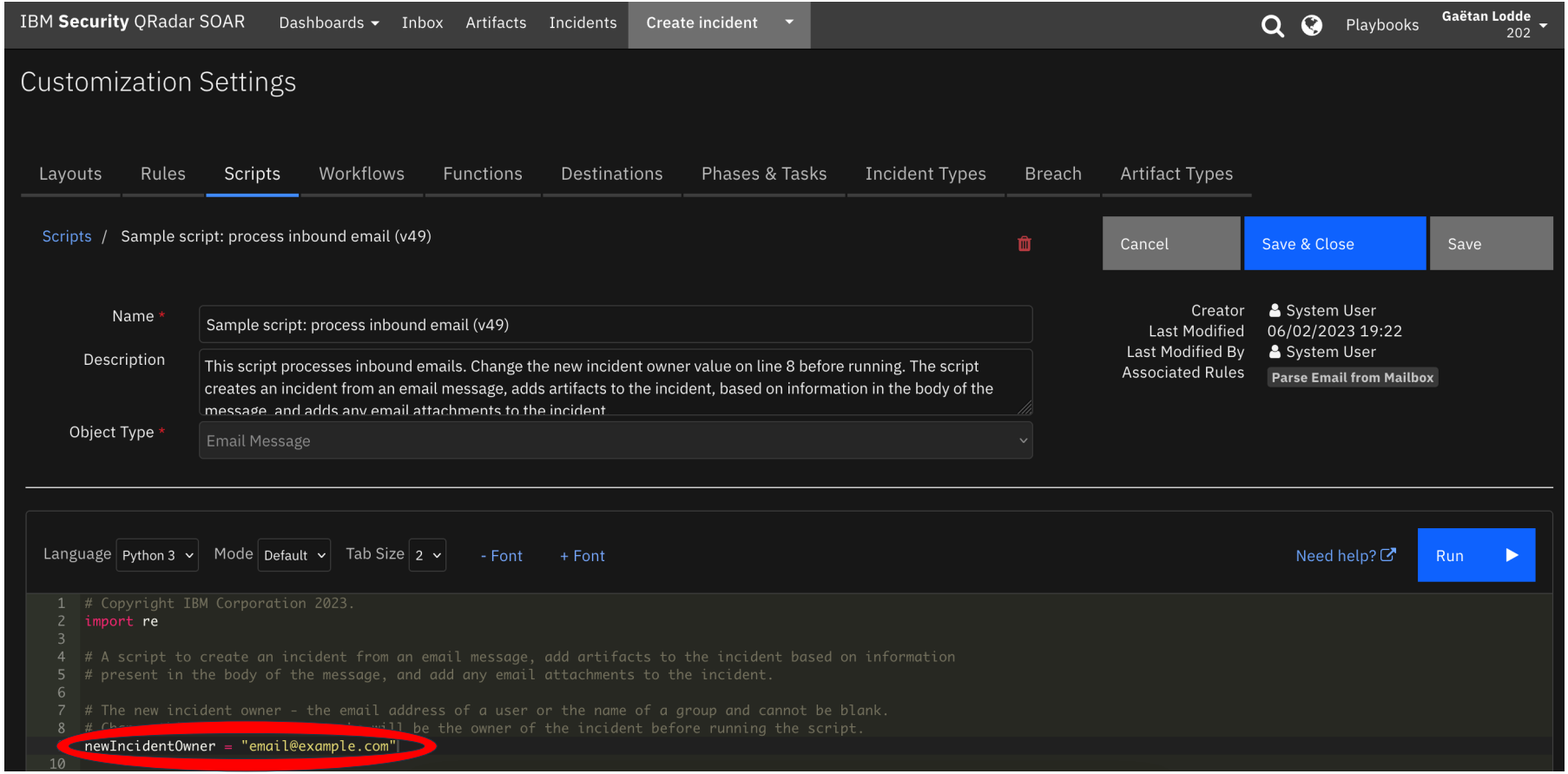


Create the rule



The Condition has to be: **Email Message is created**

Change in the « Sample script: process inbound email (v49)» the value of the owner of the incident when the incident is created (line 9)



Test the automatic creation of the incident by sending an email to the technical account (resilientaccount@gmail.com)

QRADAR SOAR TRAINING

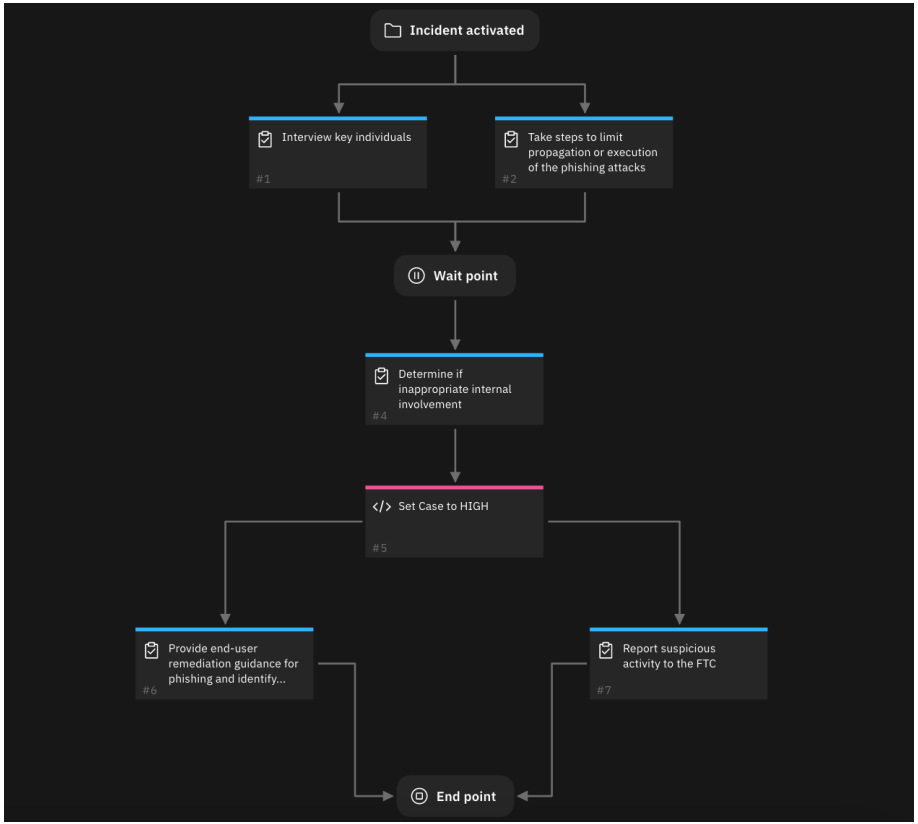
Lab 11: Playbook

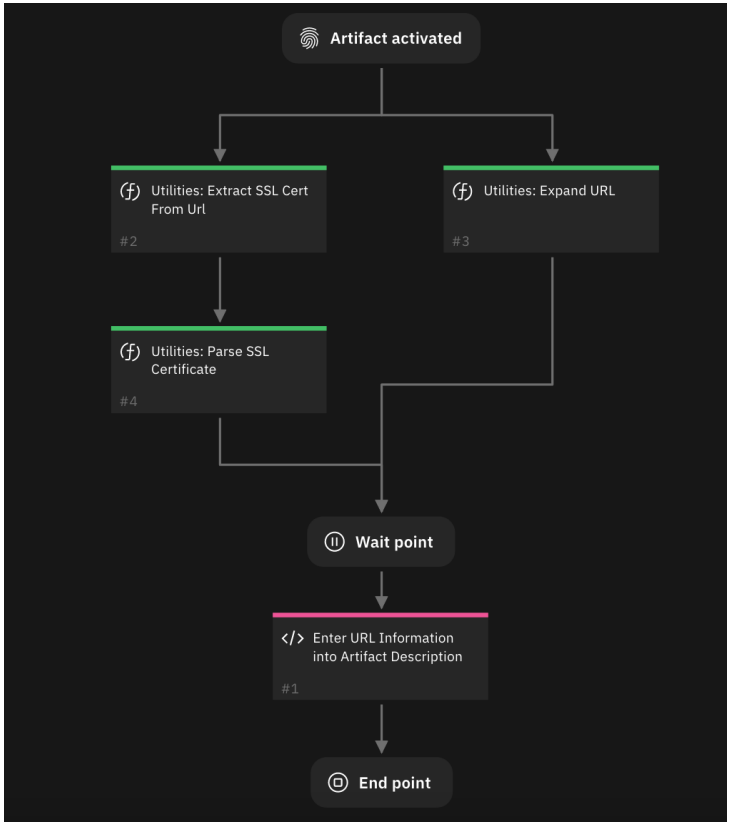
Goal:

- Know how to design a workflow to answer an issue
- Design Thinking, virtual team whiteboarding
- BPMN practice, test and validation

Lab 11: Playbook Training

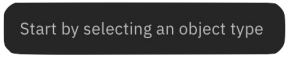
- Example of Final Work:





Workflow Training

Use the BPMN symbol and select and click « add to canvas » to build your playbook :

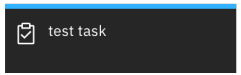


- **Start Event**. Denotes the start point for the Playbook. It define the type of object on which the playbook will work and if it is triggered manually or automatically.

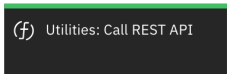
- **End Point**. Denotes an end point. If you have multiple End Events, all active paths are traversed even when one path reaches the end point.

- **Connector**. Logically connects components within the workflow to each other. You can add conditions to the connector when it is the output of a condition point.

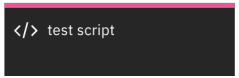
- **Condition Point**. <u>You assign conditions to the output paths (the Connector/arrow link to the next object)</u>. You can designate one output path as a “Else”, which is used when none of the conditional paths are true.



- **User Task**. Allows you to create a new task or select an existing task to be inserted in the corresponding incident’s task list. This is comparable to adding a task from a rule. **Script**. Allows you to select an existing script to invoke. The workflow object type affects which scripts are available to the workflow.



- **Function**. Allows you to select an existing function. It also allows you to create pre-process scripts for that function.



- **Script**. Allows you to create and add a new script or add an existing script in your playbook to perform action automatically.

- **Wait Point**. At this point of the playbook, it won’t progress until the previous actions are done.

Best Practice:
Always start with ONE **Start Event**
Always finish with ONE **End Event**

Lab Playbook: Ransomware Lab

Organize new tasks and fields with current phase to create a Ransomware workflow using the following possibilities, to be ordered:

1. Identify ransomware
2. Disconnect infected endpoint
3. Disconnect backup
4. Look ransomware IOC on other endpoints
5. Verify backup
6. Restore backup
7. Analyze encrypted data values
8. Decide to pay or not
9. Re-image endpoint
10. Patch SMB Vuln globally

BPMN Symbols

Person in the corner = Add Task
"When the workflow gets here, add this task"
Mnemonic: Tasks need people to do them

Gear in the corner = Append Message Destination
"When the workflow gets here, automatically carry out this action"
Mnemonic: Gears represent automation

X in a diamond = Exclusive Gate
Process Fork: "Only execute one of these paths"
Process Join: "Advance when prior path to this is executed"
Mnemonic: X for Xclusive

O in a diamond = Inclusive Gate
Process Fork: "Optionally execute whichever of these paths apply"
Process Join: "Advance when all applicable paths are executed"
Mnemonic: O is for Optional

Plus in a diamond = Parallel Gate
Process Fork: "Execute all of these things in parallel"
Process Join: "Advance when all connecting paths are executed"
Mnemonic: Plus for Parallel

Hints

- Design on a paper all process using boxes and the ID number of the task
- Add below the task, all fields that will be asked to the analyst
- When using a question (exclusive gate), write on each arrow line the condition to follow
- The next page is wrong ! Do not take it as the solution, but only as hint. They are process errors inside.

It is possible to use Mural to design the Playbook:

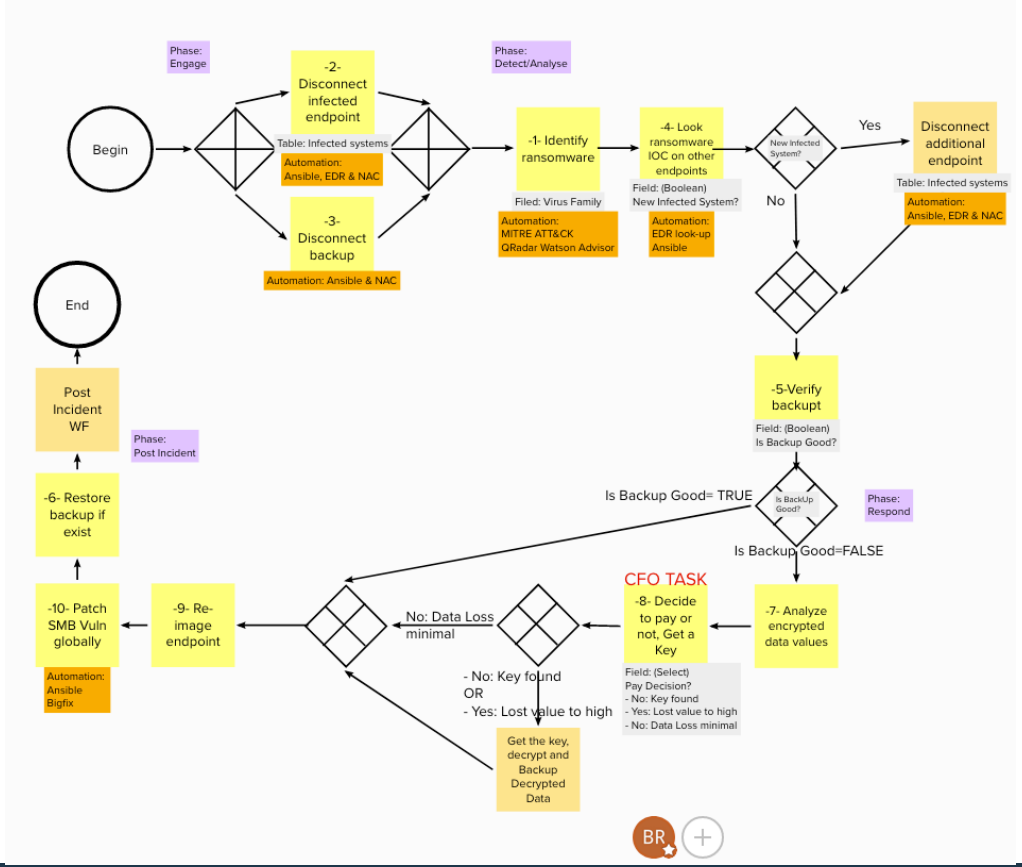
https://app.mural.co/

Mural Design

SOLUTION Slide !!!

Please do it by yourself before looking at the solution

HTTPS://APP.MURAL.CO/INVITATION/MURAL/DIGITALCOVERAGEANDCOMMERCE3006/1595540190673?SENDER=BENOITROSTAGNI4490&KEY=EA82155D-A6B8-48F0-8B0A-562860D312D4



QRADAR SOAR TRAINING

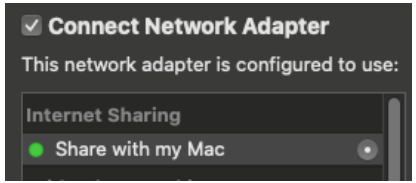
Lab 12: Install the Integration Server

Goal:

- Install and Initialize an integration Virtual Machine
- Understand the notion of Message Bus, and communication flows direction
- Understand the notion of API Key

Lab 12: Install the Integration Virtual Machine

- *Official documentation:* *https://www.ibm.com/docs/en/sqsp/44?topic=isg-introduction*
- Download the Virtual Image of the integration server from Box : https://ibm.box.com/v/TrainingAppHostOVA
- Install it in your Virtual Box or VMWare Fusion / Player software
- The machine should be put in a Shared Internet network with your system.



- You can connect to the console using **root** or **appadmin** users, with the unique password : **resilient**
- In the console run ifconfig to get the IP given by the DHCP:

```
Red Hat Enterprise Linux Server 7.7 (Maipo)
Kernel 3.10.0-1062.12.1.el7.x86_64 on an x86_64

resilient login: root
Password:
Last login: Thu Mar 19 15:18:55 on pts/1
[root@resilient ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.113.147  netmask 255.255.255.0  broadcast 192.168.113.255
```

- For the rest of the lab, we will use **appadmin / resilient** to connect in SSH using putty in windows
  Please DO NO USE ROOT or sudo su in ALL LABS, except when explicitly written

Lab 12: Integrations Setup

- We will use our own QRadar SOAR server but in production you should do this on a different server.
- First, we will use nano to simplify the edition of files. Of course, if you prefer vi, you can stay with vi!

```
# Get NANO on board
wget https://vault.centos.org/centos/8/BaseOS/x86_64/os/Packages/nano-2.9.8-1.el8.x86_64.rpm
sudo rpm -Uvh nano-2.9.8-1.el8.x86_64.rpm
nano .nanorc
include /usr/share/nano/python.nanorc
include /usr/share/nano/sh.nanorc
```

- We will use a specific CLI user, for the **integration server**: **integration**
  - sudo useradd integration

  You should be root for the next commands :
  - sudo su
  - echo "integration ALL=(ALL) ALL" >> /etc/sudoers
  - passwd integration
  - su integration
  - cd

```
[root@apphost ~]# su integration
[integration@apphost root]$ cd
[integration@apphost ~]$ pwd
/home/integration
[integration@apphost ~]$
```

Setting up QRadar SOAR Circuits

Beware not to mess up between both 2.7 & 3.6 environments in using other users.

Install the Python components:

- The utility functions package contains Python components that will be called by the QRadar SOAR platform to execute the functions during your workflows. These components run in the 'resilient-circuits' integration framework.
- The package also includes QRadar SOAR customizations that will be imported into the platform later.
  Ensure that the environment is up to date:

```
sudo pip3 install --upgrade pip3
sudo pip3 install --upgrade setuptools
sudo pip3 install --upgrade resilient-circuits
sudo pip3 install --upgrade resilient-sdk
```

Note: If you encounter any error launching those commands try adding:
--ignore-installed
Note 2: pip3 upgrade might not be necessary anymore, if you have an error just ignore the update (first sudo line)

Note: For all other « pip » installation, we will stay « integration », **no sudo !**

```
[integration@apphost ~]$ resilient-circuits help
/usr/lib/python2.7/site-packages/urllib3/contrib/pyopenssl.py:51: CryptographyDeprecationWarning: Python 2 is no longer supported
by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography import x509
usage: resilient-circuits [-h] [-v]

                          {run,list,test,service,config,codegen,extract,customize,selftest,clone}
                          ...
resilient-circuits: error: argument cmd: invalid choice: 'help' (choose from 'run', 'list', 'test', 'service', 'config', 'codegen'
, 'extract', 'customize', 'selftest', 'clone')
```

Configure QRadar SOAR Circuits

BE SURE TO USE integration USER HERE

QRadar SOAR Circuits configurations are maintained in the app.config file

Auto generate the **app.config file**:

```
$ su integration
$ cd /home/integration
$ /usr/local/bin/resilient-circuits config -c
```

This creates /home/integration/.resilient/app.config

Open this file in **NANO:**

```
$ nano /home/integration/.resilient/app.config
```

Change the various values with these configuration settings:

```
host=emea-training.poc.resilientsystems.com
port=443
org=YourOrg - like 202
componentsdir=/home/integration/.resilient/components
logdir=/home/integration/.resilient/logs
cafile=false
```

For cafile and componentsdir make sure you have removed the # at the start of the line too

Save by pressing **CTRL+X**

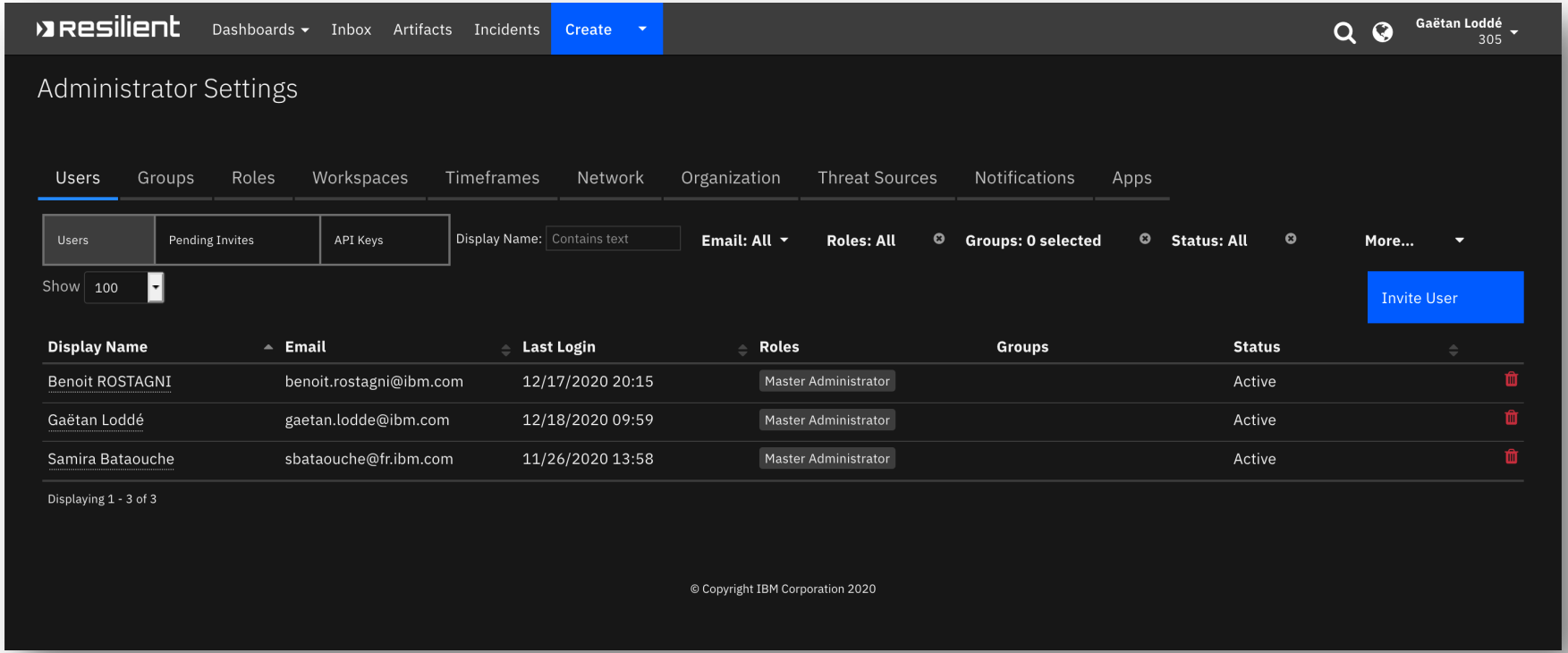
And Create directories:

```
$ mkdir /home/integration/.resilient/logs
$ mkdir /home/integration/.resilient/components
```

Setup the API key

We now need to get an API Key - Open QRadar SOAR in your browser and login with your credentials.

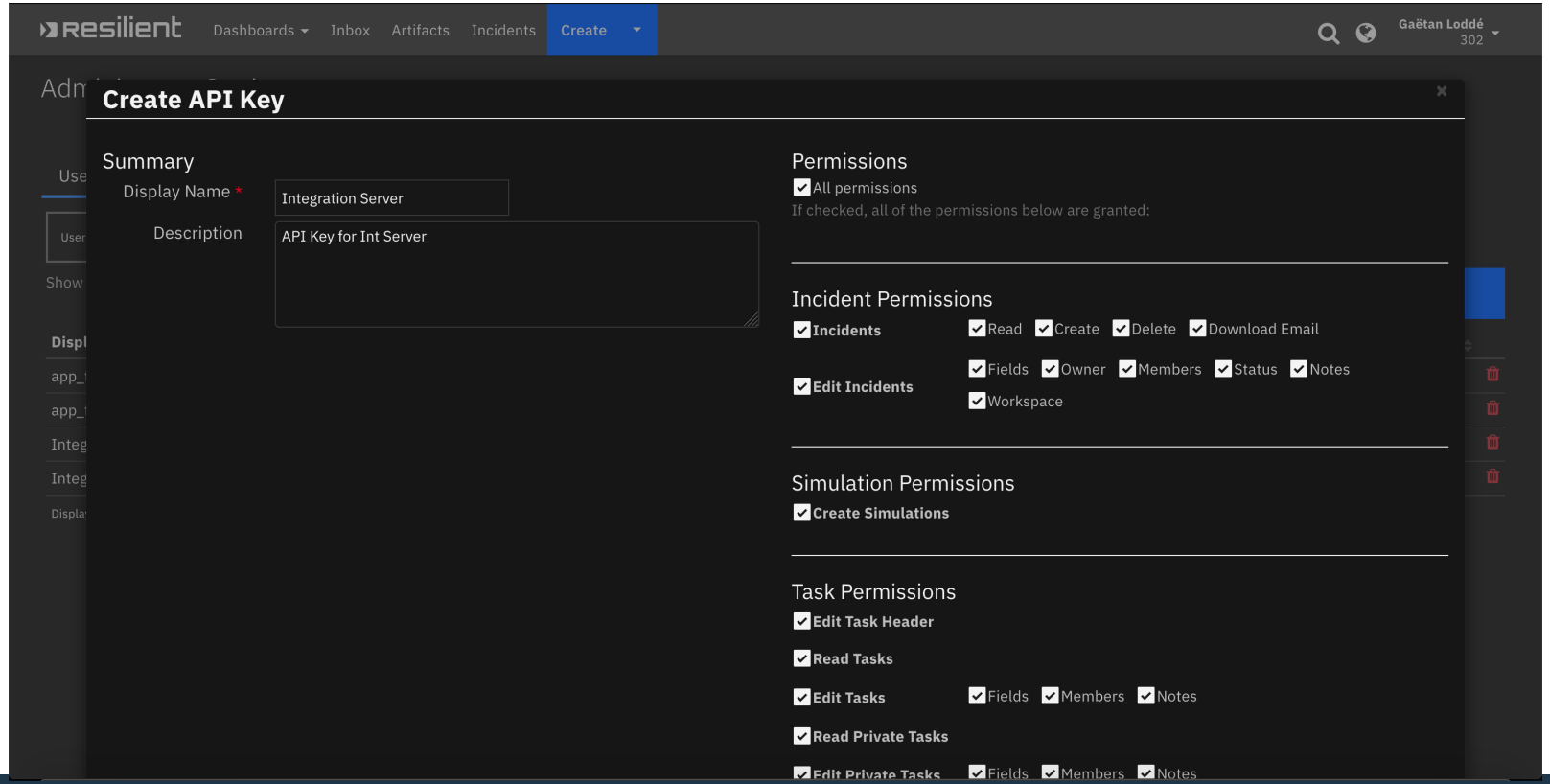
Go to the **`username`** at the top right and click **`Administration Settings`**.



Setup the API key

Click **`API Keys`** on the **`Users`** tab you are on, then **`Create API Key`** on the right.

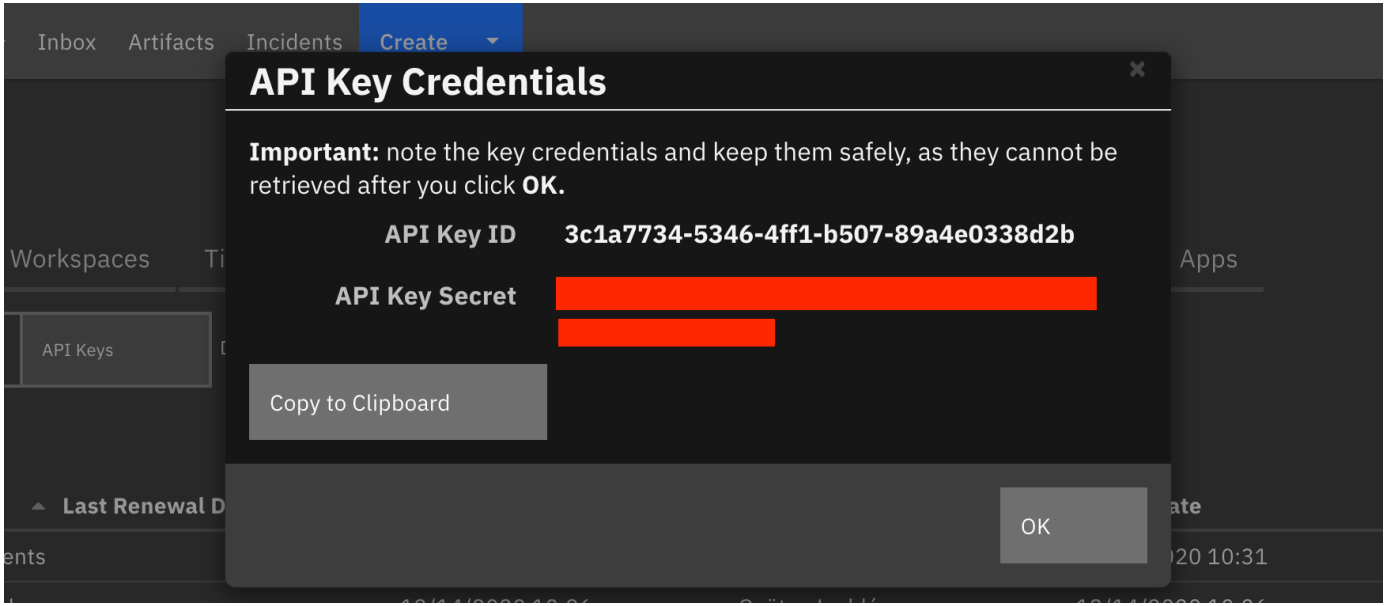
Click **`All Permissions`** for now and name your key



Setup the API key

Scroll down and click Create, a new window will pop up, click Copy to Clipboard.

Reopen the `app.config` file using nano and paste the values. Then put the `Key ID` value against the `api_key_id` and `Key Secret` as the value for `api_key_secret`



- Note: sometimes the api_key_id can paste with a slash at the end, this should be removed.

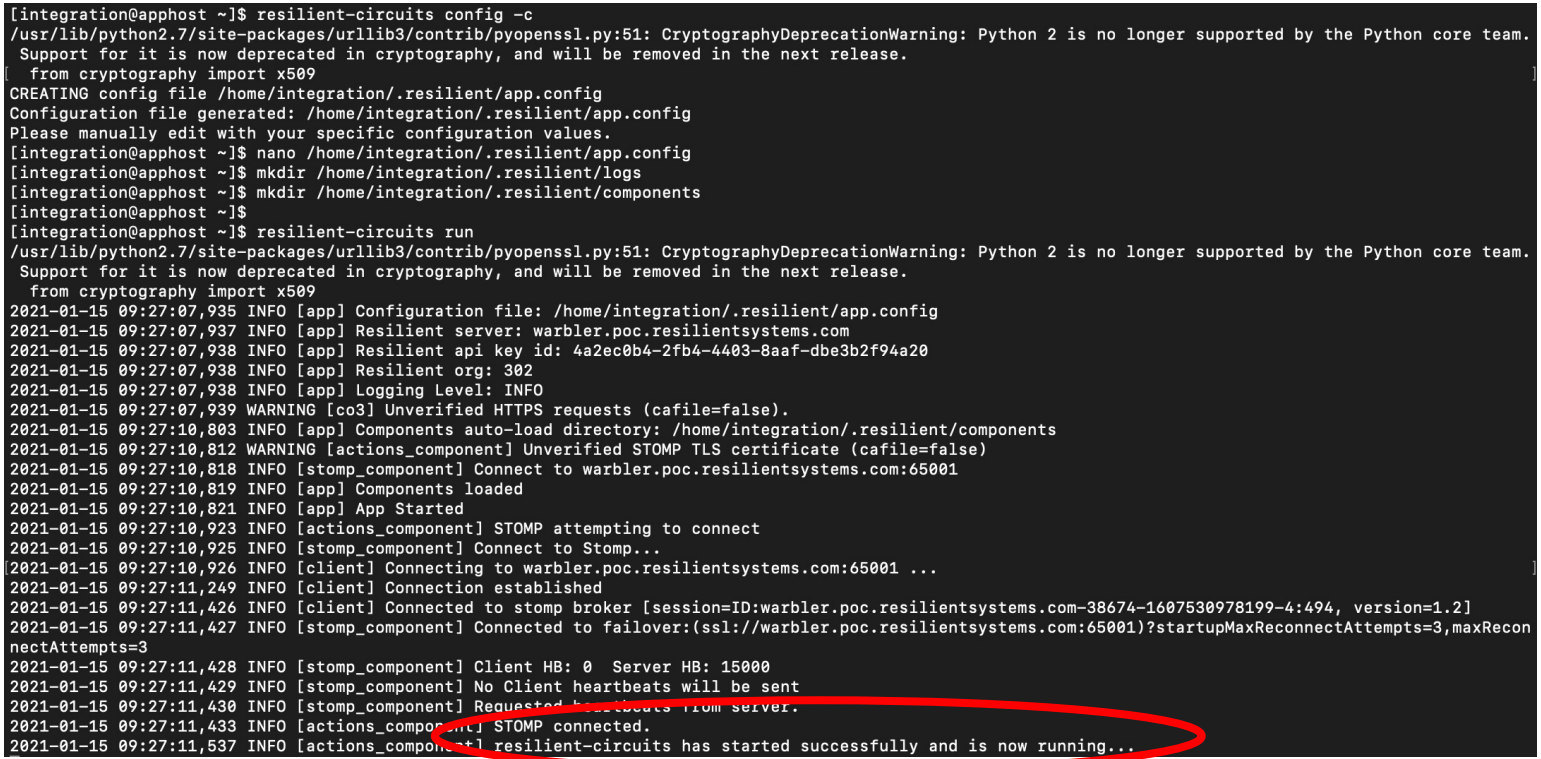
Run QRadar SOAR Circuits

Open **Terminal**

Run resilient-circuits:

```
$ /usr/local/bin/resilient-circuits run
```

You should get this output if resilient-circuits is running successfully:



QRADAR SOAR TRAINING

Lab 13: (optional) Installing QRadar SOAR Circuits as a Service

Goal:

- Understand the notion of a run-on-demand app and an app initialized by a permanent service

Install QRadar SOAR Circuits as a service

- When developing or customizing integrations, it is best to manually run resilient-circuits to quickly iterate and re-run. However in production we want to be able to restart as needed and on reboot etc.
- Resilient-Circuit can be installed as a service on linux or windows (see docs for windows).
- Create a file with `sudo touch /etc/systemd/system/resilient_circuits.service`
- Give the proper permissions with `sudo chmod 664 /etc/systemd/system/resilient_circuits.service`
- Put the contents below in the file: `sudo nano /etc/systemd/system/resilient_circuits.service`

```
[Unit]
Description=Resilient-Circuits Service
# If the Integration Server is added on the same machine than QRadar SOAR (do not do that on Prod),
# uncomment the 2 following lines
# After=resilient.service
# Requires=resilient.service

[Service]
Type=simple
User=integration
WorkingDirectory=/home/integration
ExecStart=/usr/local/bin/resilient-circuits run
Restart=always
TimeoutSec=10
Environment=APP_CONFIG_FILE=/home/integration/.resilient/app.config
Environment=APP_LOCK_FILE=/home/integration/.resilient/resilient_circuits.lock

[Install]
WantedBy=multi-user.target
```

Warning : This line is different from the official doc, in our case the path is different : ExecStart=/usr/bin/resilient-circuits run

Installing QRadar SOAR Circuits as a service

```
sudo systemctl daemon-reload

sudo systemctl enable resilient_circuits.service

sudo systemctl start resilient_circuits.service

sudo systemctl status resilient_circuits.service
```

- Circuits will reboot if there is any errors, it will also start on boot.

```
[integrations@resilient ~]$ sudo systemctl status resilient_circuits.service
● resilient_circuits.service - Resilient-Circuits Service
   Loaded: loaded (/etc/systemd/system/resilient_circuits.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2017-12-13 13:34:24 UTC; 6s ago
 Main PID: 2094 (resilient-circu)
   CGroup: /system.slice/resilient_circuits.service
           └─2094 /usr/local/bin/python2.7 /usr/local/bin/resilient-circuits run

Dec 13 13:34:25 resilient.localdomain resilient-circuits[2094]: 2017-12-13 13:34:25,454 INFO [client] Connecting to resilient.localdomain:65001 ...
Dec 13 13:34:25 resilient.localdomain resilient-circuits[2094]: 2017-12-13 13:34:25,529 INFO [client] Connection established
Dec 13 13:34:25 resilient.localdomain resilient-circuits[2094]: 2017-12-13 13:34:25,631 INFO [client] Connected to stomp broker [session=ID:resilient.localdomain-46049-1513168481358-5...ersion=1.2]
Dec 13 13:34:25 resilient.localdomain resilient-circuits[2094]: 2017-12-13 13:34:25,631 INFO [stomp_component] Connected to failover:(ssl://resilient.localdomain:65001)?maxReconnectAt...tAttempts=1
Dec 13 13:34:25 resilient.localdomain resilient-circuits[2094]: 2017-12-13 13:34:25,631 INFO [stomp_component] Client HB: 0  Server HB: 15000
Dec 13 13:34:25 resilient.localdomain resilient-circuits[2094]: 2017-12-13 13:34:25,632 INFO [stomp_component] No Client heartbeats will be sent
Dec 13 13:34:25 resilient.localdomain resilient-circuits[2094]: 2017-12-13 13:34:25,632 INFO [stomp_component] Requested heartbeats from server.
Dec 13 13:34:25 resilient.localdomain resilient-circuits[2094]: 2017-12-13 13:34:25,633 INFO [actions_component] STOMP connected.
Dec 13 13:34:25 resilient.localdomain resilient-circuits[2094]: 2017-12-13 13:34:25,735 INFO [actions_component] Subscribe to message destination 'shell'
Dec 13 13:34:25 resilient.localdomain resilient-circuits[2094]: 2017-12-13 13:34:25,736 INFO [stomp_component] Subscribe to message destination actions.201.shell
Hint: Some lines were ellipsized, use -l to show in full.
[integrations@resilient ~]$
```

QRADAR SOAR TRAINING

Lab 14: Install the apphost package

Goal:

- Install an AppHost Package on a GUI
- Understand the notion of app pairing

App Host creation

Not Applicable on Remote Saas ***emea-training*** Lab
OVA already prepared for AppHost

- Official documentation to install the run file on your BYORH:
  https://www.ibm.com/support/knowledgecenter/SSBRUQ_39.0.0/doc/apps/deploy_run_file.html

- If you are installing the AppHost on your own Red Hat, please do

```
sudo yum install -y createrepo
sudo subscription-manager repos --enable=rhel-7-server-extras-rpms
sudo yum install container-selinux
sudo firewall-cmd --permanent --zone=trusted --add-interface=cni0
sudo firewall-cmd --permanent --zone=trusted --add-interface=flannel.1
sudo firewall-cmd --permanent --zone=trusted --add-port=443/tcp
sudo firewall-cmd --permanent --zone=trusted --add-port=6443/tcp
sudo firewall-cmd --permanent --zone=trusted --add-port=10250/tcp
sudo firewall-cmd --permanent --zone=public --add-port=22/tcp
sudo firewall-cmd --reload
sudo systemctl restart firewalld
```

And install the program from https://ibm.ent.box.com/file/817940292554?v=ResilientAppHostRun :

```
sudo bash apphost-1.5.218.run
```

```
Creating log directory /usr/share/apphost/logs
Performing post-installation setup...
Installation complete
```

- **The OVA for the training has been prepared for App Host**

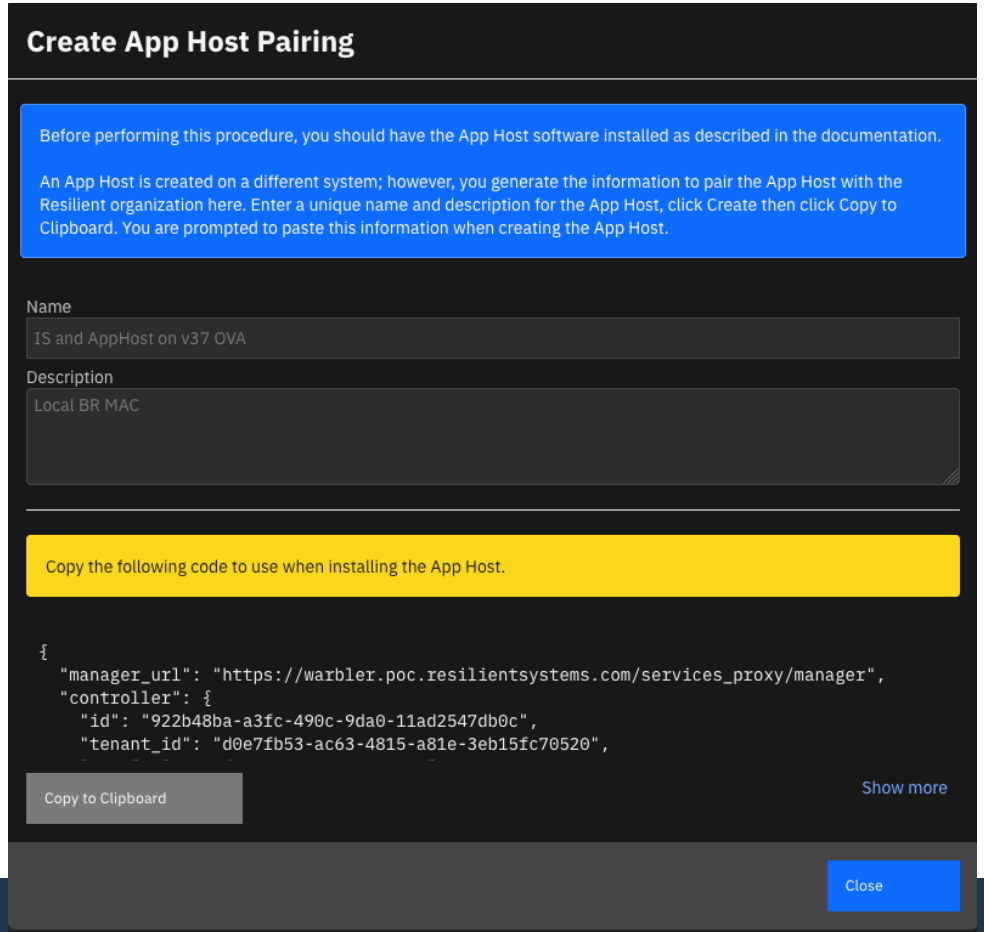
- **Just use the appadmin user and enter the pairing key on the commands:**

```
su appadmin
cd
sudo manageAppHost install
```

App Host Pairing

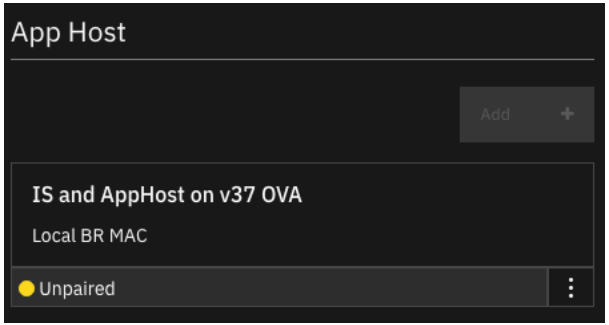
- And go to QRadar SOAR GUI, Administrator Settings > Apps > App Host > [Add +]
- Create the pairing key
- Copy the pairing key to clipboard
- Paste it in the SSH window:

```
-bash-4.2$ sudo manageAppHost install
Enter pairing info: {
  "manager_url": "https://warbler.poc.resilientsystems.com/services_proxy/manager",
  "controller": {
    "id": "922b48ba-a3fc-490c-9da0-11ad2547db0c",
    "tenant_id": "d0e7fb53-ac63-4815-a81e-3eb15fc70520",
    "name": "IS and AppHost on v37 OVA",
    "description": {
      "format": "text",
      "content": "Local BR MAC"
    },
    "app_ids": [],
    "keypair": {
      "private": "-----BEGIN PRIVATE KEY-----\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggS
9En7nJPTKHHP2ESJJaO1eN8Sro9nLffL7G36cKGJ5CEQpLrm4f7s8pmbMC86EfZwnbARd3kjuBU+U6eD2LY9D
L86ID6RkXX8otgyxqT4Xc8E7h8+j8HJE6qPMPydhFzufU+NwOz5sCB1uphnBGUFfcvE9MOhozJsimyaFrZdaT
nxidMXuKWzpJ20yrbKzSbzEfb6/ZgWdwabAUDpfjnSmiQXWAQKBgQDwxbtqogWIbxMQIYB1PuhlIYKbBC3aw6
bh2YaszXzdJ9DTxG8upSV4hnVrk7lRZXCFE4iLjeTGn1gPH/1CIVYvCQhpw7FgIm7prjQz5RNd8MMhNkOYQkH
0jnrfI+a3zbaGgr+ylVcZ4EA2hGaX6aN+6J/SscTaAHlfzPGnzPZan0Qf+sBloVt/FVG1hPtKXdqpzhAoGAN8
oUnGXUCgYAuGeIEnUBVU8JX/CzEoZBHZjb8AfWwCjac63pcEulMJF45VW6syDEaJ+dBGqcK+9C3vfTzedK32g
    },
    "created": 1596477483063,
    "modified": 1596477483063
  }
}_
```

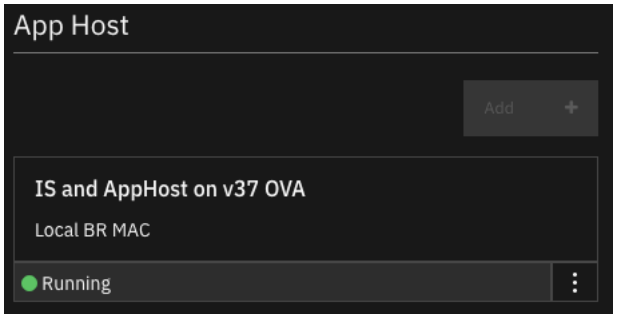


App Host Paired

```
INFO: Installing
INFO: Registering custom resource
INFO: Creating/Updating controller namespace: 922b48ba-a3fc-490c-9da0-11ad2547db0c
INFO: Creating shared secret
INFO: Creating synchronizer secret
INFO: Creating controller-registry-secret secret
INFO: Creating/updating role: operator-role
INFO: Creating/updating role: synchronizer-role
INFO: Creating/updating service account: apphost-synchronizer-service-acct
INFO: Creating/updating service account: apphost-operator-service-acct
INFO: Creating role binding: synchronizer-role-binding
INFO: Creating role binding: operator-role-binding
INFO: Creating/updating synchronizer deployment
INFO: Creating/updating operator deployment
INFO: Sending heartbeat for app host
INFO: Installation succeeded
```



Wait some minutes (5)
or
Reboot the server



QRADAR SOAR TRAINING

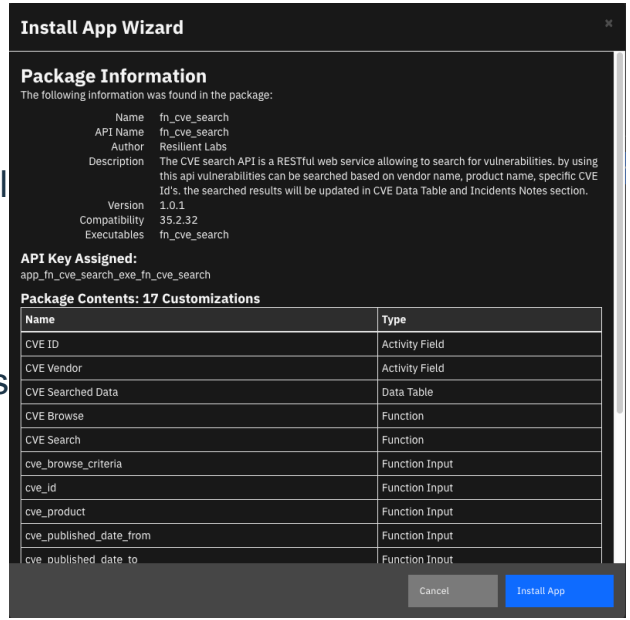
Lab 15: Install your first app

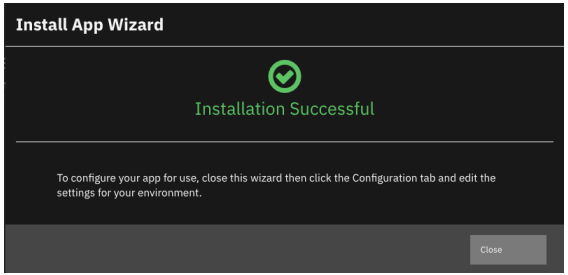
Goal:

- Install and initialize an app in a GUI
- Understand the notion of vulnerability and its significance in a SOC for remediation

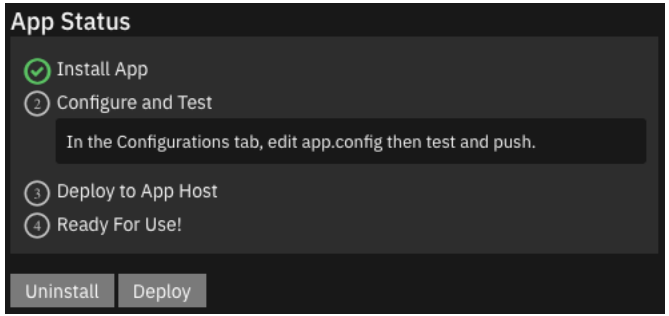
Install your first App

- In QRadar SOAR GUI, Administrator Settings > Apps > App > [Instal
- Select the package `app-fn_cve_search-1.0.1.zip` and upload the file
- You get the result of the package and version, and all customizations
- Press [Install App]



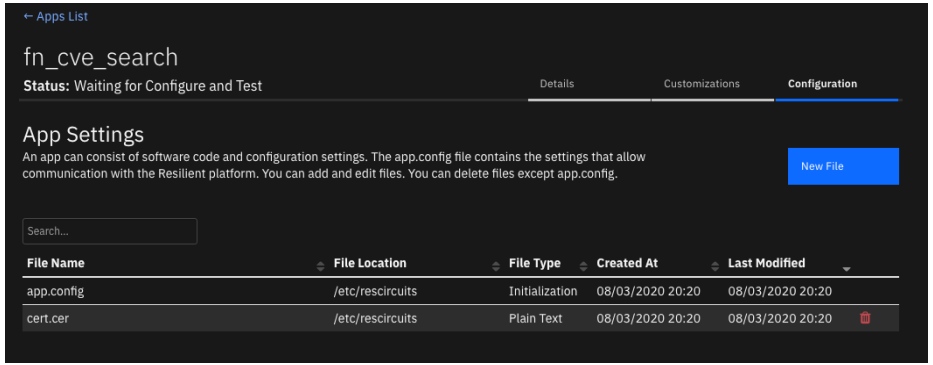


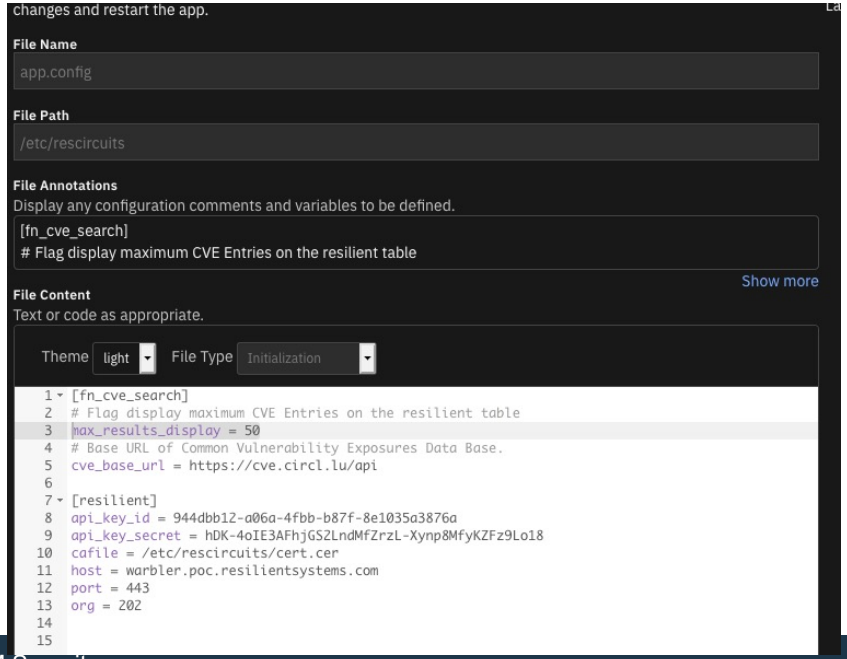
- You get the status:



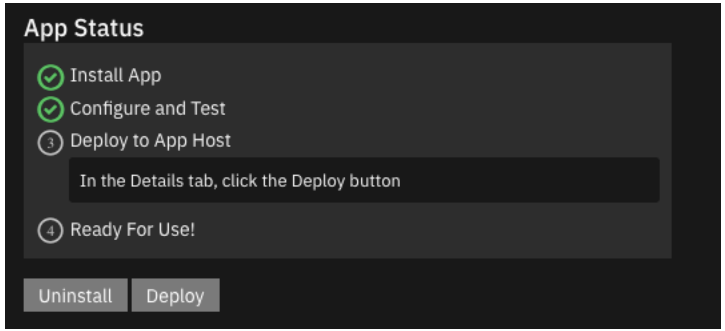
Configure the first App

- Go to Configuration,
  double clic on the file app.config
- If needed, update if necessary the file with the integration specific ApiKey and configuration fields
  Adjust if needed the certificate control (cafile = false)

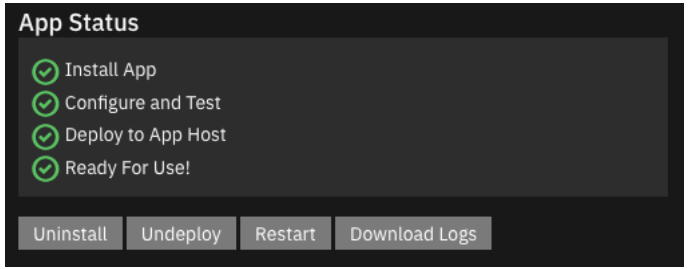




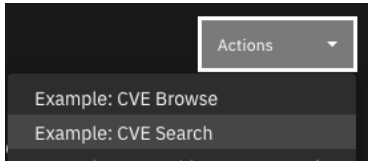
- You can save and push the changes.
  If the configuration is complex, you may test before.
  Note : testing may last for several minutes
- You can deploy to App Host:

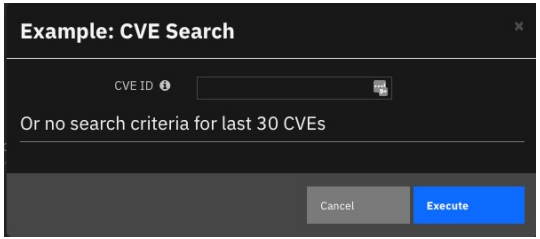


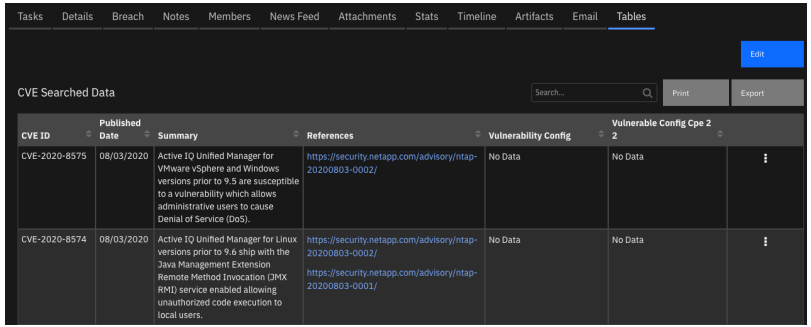
Test your first App



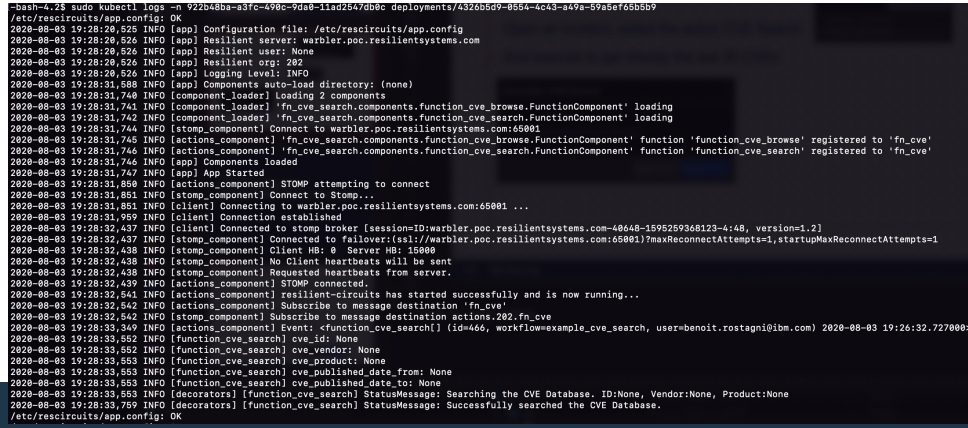
- When done, the apps look like that:
- Create a Table layout tab, and add the CVE Searched Data table in. Save.
- Open an incident, select the action CVE Search
- And execute to get directly the last 30 CVEs and wait for the result.







- Downloading logs show the log like a standard Integration Server



QRADAR SOAR TRAINING

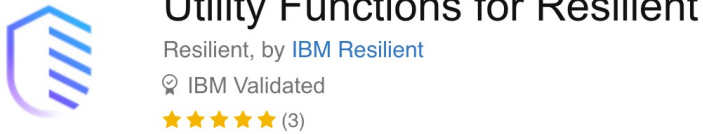
Lab 16: Install fn_utilities app

Goal:

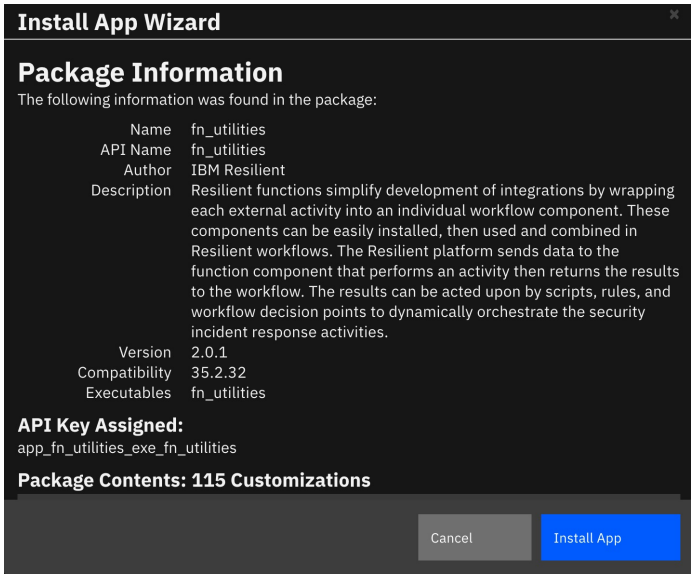
- Install and initialize a complex app on a GUI

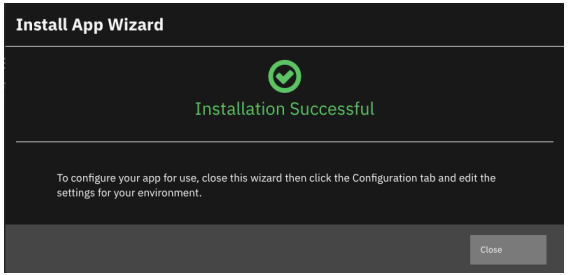
Install your App

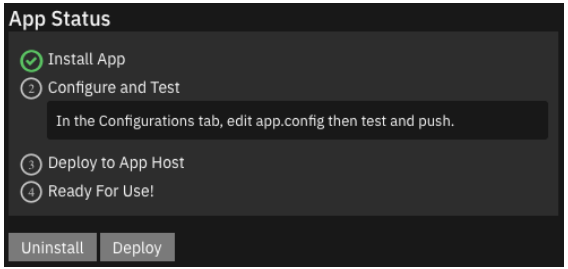
- Go to IBM App Exchange and download Utility Functions for QRadar SOAR



- Like in the previous lab, install the app in your QRadar SOAR GUI
- Select the package app-fn_utilities-x.x.x.zip

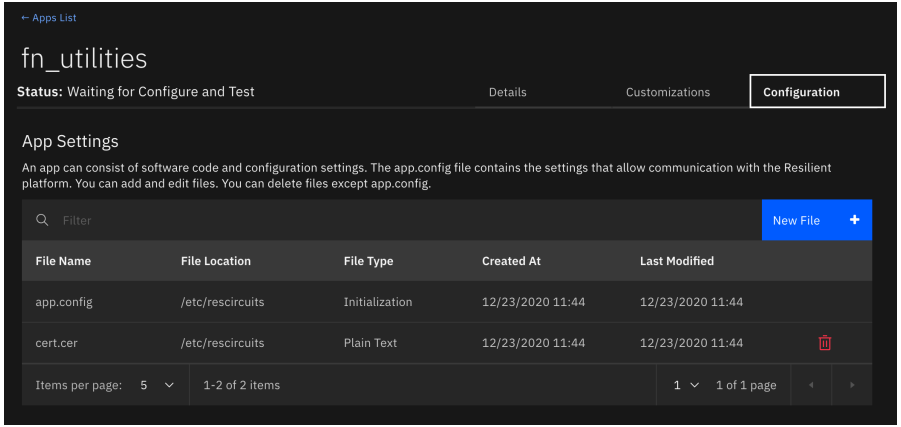


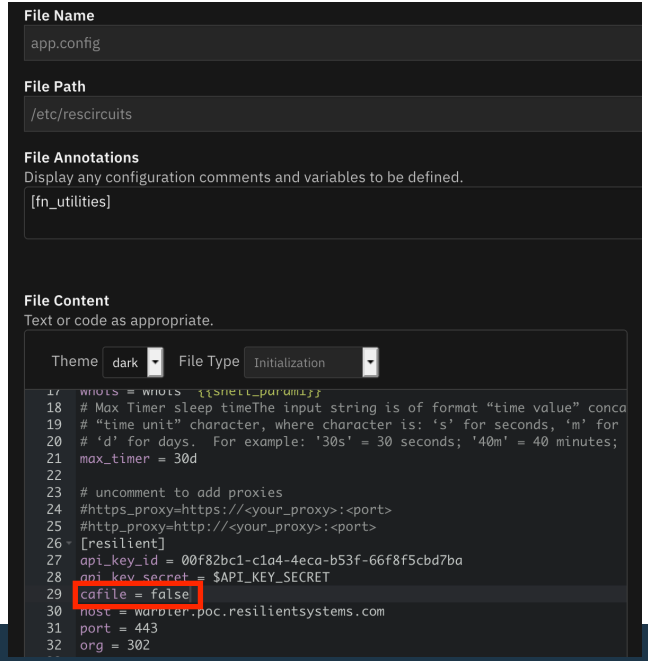




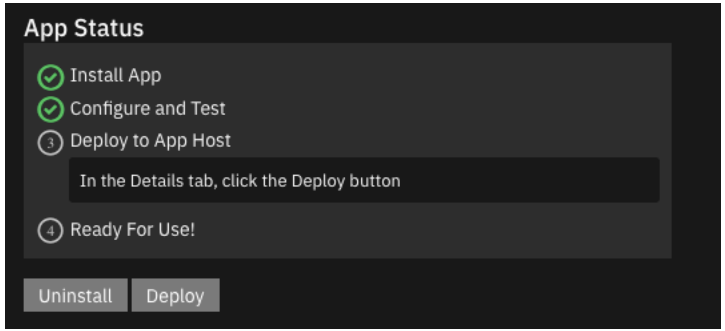
Configure the App

- Go to Configuration,
  double clic on the file app.config
- If needed, update if necessary the file with the integration specific ApiKey and configuration fields
  Adjust if nedeed the certifcate control (cafile = false)





- You can save and push the changes.
  If the configuration is complex, you may test before.
  Note : testing may last for several minutes
- You can deploy to App Host:



QRADAR SOAR TRAINING

Lab 17: Create new Custom Playbook that uses the Shell Command Function

Goal:

- Understand the needs to modify an out-of-the-box solution to match the local production needs
- Apply creating a playbook using functions from another installed app

Create new Custom Playbook that uses our Shell Command Function

Click the **Playbook** Tab

Create a New Playbook:

Name: **nslookup**

For the first object:

Activation Type: **Manual**

Object Type: **Artifact**

Add the Function:

Utilities: ShellCommand

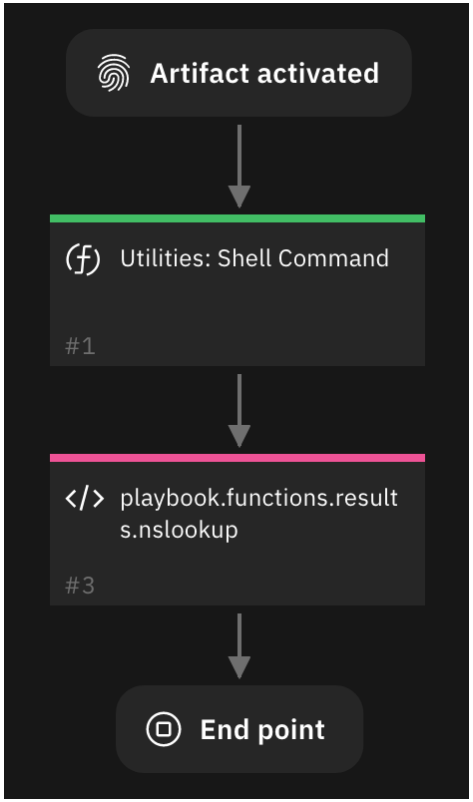
Add a new script:

Name: **playbook.functions.results.nslookup**

(as you can't leave it empty just add any caracter in it and save, you can change the content later)

Add an **End Point** to the canvas

Finally, link everything like in this screenshot



Edit Function script and Post-Process Scripts

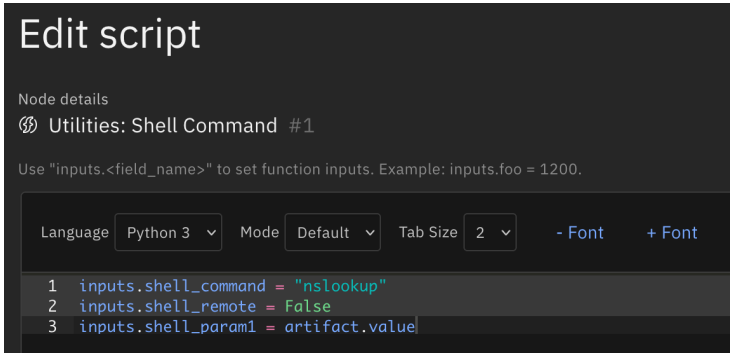
Edit the **Function:**

Function input: **script**

Output name: **nslookup**

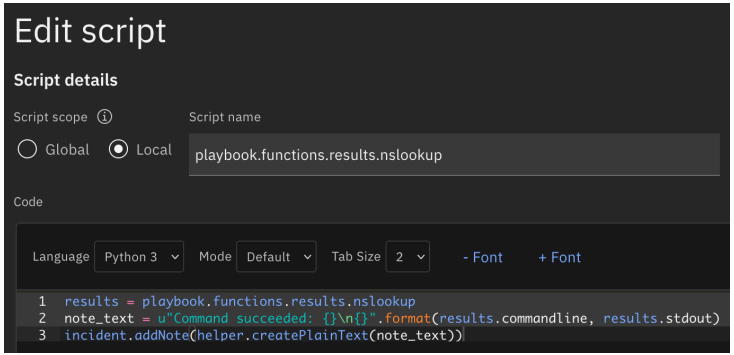
Edit the **Function Script:**

```
inputs.shell_command = "nslookup"
inputs.shell_remote = False
inputs.shell_param1 = artifact.value
```



Edit the **playbook.functions.results.nslookup Script:**

```
results = playbook.functions.results.nslookup
note_text = u"Command succeeded: {}\n{}".format(results.commandline, results.stdout)
incident.addNote(helper.createPlainText(note_text))
```

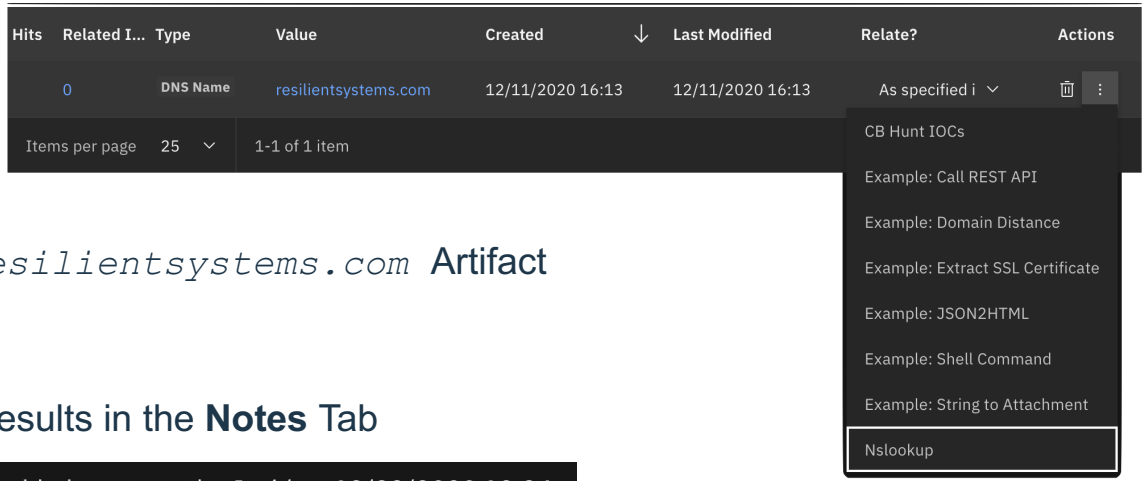


Run our Custom Workflow

to your **Incident**

Click the **Artifacts** Tab

Run *`Nslookup`* playbook on the *`resilientsystems.com`* Artifact



See the **logs in the terminal** and results in the **Notes** Tab

```
app_fn_utilities_exe_fn_utilities added a note to the Incident 12/23/2020 12:34
Command Succeeded: nslookup "resilientsystems.com"
Server:      10.43.0.10
Address:    10.43.0.10#53

Non-authoritative answer:
Name:    resilientsystems.com
Address: 169.46.89.149
Name:    resilientsystems.com
Address: 2620:12a:8001::3
Name:    resilientsystems.com
Address: 2620:12a:8000::3
```

QRADAR SOAR TRAINING

Lab 18: (optional) Install nmap in AppHost and remote control to VM

Goal:

- Understand the notion of Kubernetes, Docker and constraints linked to those environments
- Overtake those constraints configuring remote commands from an app to answer a need

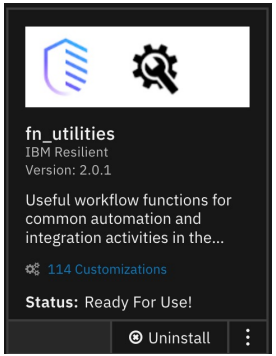
Install nmap in AppHost in ssh

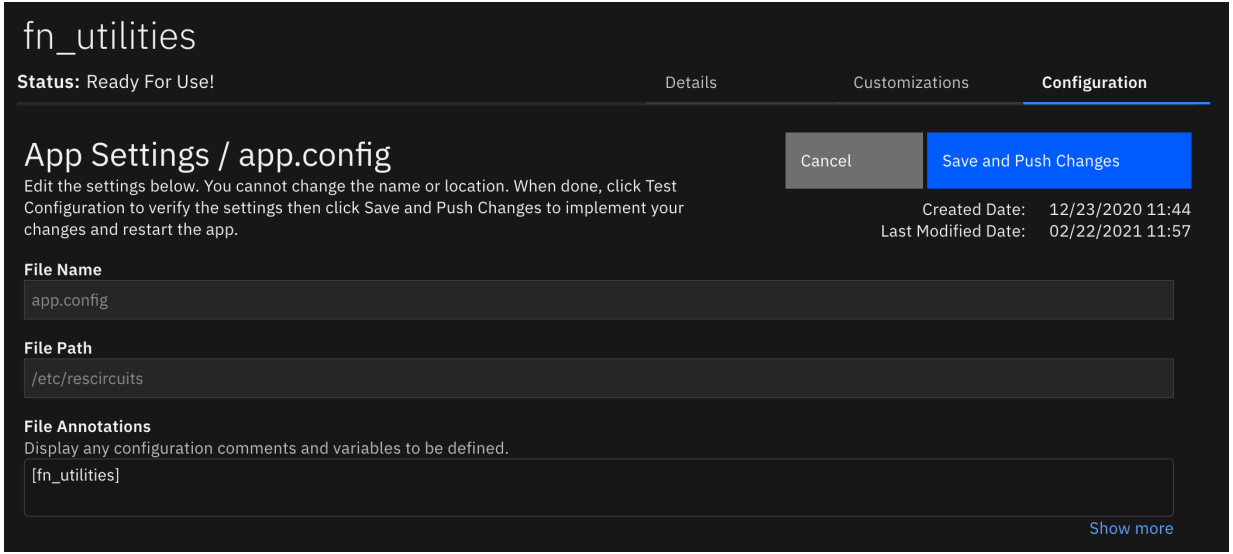
wget https://www.rpmfind.net/linux/centos/7.9.2009/os/x86_64/Packages/libpcap-1.5.3-12.el7.x86_64.rpm
wget http://mirror.ghettoforge.org/distributions/gf/el/7/plus/x86_64/nmap-ncat-7.10-1.gf.el7.x86_64.rpm
wget http://mirror.ghettoforge.org/distributions/gf/el/7/plus/x86_64//nmap-7.10-1.gf.el7.x86_64.rpm

sudo rpm -Uvh libpcap-1.5.3-12.el7.x86_64.rpm
sudo rpm -Uvh nmap-ncat-7.10-1.gf.el7.x86_64.rpm
sudo rpm -Uvh nmap-7.10-1.gf.el7.x86_64.rpm

Configure remote command

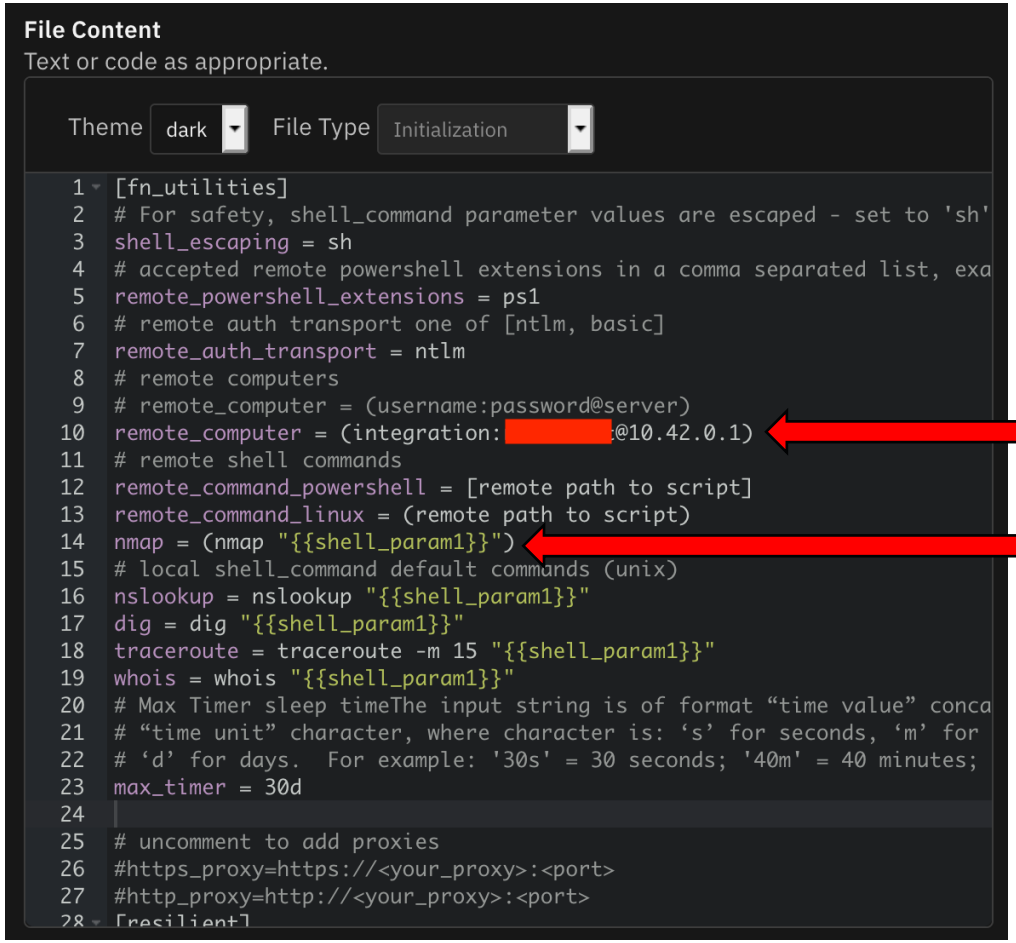
- Go to QRadar SOAR GUI, administrator settings > app
- Click on fn_utilities > configuration > app.config





Configure remote command

- In the « file content » section,



Note: the 10.42 network is the network used by the Apphost micro kubernetees inside the App Host system. We will use the gateway to run the command on the App Host system.
You can check the gateway using the traceroute default example shell command action

Copy this line with your apphost password

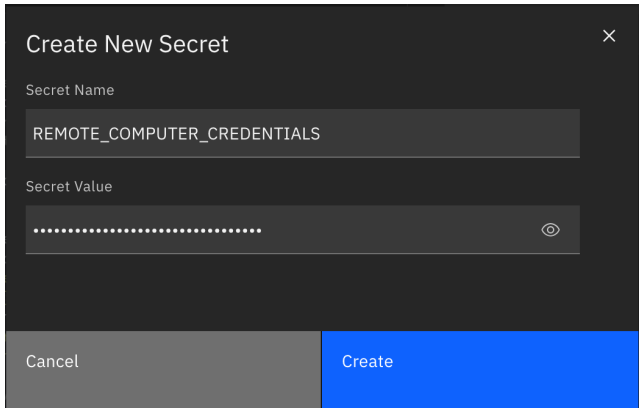
Configure the remote command by writing it in parentheses, the parameter will be added by the workflow

Configure remote command

- For security purposes, you can also add the remote computer as a secret

```
9  # remote_computer = (username:password@server)
10 remote_computer = (integration:[redacted]@10.42.0.1)
```

Copy this line with your apphost password



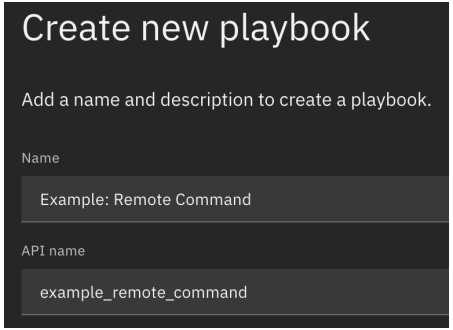
Create a new secret and paste your remote computer credentials **(including the parentheses)** in the Secret Value field

```
8  # remote computers
9  remote_computer = $REMOTE_COMPUTER_CREDENTIALS
```

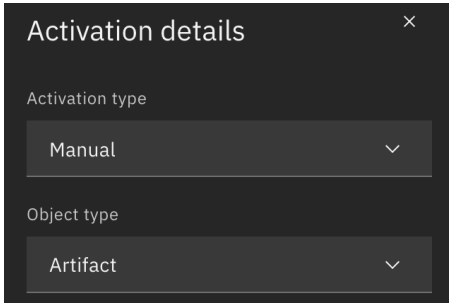
Then replace the remote computer credentials in the file content scipt with the name of the secret

Configure the Playbook

Create a new playbook :
Name: **Example: Remote Command**



First Object:
Activation Type: **Manual**
Object Type: **Artifact**



Add a condition to the first object:

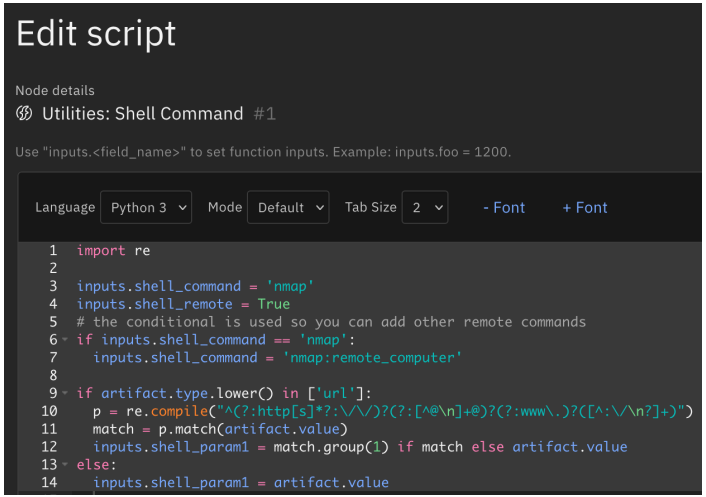
Add the function « Utilities: Shell Command »:

Select Function Input : **Script**

Output name: **nmap**

Function's script:

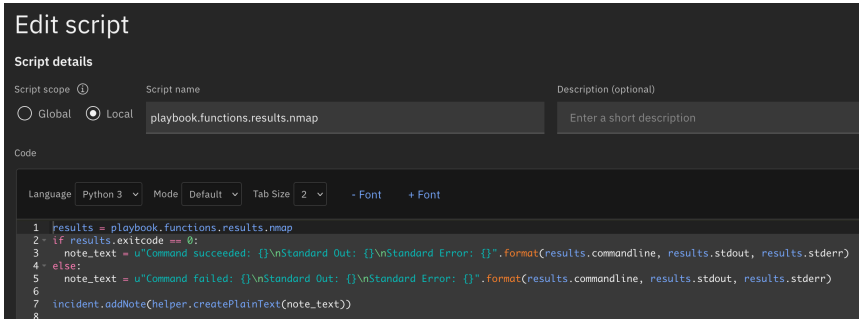
Note : Find the python code in the directory lab file in text for a better copy and paste

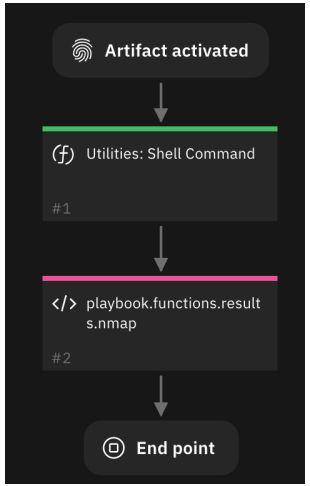


Add a new script:

Name: **playbook.functions.results.nmap**

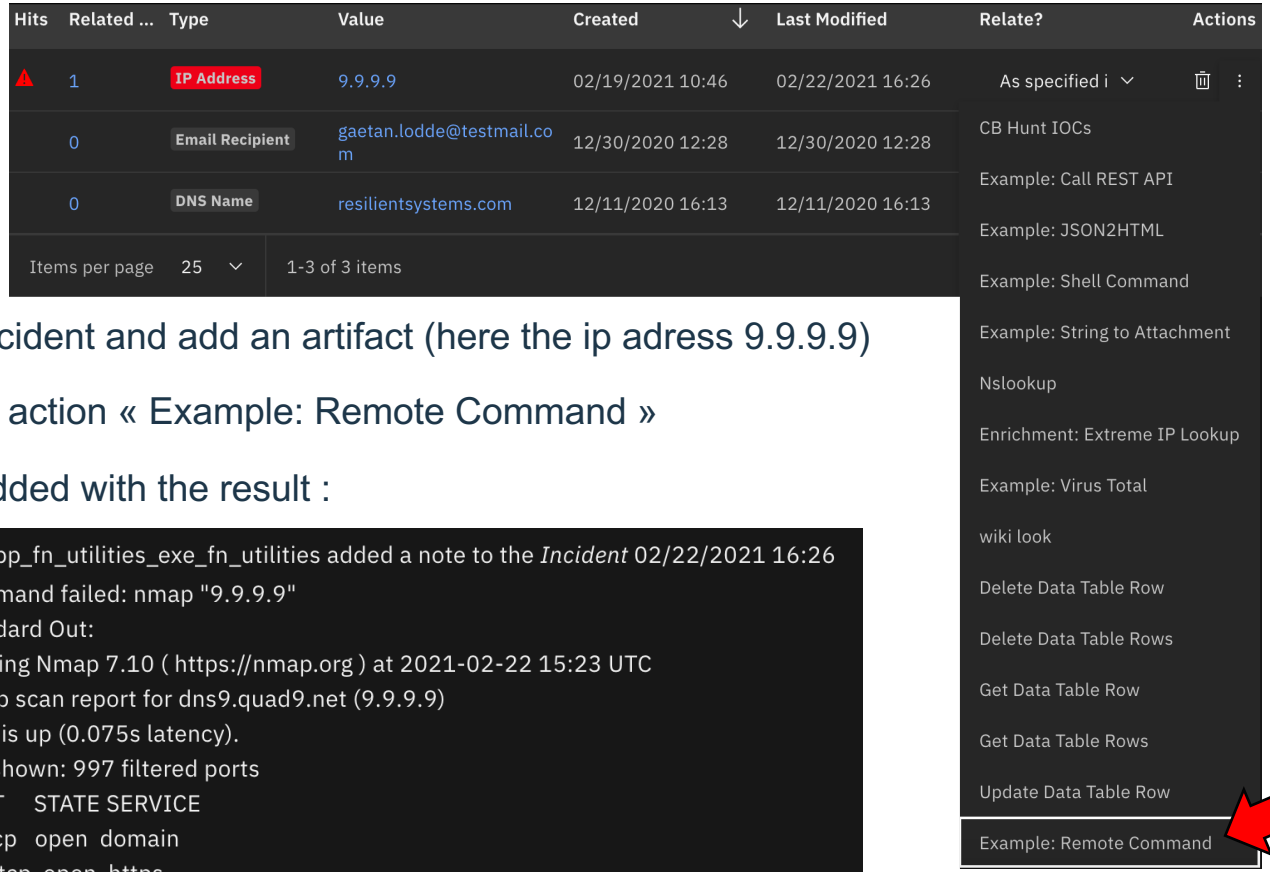
Note : Also find this python code in the directory lab file in text for a better copy and paste





Add an End Point, link every object, Save and enable the playbook

Configure the remote comand



- Go to an incident and add an artifact (here the ip adress 9.9.9.9)
- Launch the action « Example: Remote Command »
- A note is added with the result :

app_fn_utilities_exe_fn_utilities added a note to the *Incident* 02/22/2021 16:26

```
Command failed: nmap "9.9.9.9"
Standard Out:
Starting Nmap 7.10 ( https://nmap.org ) at 2021-02-22 15:23 UTC
Nmap scan report for dns9.quad9.net (9.9.9.9)
Host is up (0.075s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE
53/tcp  open  domain
443/tcp  open  https
8443/tcp open  https-alt

Nmap done: 1 IP address (1 host up) scanned in 6.58 seconds
```

QRADAR SOAR TRAINING

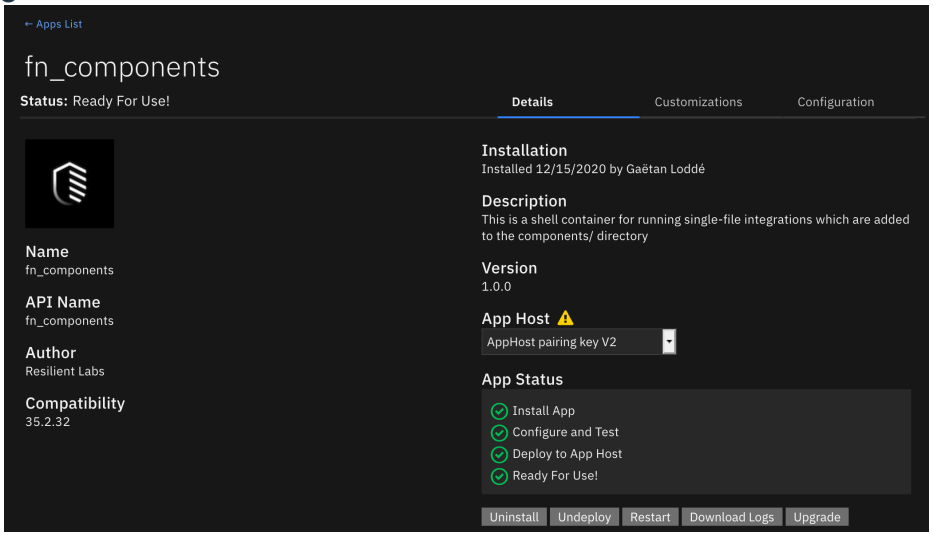
Lab 19: (optional) Install the Components File for App Host

Goal:

- Use an existing app to overcome specific app creation to solve an issue (here collect JSON data from a third-party website and show usable results for the Analyst)

Install fn_components

- Install fn_components in your Lab 21 folder
- In QRadar SOAR GUI, Administrator Settings > Apps > App > [Install +]
- Select the package `app-fn_components-x.x.x.zip` and upload the file



Resilient

App Host Components for Resilient

Allow single-file python integrations to run in App Host

By Resilient Labs

Community Provided

Install and deploy the app as seen in the previous labs

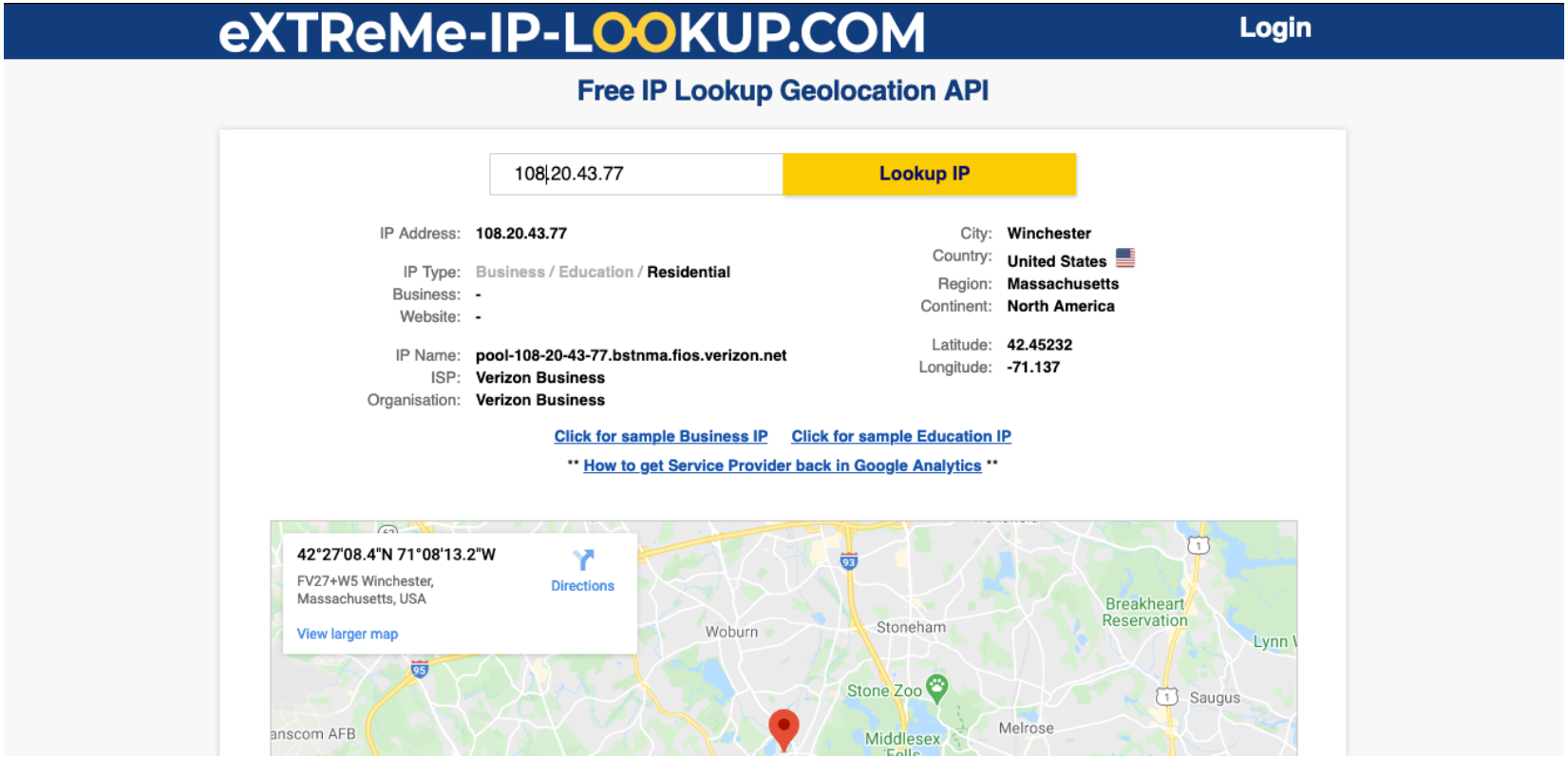
Add the line:

`componentsdir = /var/rescircuits/components`

in the app.config

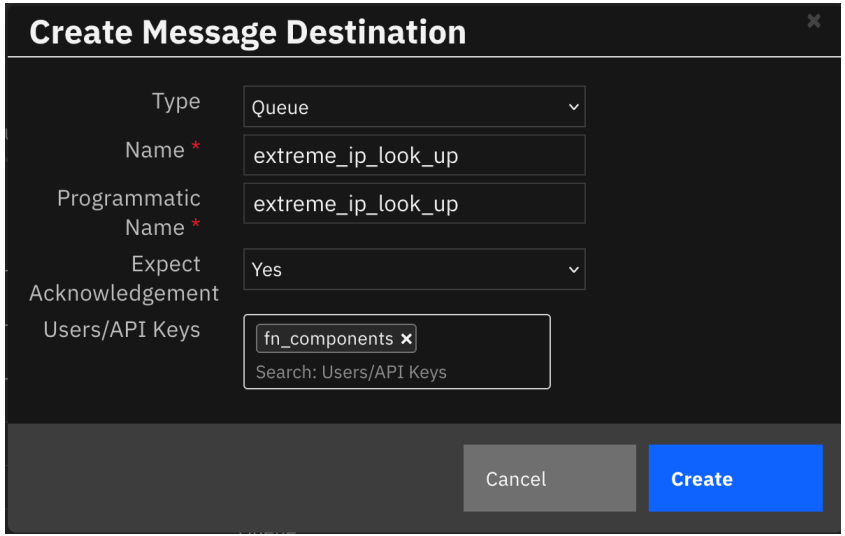
Use Case : Create a new integration

- We are going to create a completely new integration that will collect the json of https://extreme-ip-lookup.com/



Create a Message Destination

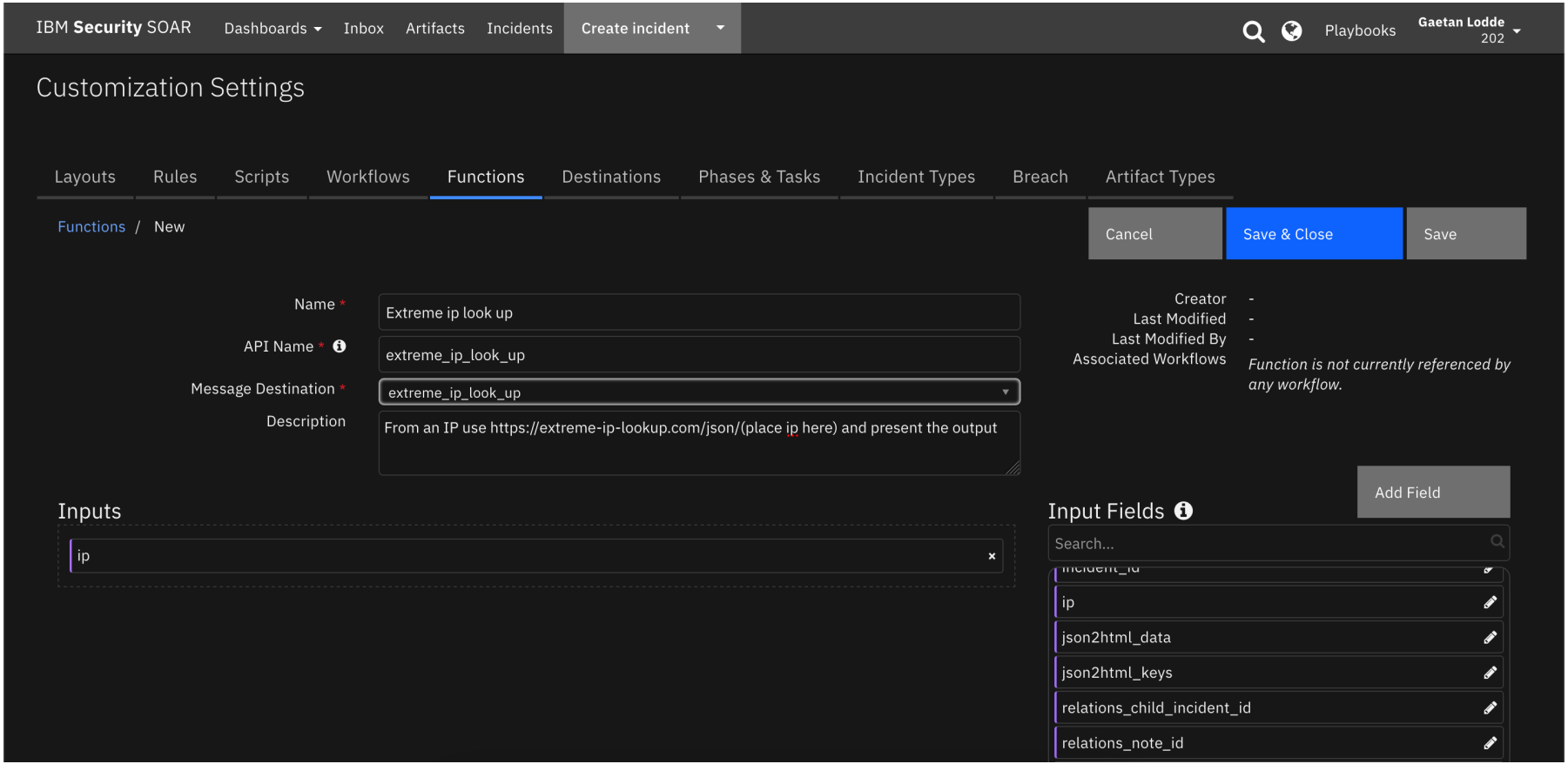
- Create a new message using the app_fn_components app key:



- Remember the Programmatic Name for later `extreme_ip_look_up`

Create the function

- Create the function that will use an IP as source text field



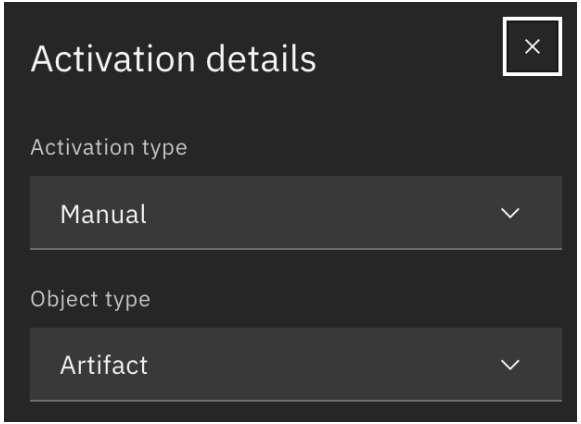
Create the playbook that will run the function from an IP artifact

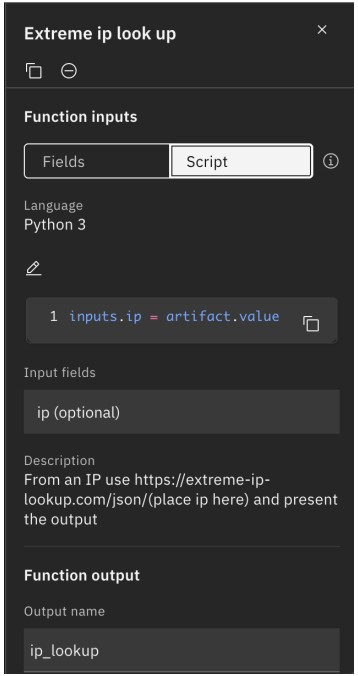
Create a new Playbook:
Name: **Enrichment: Extreme IP Look Up**

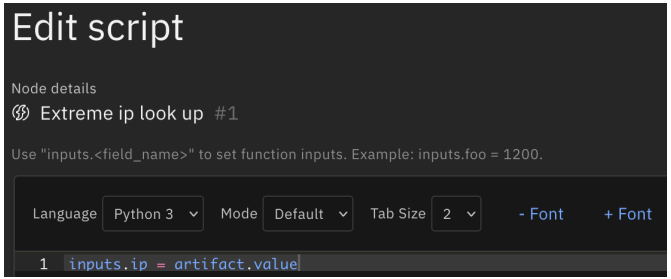
First Object:
Activation Type: **Manual**
Object Type: **Artifact**

Add the function you created:
Name: **Extreme ip look up**
Function Input: **Script**
Function Output: **ip_lookup**

Function's script:
inputs.ip = artifact.value







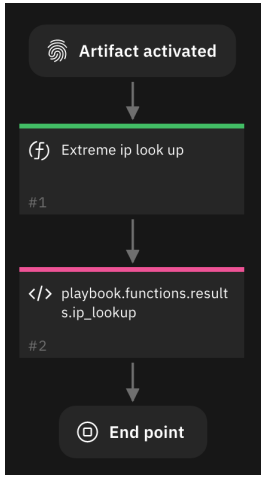
Create a new script to present the data the way you want

Script name: **playbook.functions.results.ip_lookup**

The script is available to copy and paste in your Lab 21 folder, name: **ip_lookup script**

```
"""
Example output from extreme-ip-lookup.com
with numbered my usage in rich text
{
u'status': u'success',
9   u'city': u'Berkeley',
3   u'businessWebsite': u'www.quad9.net',
1   u'ipType': u'Business',
u'countryCode': u'US',
7   u'country': u'United States',
8   u'region': u'California',
10  u'isp': u'Quad9',
11  u'lon': u'-122.27275',
12  u'lat': u'37.87159',
0   u'query': u'9.9.9.9',
2   u'businessName': u'Quad9.net',
4   u'ipName': u'dns9.quad9.net',
5   u'org': u'Quad9',
6   u'continent': u'North America'
}
https://maps.google.com/?q=<lat>,<lng>
"""
```

Add an End Point
Link every object and save the playbook




```
if results.status == 'success':
  rich_text = '<h4><u><b>Extreme IP Lookup on {0}</h4></u></b><br>IP Type: <b>{1}</b><br>Business Name: <b>{2}</b><br>Business Website:
<b>{3}</b><br>ipName: <b>{4}</b><br>Org: <b>{5}</b><br>ISP: <b>{10}</b><br>Continent: <b>{6}</b><br>Country: <b>{7}</b><br>Region:
<b>{8}</b><br>City: <b>{9}</b><br>GeoLocation: <b><a href="https://maps.google.com/?q={11},{12}">Google Maps link on
{11},{12}</a></b>'.format(results['query'], results['ipType'], results['businessName'], results['businessWebsite'], results['ipName'],
results['org'], results['continent'], results['country'], results['region'], results['city'], results['isp'], results['lat'],
results['lon'])
  low_text = ">>Extreme IP Lookup\IP Type: {0}\Business Name: {1}\nCountry: {2}".format(results['ipType'], results['businessName'],
results['country'])

  incident.addNote(helper.createRichText(rich_text))

# write artifact description
if artifact.description is None:
  artifact.description = "{}".format(low_text)
else:
  artifact.description = "{} \n {}".format(artifact.description["content"], low_text)
```

Create a new file in the fn_components App Setting

- Use the Programmatic Name `extreme_ip_look_up.py`
  Enter the File Path: `/var/rescircuits/components`
  The following script can be found in your Lab 21 folder name "extreme_ip_look_up (final) with API Key.py"
  Paste the standard canvas, replacing **xxxxxxxxxxxxxxxxx** by **extreme_ip_look_up** :

```
# -*- coding: utf-8 -*-
# pragma pylint: disable=unused-argument, no-self-use
"""Function implementation"""
import logging
from resilient_circuits import ResilientComponent, function, handler, StatusMessage,
FunctionResult, FunctionError

PACKAGE_NAME = "xxxxxxxxxxxxxxxxx"

class FunctionComponent(ResilientComponent):
    """Component that implements Resilient function 'xxxxxxxxxxxxxxxxx'"""
    def __init__(self, opts):
        """constructor provides access to the configuration options"""
        super(FunctionComponent, self).__init__(opts)
        self.options = opts.get(PACKAGE_NAME, {})

    @handler("reload")
    def _reload(self, event, opts):
        """Configuration options have changed, save new values"""
        self.options = opts.get(PACKAGE_NAME, {})
```

Canvas page 2

- Continue adding the core, again replacing **xxxxxxxxxxxxxxxxx** by **extreme_ip_look_up**
- Duplicate the red line and adjust the name of the fields used in the function definition **zzzzzzzzzzz**

```
    @function("xxxxxxxxxxxxxxxxx")
    def _xxxxxxxxxxxxxxxxx_function(self, event, *args, **kwargs):
        """Place here the program description"""
        try:
            # Get the wf_instance_id of the workflow this Function was called in
            wf_instance_id = event.message["workflow_instance"]["workflow_instance_id"]
            yield StatusMessage("Starting 'xxxxxxxxxxxxxxxxx' running in workflow
'{0}'".format(wf_instance_id))
            # Get the function parameters:
            zzzzzzzzzzz = kwargs.get("zzzzzzzzzzz")  # text

            log = logging.getLogger(__name__)
            log.info("zzzzzzzzzzz : %s", zzzzzzzzzzz)
```

Canvas Page 3

- Continue adding the core, again replacing **xxxxxxxxxxxxxxxxx** by **extreme_ip_look_up**
- Prepare to replace in RED the core of your working Python code, and the output in Json

```
            ##############################################
            # PUT YOUR FUNCTION IMPLEMENTATION CODE HERE #
            ##############################################
            yield StatusMessage("Finished 'xxxxxxxxxxxxxxxxx' that was running in
workflow '{0}'".format(wf_instance_id))
            results = {
                "content": "xyz"
            }

            # Produce a FunctionResult with the results
            yield FunctionResult(results)
        except Exception:
            yield FunctionError()
```

Analyse the working Python Code

This part will go along the other import

```
import requests
from requests.packages.urllib3.exceptions import InsecureRequestWarning
```

This part will go in the middle of the hash part, Except the ip = that already exist from the `kwargs.get`

```
session = requests.session()
session.verify = False
requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
ip='9.9.9.9'

try:
    response = session.get("https://extreme-ip-lookup.com/json/%s" % ip)
    response.raise_for_status()
    ip_data = response.json()

    ipName = ip_data['ipName']
except:
    ipName = ''
```

This part will go in the result, of course without the print

```
print(ip_data)
```

PUT YOUR FUNCTION IMPLEMENTATION CODE HERE

- Replace in RED the core of your working Python code, and the output in Json

```
            ##############################################
            # PUT YOUR FUNCTION IMPLEMENTATION CODE HERE #
            ##############################################

            yield StatusMessage("Finished 'XXXXXXXXXXXXXXXXX' that was running in
workflow '{0}'".format(wf_instance_id))
            results = {
                "content": "xyz"
            }

            # Produce a FunctionResult with the results
            yield FunctionResult(results)
        except Exception:
            yield FunctionError()
```

New in November 2021: we need an API KEY !

- Go to Xtreme-IP web site, tegister your email and get a FREE api Key
- In the dans le `app.config` of the `fn_components`, add the section, and the api key in the secret:

```
[extreme_ip_look_up]
api_key_secret_extremeip = $API_eXtreMe_IP
```

- In the Source code off `extreme_ip_look_up.py`, in the part `Get the function parameters`, add the api key secret lookup:

```
# Get the function parameters:
ip = kwargs.get("ip")  # text
api = self.options.get("api_key_secret_extremeip", None)
```

who works well only if the package name match the section:

```
PACKAGE_NAME = "extreme_ip_look_up"
```

- The session.get command became:

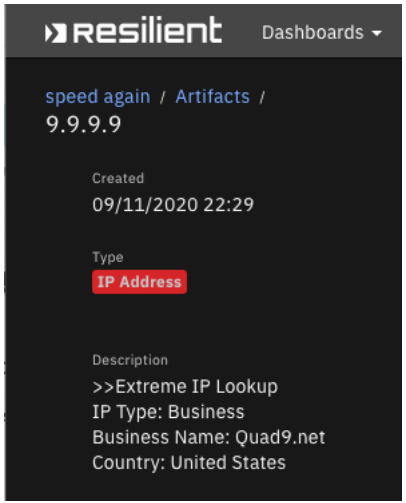
```
try:
  response = session.get("https://extreme-ip-lookup.com/json/%s?key=%s" % (ip,api))
```

Final Check:

```python
# -*- coding: utf-8 -*-
# pragma pylint: disable=unused-argument, no-self-use
"""Function implementation"""

import logging
from resilient_circuits import ResilientComponent, function, handler, StatusMessage, Functi
# Additional Import for this integration
import requests
from requests.packages.urllib3.exceptions import InsecureRequestWarning

PACKAGE_NAME = "extreme_ip_look_up"

class FunctionComponent(ResilientComponent):
    """Component that implements Resilient function 'extreme_ip_look_up'"""

    def __init__(self, opts):
        """constructor provides access to the configuration options"""
        super(FunctionComponent, self).__init__(opts)
        self.options = opts.get(PACKAGE_NAME, {})

    @handler("reload")
    def _reload(self, event, opts):
        """Configuration options have changed, save new values"""
        self.options = opts.get(PACKAGE_NAME, {})

    @function("extreme_ip_look_up")
    def _extreme_ip_look_up_function(self, event, *args, **kwargs):
        """Function: from an IP use https://extreme-ip-lookup.com/json/(place ip here) and
        try:

            # Get the wf_instance_id of the workflow this Function was called in
            wf_instance_id = event.message["workflow_instance"]["workflow_instance_id"]

            yield StatusMessage("Starting 'extreme_ip_look_up' running in workflow '{0}'".f

            # Get the function parameters:
            ip = kwargs.get("ip")  # text

            log = logging.getLogger(__name__)
            log.info("ip: %s", ip)
```

```python

            ##########################################
            # PUT YOUR FUNCTION IMPLEMENTATION CODE HERE #
            session = requests.session()
            session.verify = False
            requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

            try:
                response = session.get("https://extreme-ip-lookup.com/json/%s" % ip)
                response.raise_for_status()
                ip_data = response.json()

                ipName = ip_data['ipName']
            except:
                ipName = ''
            ##########################################

            yield StatusMessage("Finished 'extreme_ip_look_up' that was running in workflow

            # Put the JSON output in results
            results = ip_data

            # Produce a FunctionResult with the results
            yield FunctionResult(results)
        except Exception:
            yield FunctionError()
```

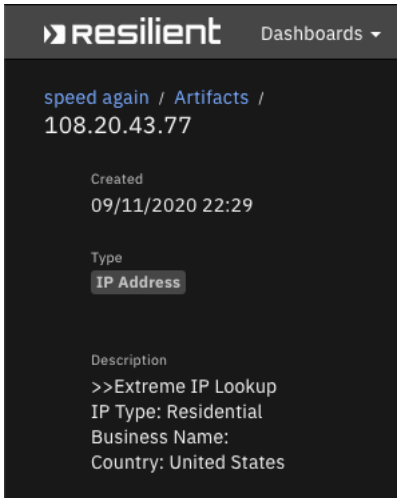
Run & Get the results



app_fn_components_exe_fn_components added a note to the *Incident* 09/12/2020 19:23

Extreme IP Lookup on 9.9.9.9

IP Type: **Business**
Business Name: **Quad9.net**
Business Website: **www.quad9.net**
ipName: **dns9.quad9.net**
Org: **Quad9**
ISP: **Quad9**
Continent: **North America**
Country: **United States**
Region: **California**
City: **Berkeley**
GeoLocation: **Google Maps link on 37.87159,-122.27275**



app_fn_components_exe_fn_components added a note to the *Incident* 09/12/2020 19:23

Extreme IP Lookup on 108.20.43.77

IP Type: **Residential**
Business Name:
Business Website:
ipName: **pool-108-20-43-77.bstnma.fios.verizon.net**
Org: **Verizon Business**
ISP: **Verizon Business**
Continent: **North America**
Country: **United States**
Region: **Massachusetts**
City: **Winchester**
GeoLocation: **Google Maps link on 42.45232,-71.137**

QRADAR SOAR TRAINING

Lab 20: (Requires IBMiD)

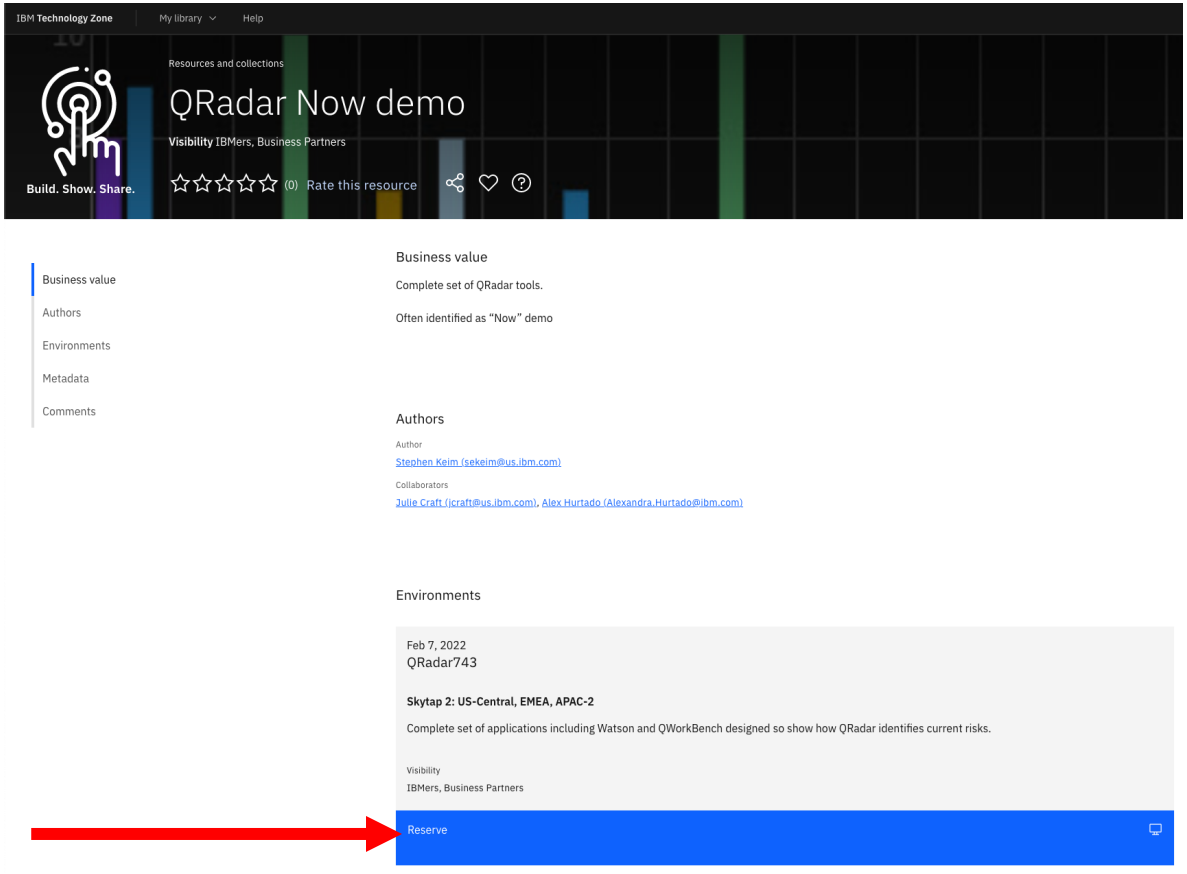
Reserve a QRadar for demo on TechZone, or use your own QRadar Lab (NOT PROD DATA ALLOWED)

Goal:

- Reserving a QRadar environment in TechZone

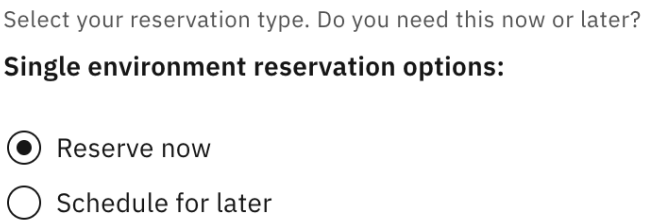
Create a QRadar environment

Reserve a test QRadar environment at https://techzone.ibm.com/collection/q-radar-nowdemo



Create a QRadar environment

Select « Reserve now »

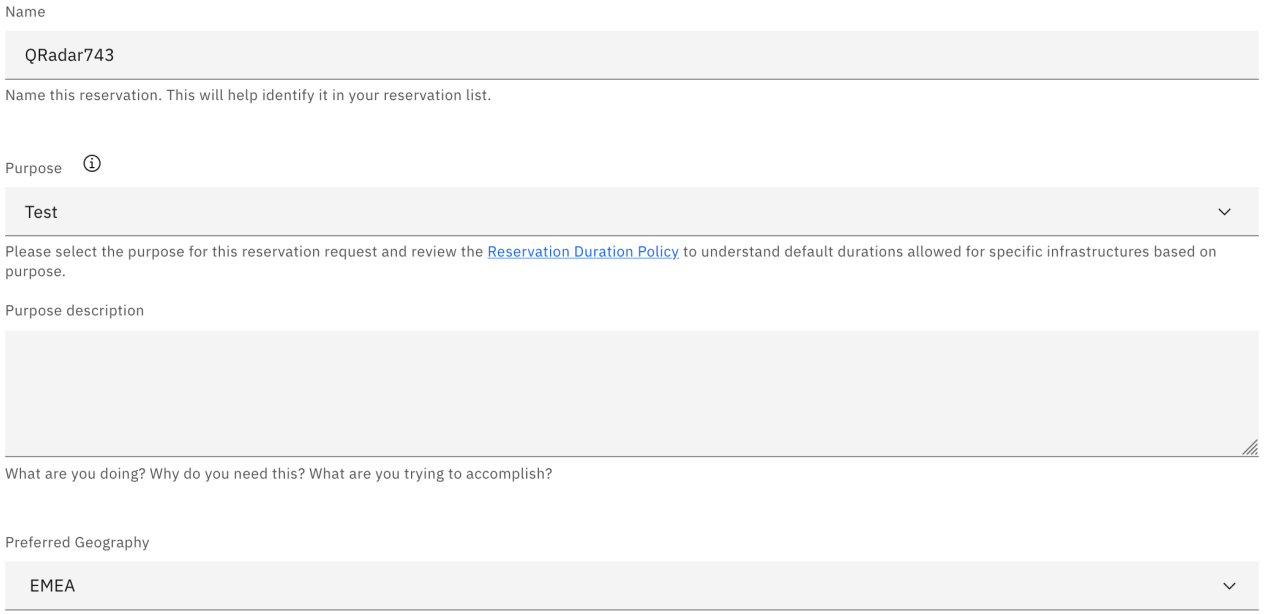


Select « Test » purpose and your prefered geography

You can also select the time of availability of the platform (up to 7 days)

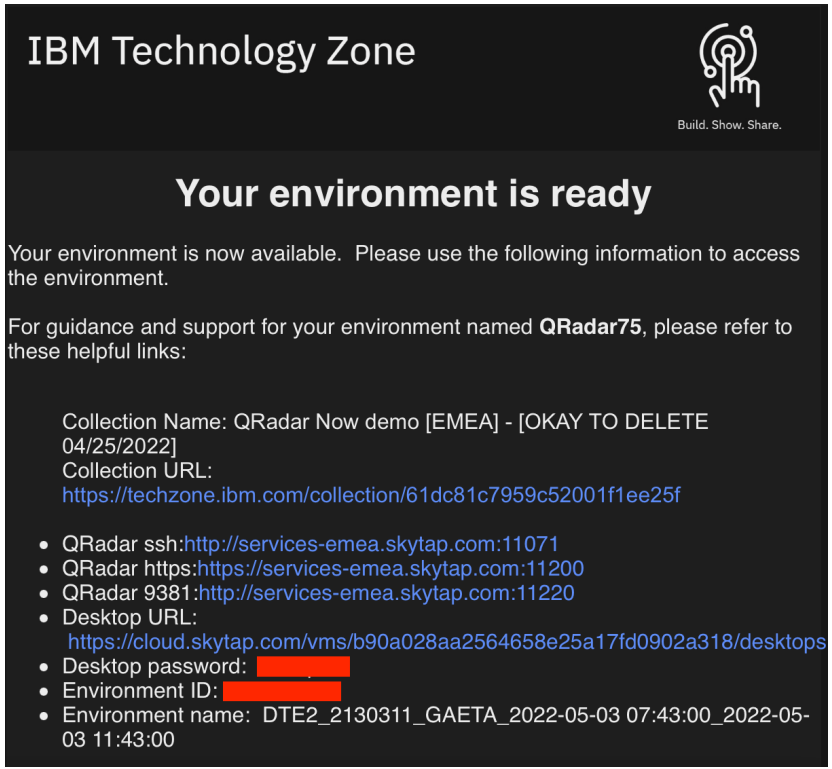
After that you can submit your request and press « Done » on the final page

You will receive later a mail with your connect informations to the Qradar platform



Create a QRadar environment

You should receive an email with credentials when your environment is ready



IBM Technology Zone

Build. Show. Share.

Your environment is ready

Your environment is now available. Please use the following information to access the environment.

For guidance and support for your environment named **QRadar75**, please refer to these helpful links:

Collection Name: QRadar Now demo [EMEA] - [OKAY TO DELETE 04/25/2022]
Collection URL:
https://techzone.ibm.com/collection/61dc81c7959c52001f1ee25f

- QRadar ssh:http://services-emea.skytap.com:11071
- QRadar https:https://services-emea.skytap.com:11200
- QRadar 9381:http://services-emea.skytap.com:11220
- Desktop URL:
  https://cloud.skytap.com/vms/b90a028aa2564658e25a17fd0902a318/desktops
- Desktop password: [redacted]
- Environment ID: [redacted]
- Environment name: DTE2_2130311_GAETA_2022-05-03 07:43:00_2022-05-03 11:43:00

QRADAR SOAR TRAINING

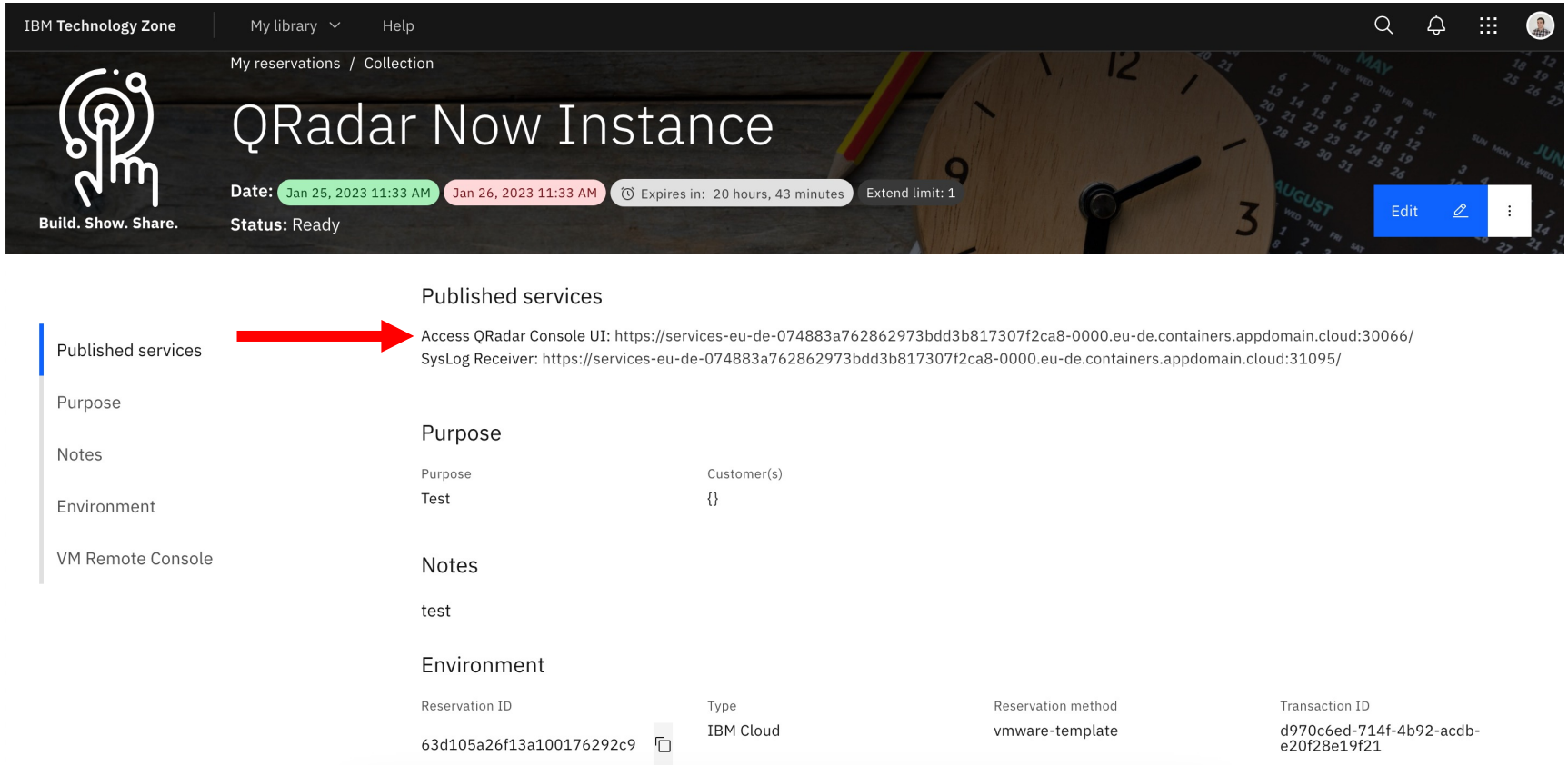
Lab 21: Configure your QRadar to send Alerts to SOAR

Goal:

- Configure QRadar and SOAR to connect each other
- Integrate QRadar in SOAR environment to open, manage and enrich incident

Configure your QRadar

From your Techzone collection page, click on the https link in your web browser



Configure your QRadar

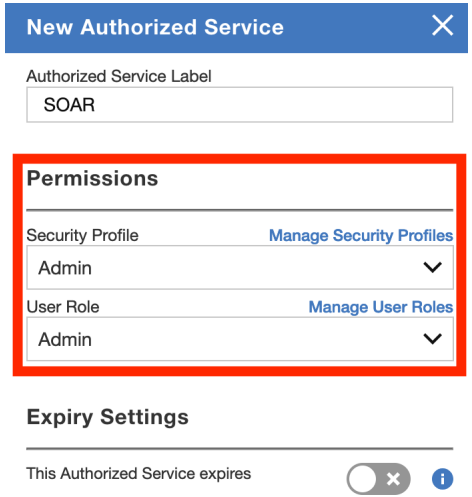
The QRadar https link will open a login page:

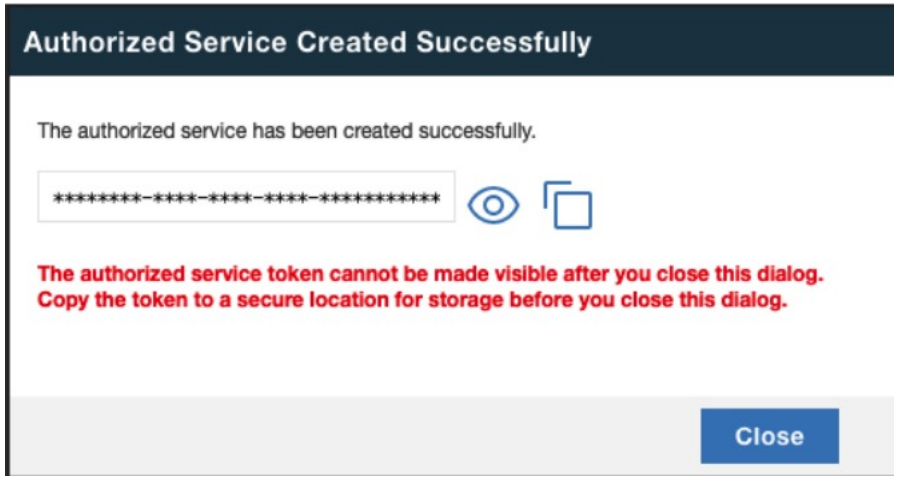
Enter username and password to access supported apps in the IBM QRadar Deployment:
UN: admin/ PW: Q1d3m0.Demo

Configure QRadar

Create a Token to authorize access from SOAR

In your QRadar environment, go to the **Admin Tab**, in **Authorized Service**, **New Authorized Service** with "*Admin*"role & profile, no expiry.



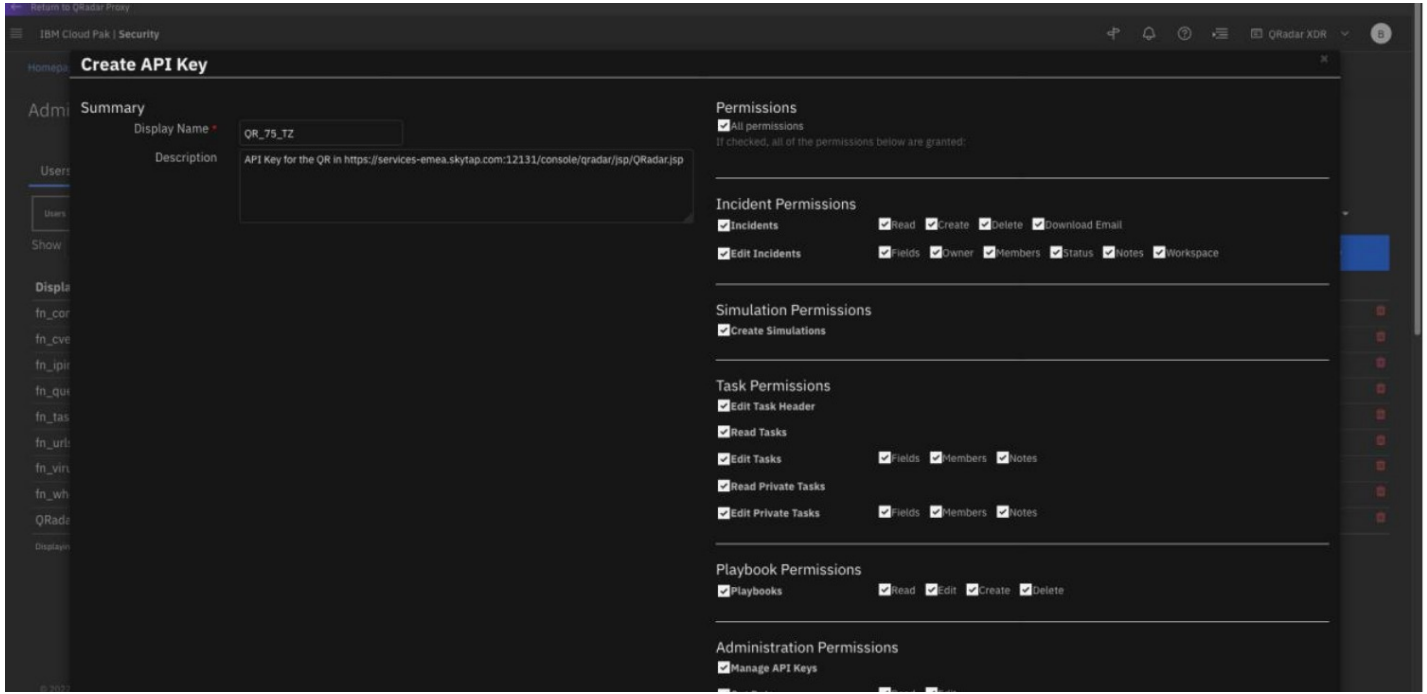


And of course save it in a Safe place

Configure SOAR

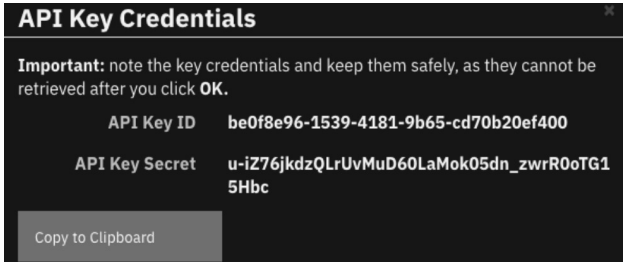
Create a SOAR new API key for QRadar

Administrator settings > Users > API keys > Create API key
Create an API key with ALL permission

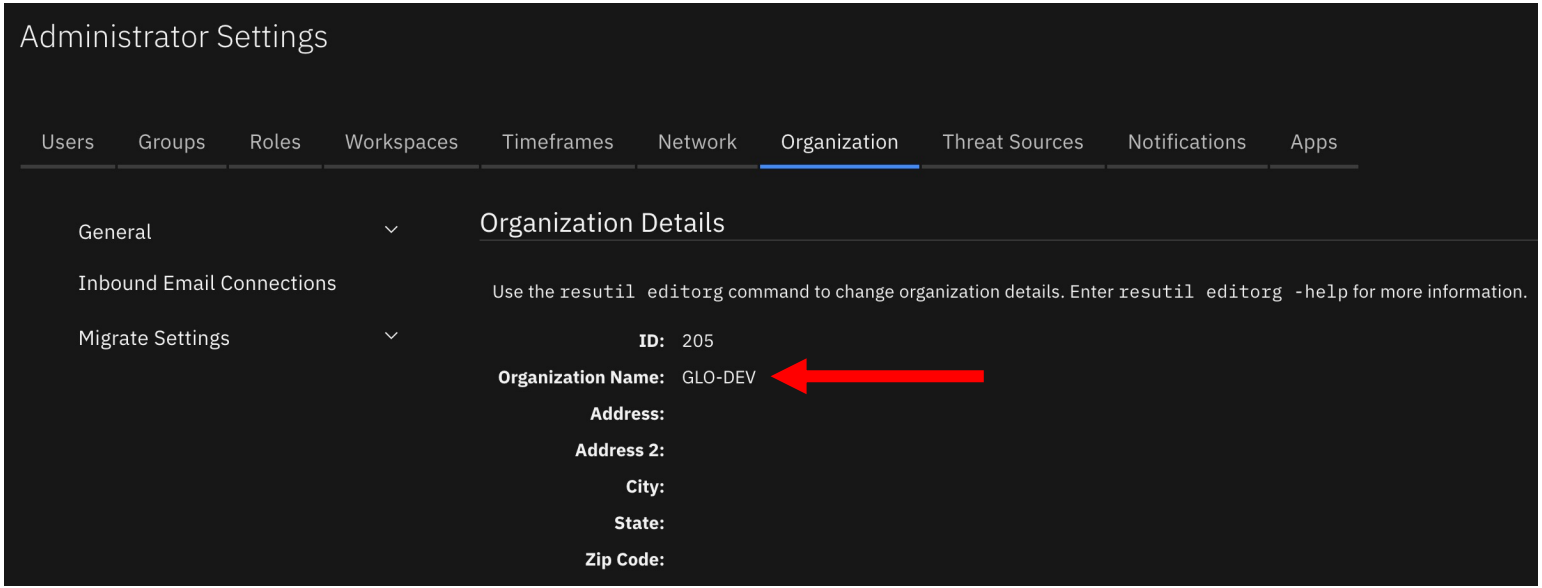


Configure SOAR

Store your credentials in a safe place

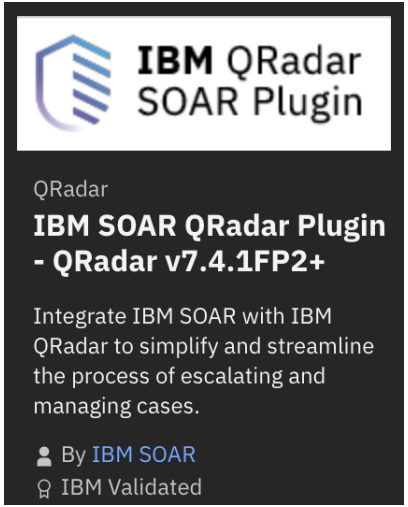


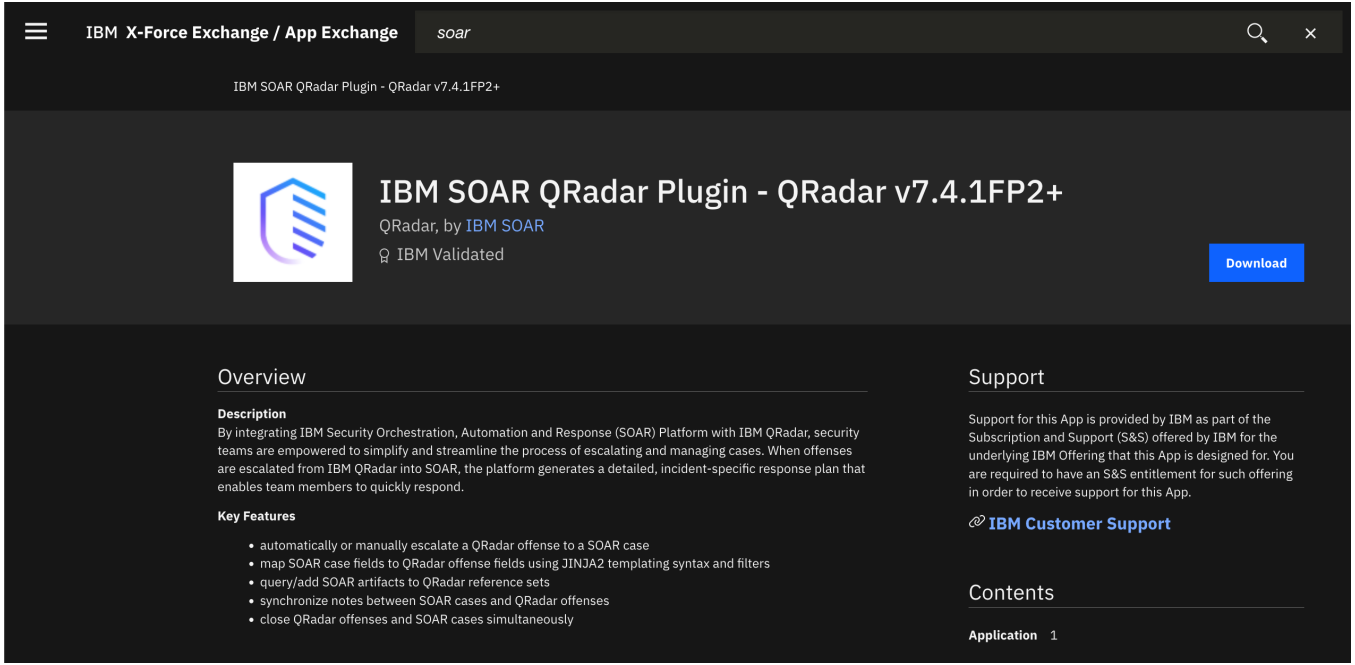
Get the organization name:
Administrator settings > Organization



Download the SOAR Plugin for QRadar

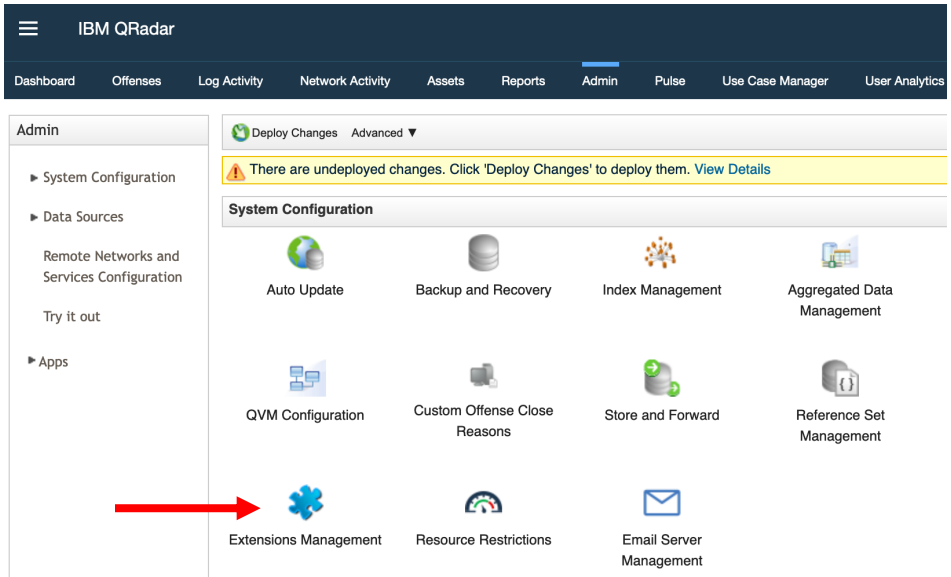
Download the app IBM SOAR Qradar Plugin from the IBM AppExchange:
https://exchange.xforce.ibmcloud.com/hub/extension/bfa8a293489f21da69efec1c4e2c7acc

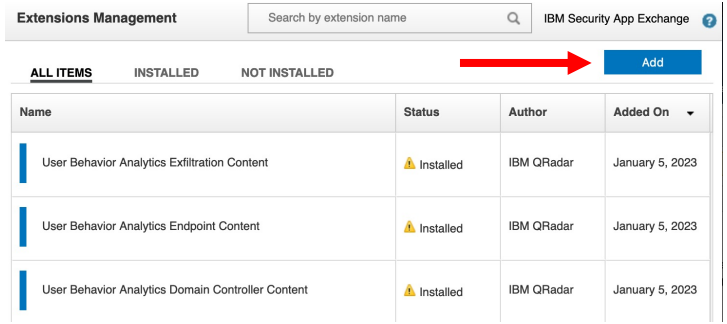


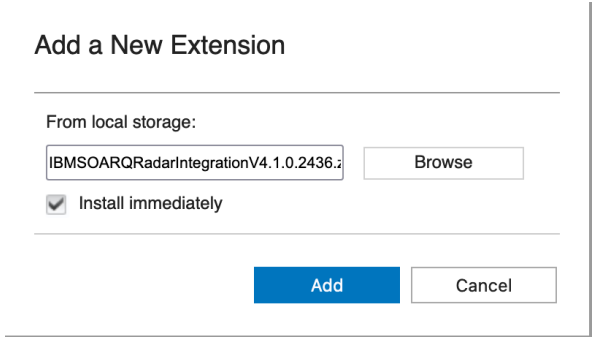


Install the SOAR Qradar Plugin

- Back in Qradar, go to the « admin » tab
- Select « Extensions Management »
- Click the « Add » button
- Select the app you just downloaded from your computer
- Check the « install immediately » checkbox and then press « Add »
- On the window that will appear next, press « install »

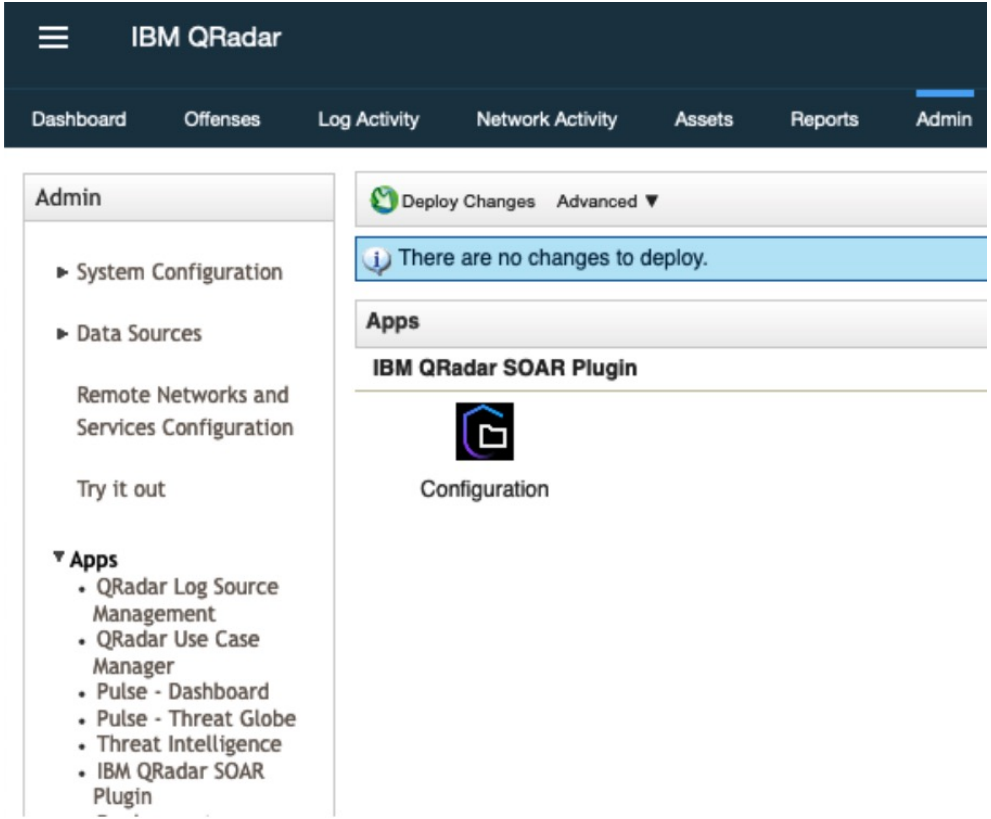




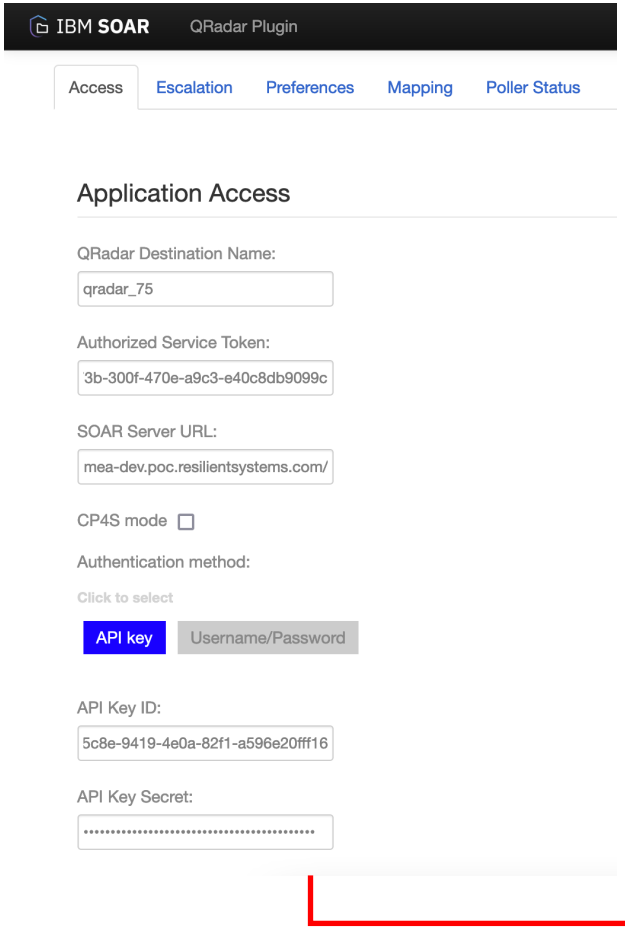


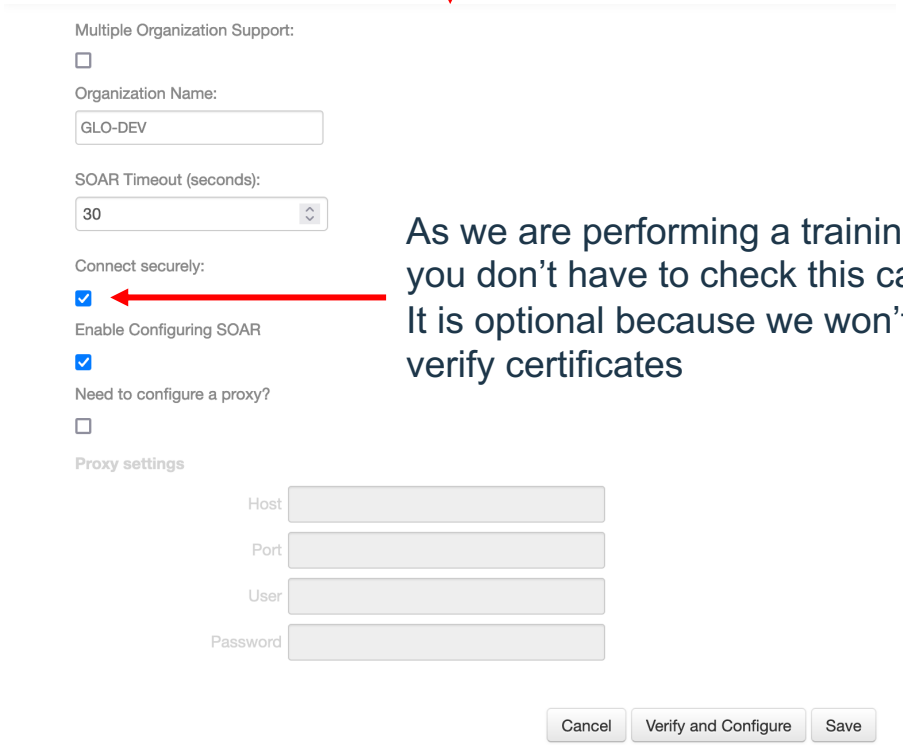
Create the configuration in QRadar

In QRadar Admin > Apps > IBM QRadar SOAR Plugin
Note: if the app does not appear after install, please refresh the page



Create the configuration in QRadar



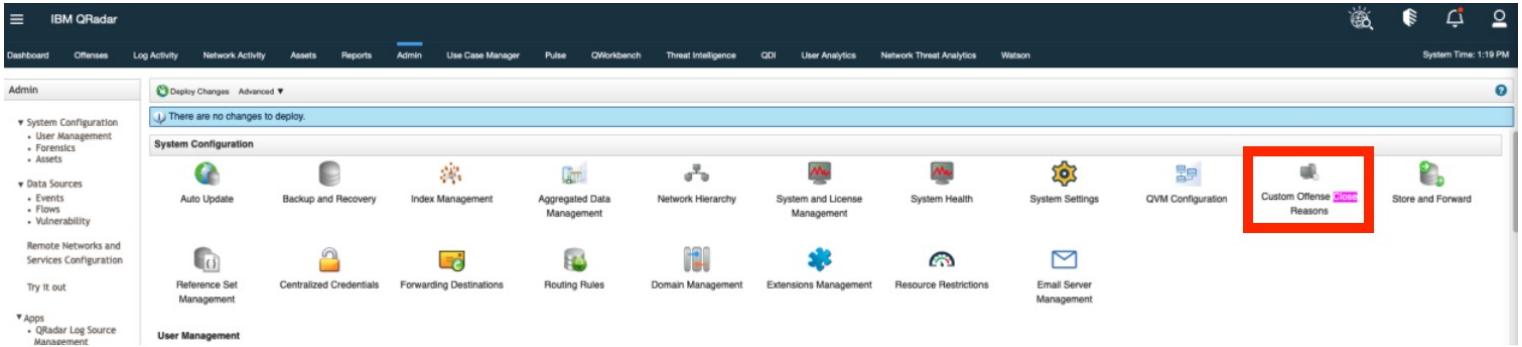


As we are performing a training you don't have to check this case It is optional because we won't verify certificates

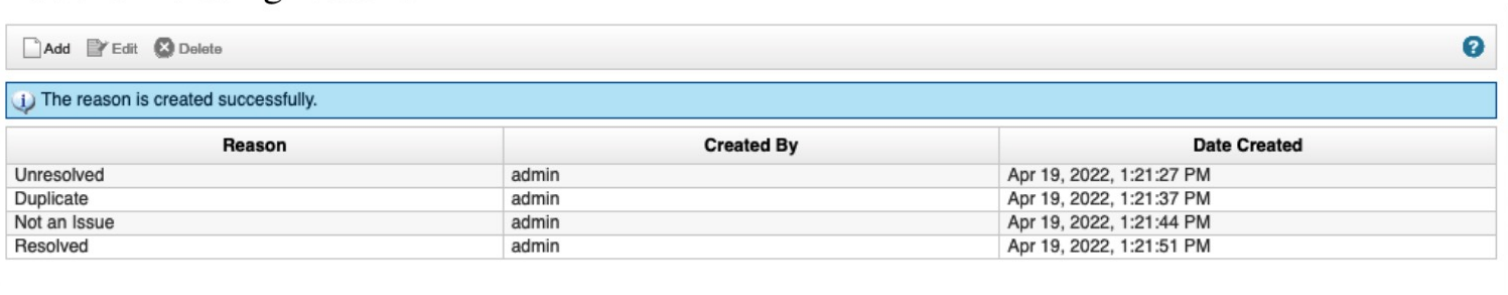
Then you can verify your configuration
Don't forget to save !

Create the configuration in QRadar

The following closing reasons exist in SOAR but not in QRadar.
Unresolved, Duplicate, Not an Issue, Resolved
They should be added to prevent a default closing reason from being used

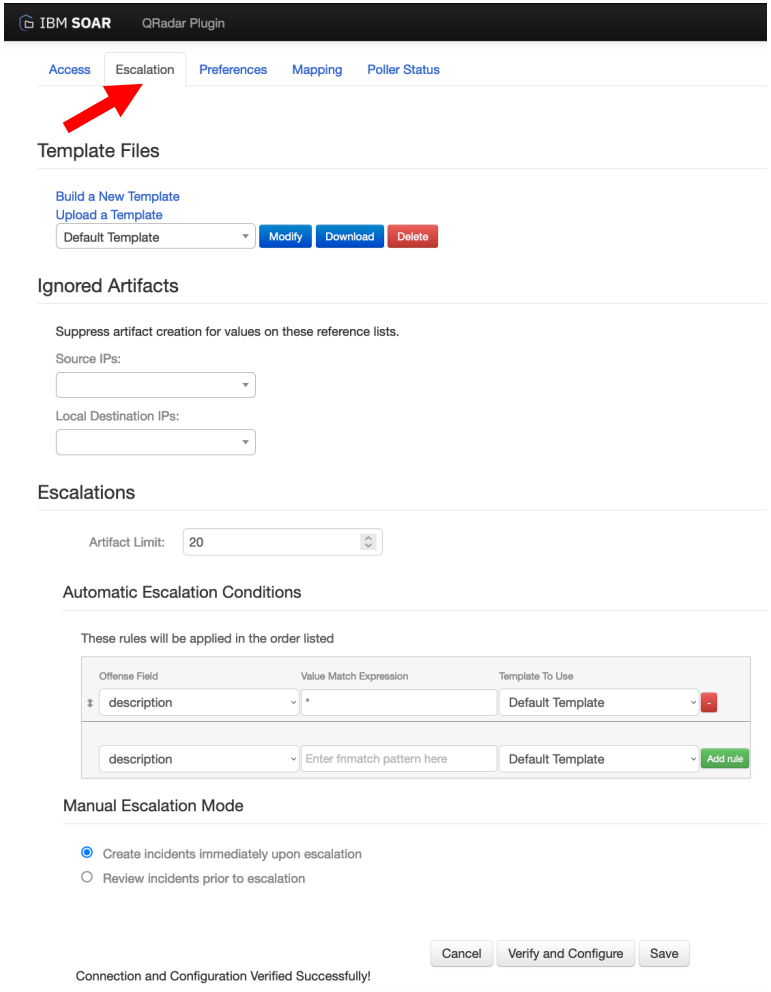


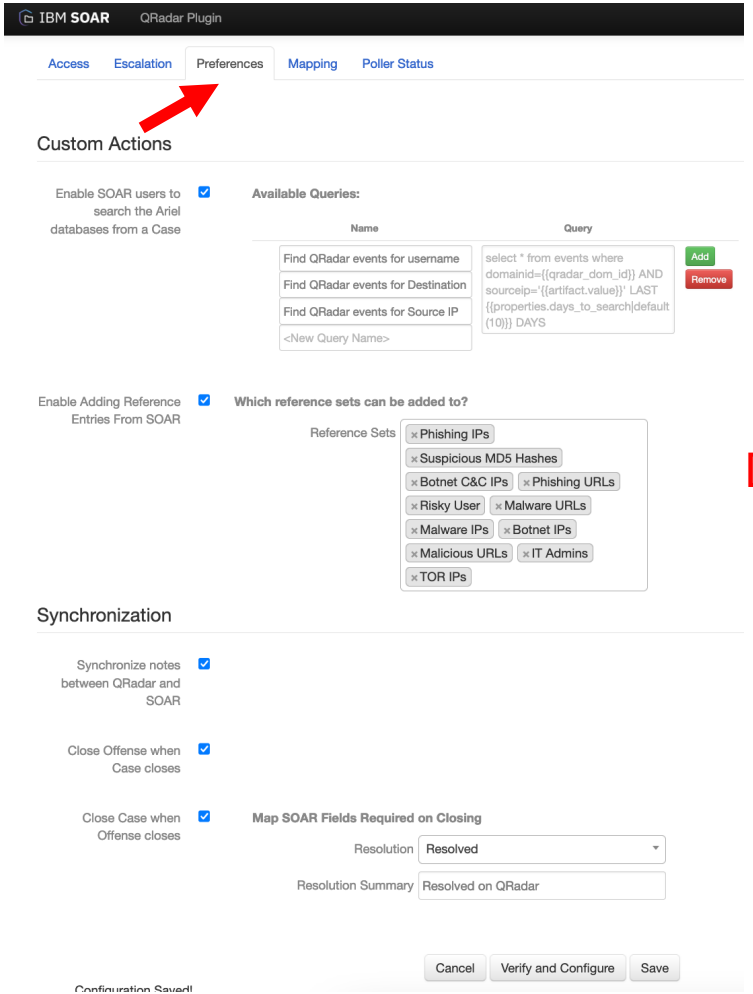
Add the 4 closing reasons



Reason	Created By	Date Created
Unresolved	admin	Apr 19, 2022, 1:21:27 PM
Duplicate	admin	Apr 19, 2022, 1:21:37 PM
Not an Issue	admin	Apr 19, 2022, 1:21:44 PM
Resolved	admin	Apr 19, 2022, 1:21:51 PM

Create the configuration in QRadar





Don't forget
To save !

QRADAR SOAR TRAINING

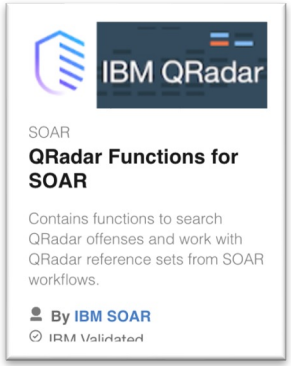
Lab 22: Configure your SOAR to Query QRadar with 2 apps

Goal:

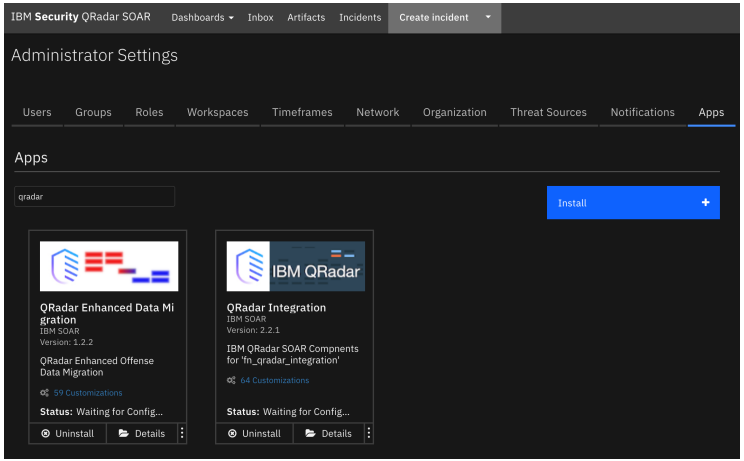
- Deploy App using Qradar to enrich an incident
- Connect SOAR to a SIEM to gather data from an external app

Create a QRadar environment

Download and add the following two apps to your SOAR platform:

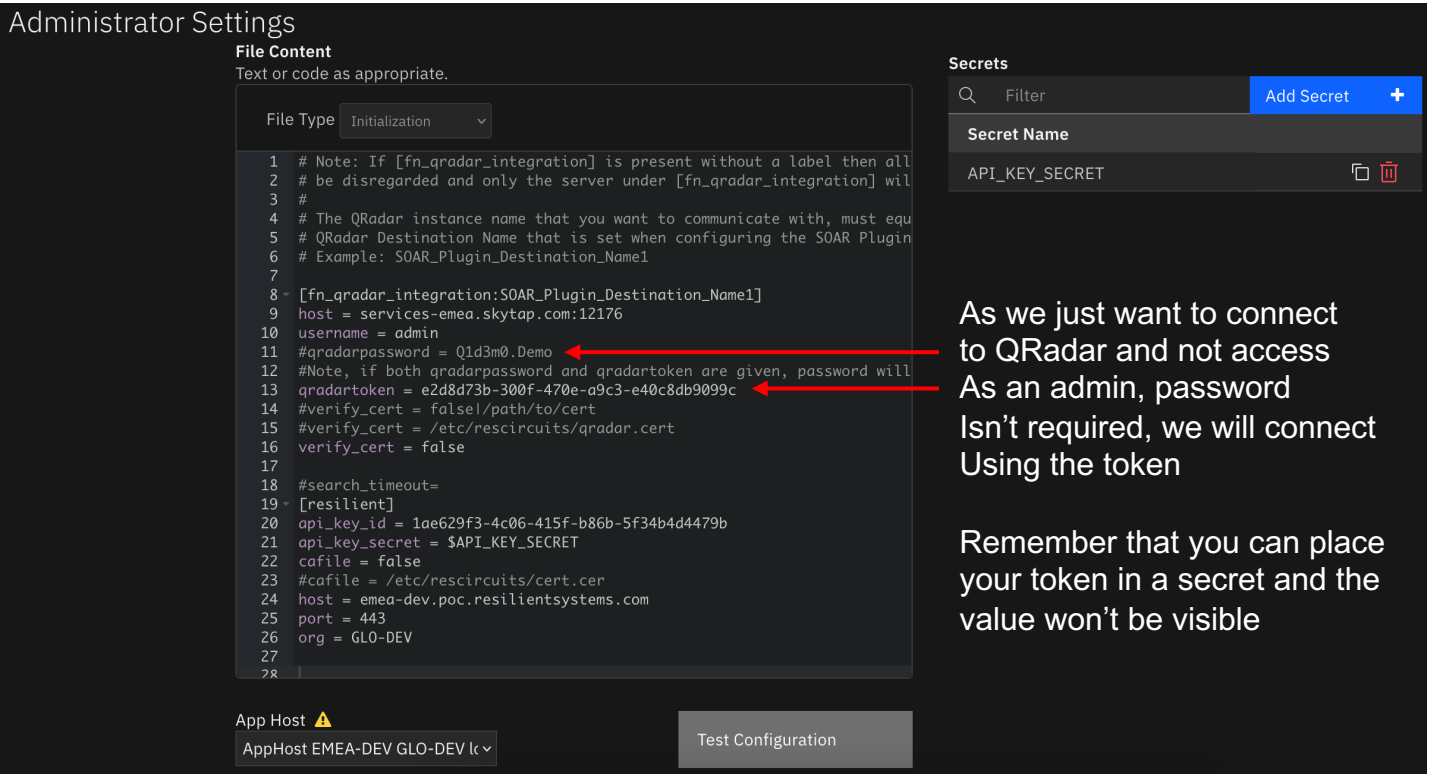


(Also available in your « lab QRadar » folder)



Configure QRadar Functions App

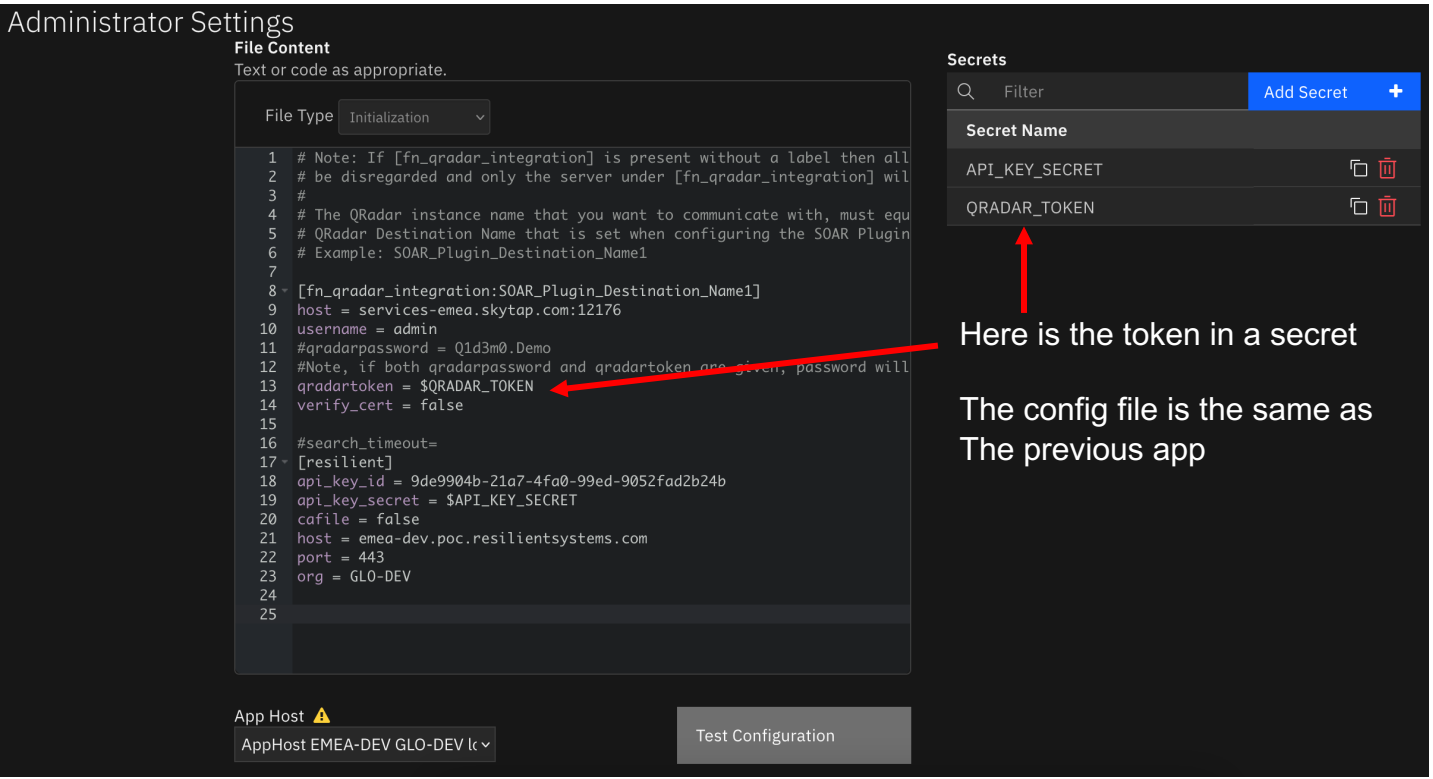
In the app tab, select QRadar Integration app and go to the app.config file
Change the credentials with the ones you were provided in skytap



When done test your Configuration, if the test is successful, save and push
Then deploy the app on your AppHost

Configure QRadar Enhanced Data Migration App

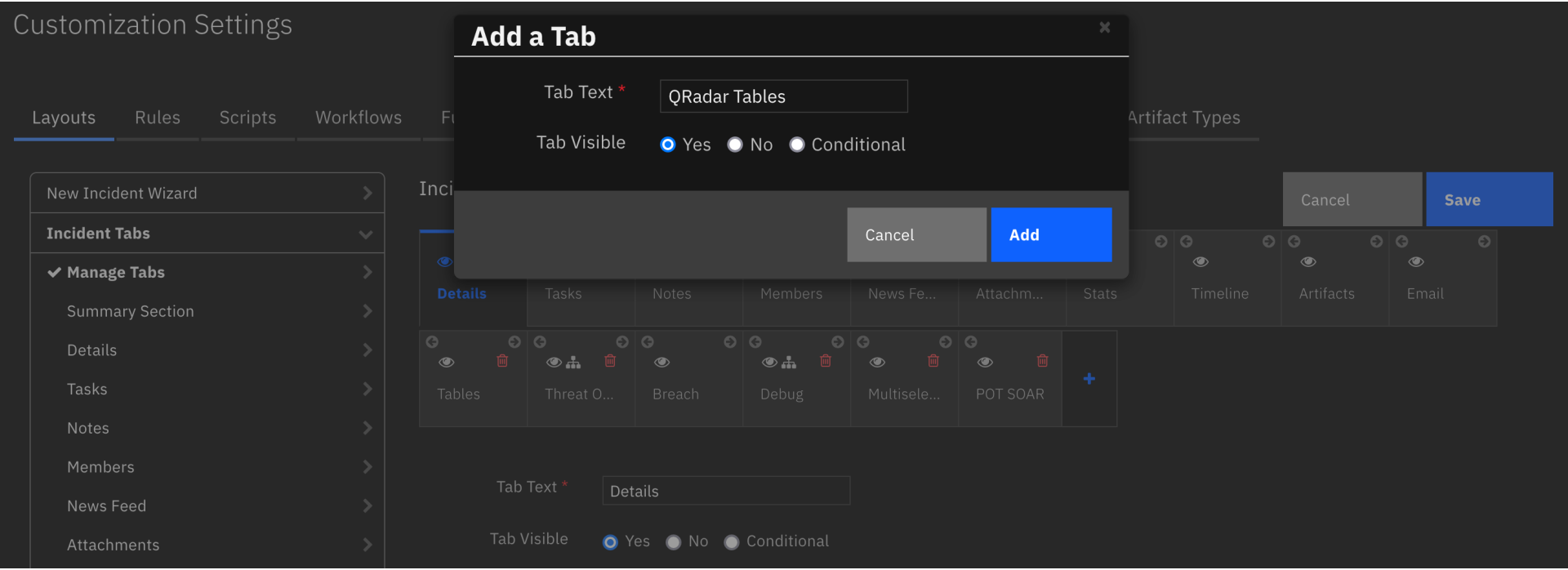
In the app tab, select QRadar Enhanced Data Migration app and go to the app.config file
Change the credentials with the ones you were provided in skytap



When done test your Configuration, if the test is successful, save and push Then deploy the app on your AppHost

Add data tables from app to incident layout

Go to your Customization Settings > Layout
Create a new tab



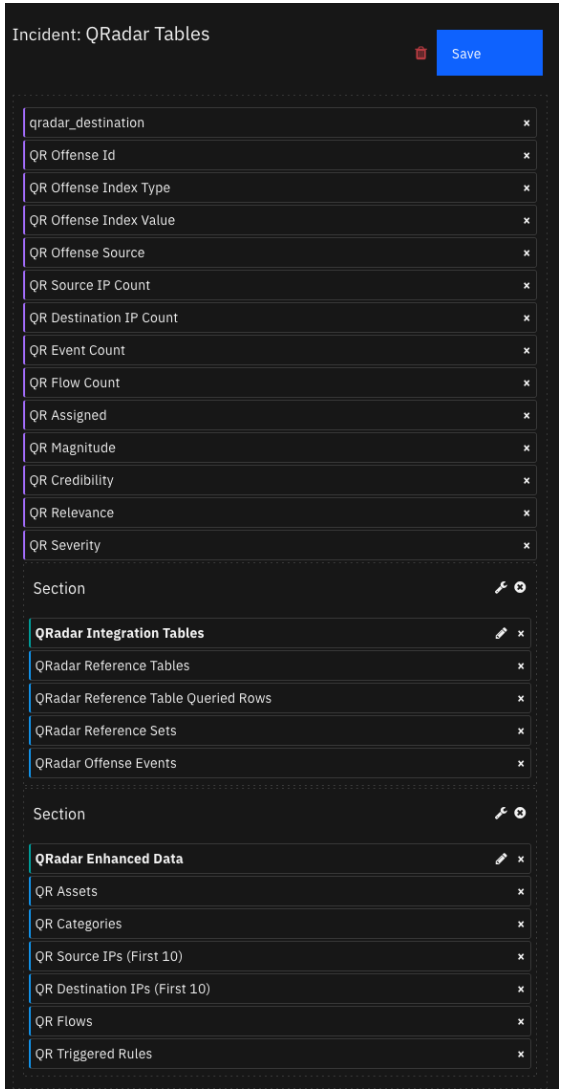
Add and Save !

Add data tables from app to incident layout

Go to the Custom tab Layout you just created and add tables in it:

Qradar_destination
QR Offense ID
QR Index Type
QR Offense Index Value
QR Offense Source
QR Source IP Count
QR Destination IP Count
QR Event Count
QR Flow Count
QR Assigned
QR Magnitude
QR Credibility
QR Relevance
QR Severity

- Section
  - Header « Qradar Integration Functions »
  - Datatable « QRadar Reference Tables »
  - Datatable « QRadar Reference Tables Queried Rows »
  - Datatable « QRadar Reference Sets »
  - Datatable « QRadar Offense Events »
- Section
  - Header « Qradar Integration Functions »
  - Datatable « QR Source IPs (First 10)
  - Datatable « QR Destination IPs (First 10)
  - Datatable « QR Assets »
  - Datatable « QR Categories »
  - Datatable « QR Flows »
  - Datatable « SR Triggered Rules »



QRADAR SOAR TRAINING

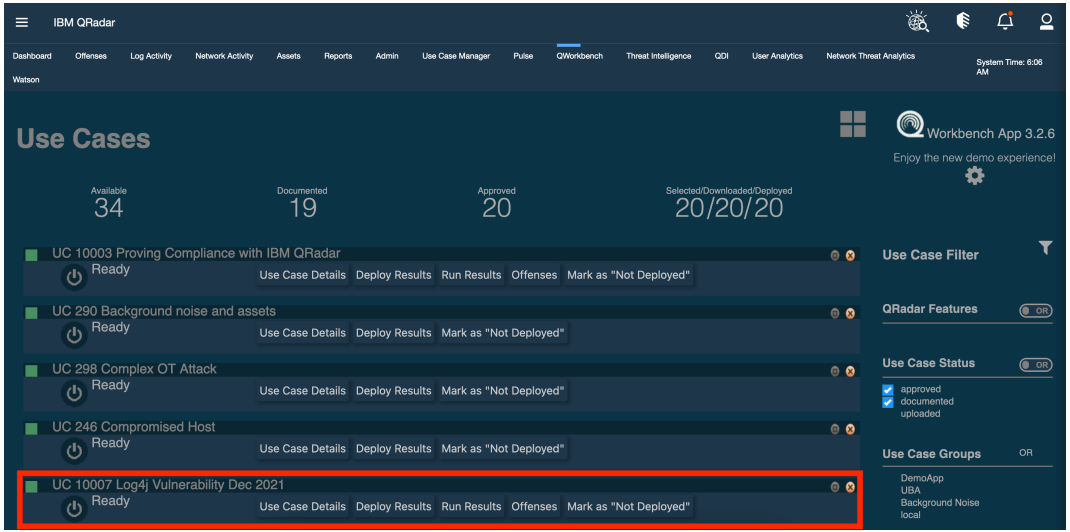
Lab 23: Create offense and see them in SOAR

Goal:

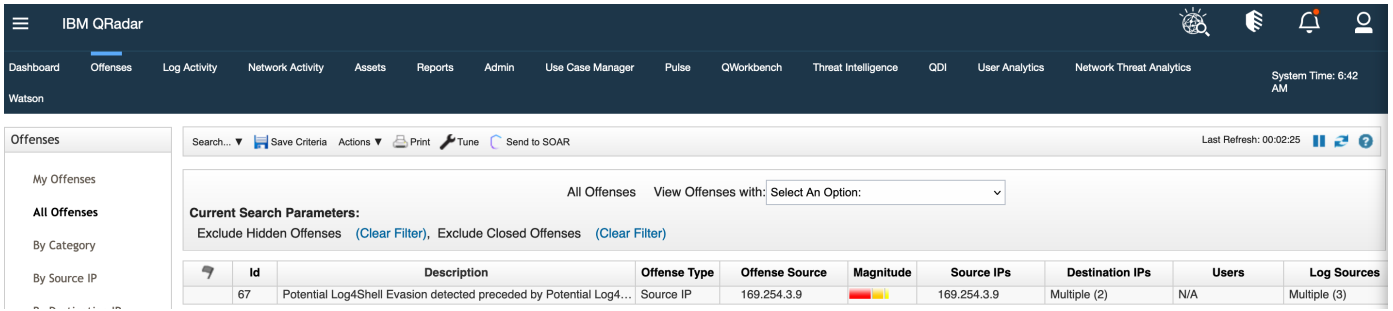
- Launch a use case and see how incidents are created and enriched from QRadar

Test Connection

To test connection, go to **Qworkbench tab** and launch « **UC 10007 Log4j Vulnerability Dec 2021** »

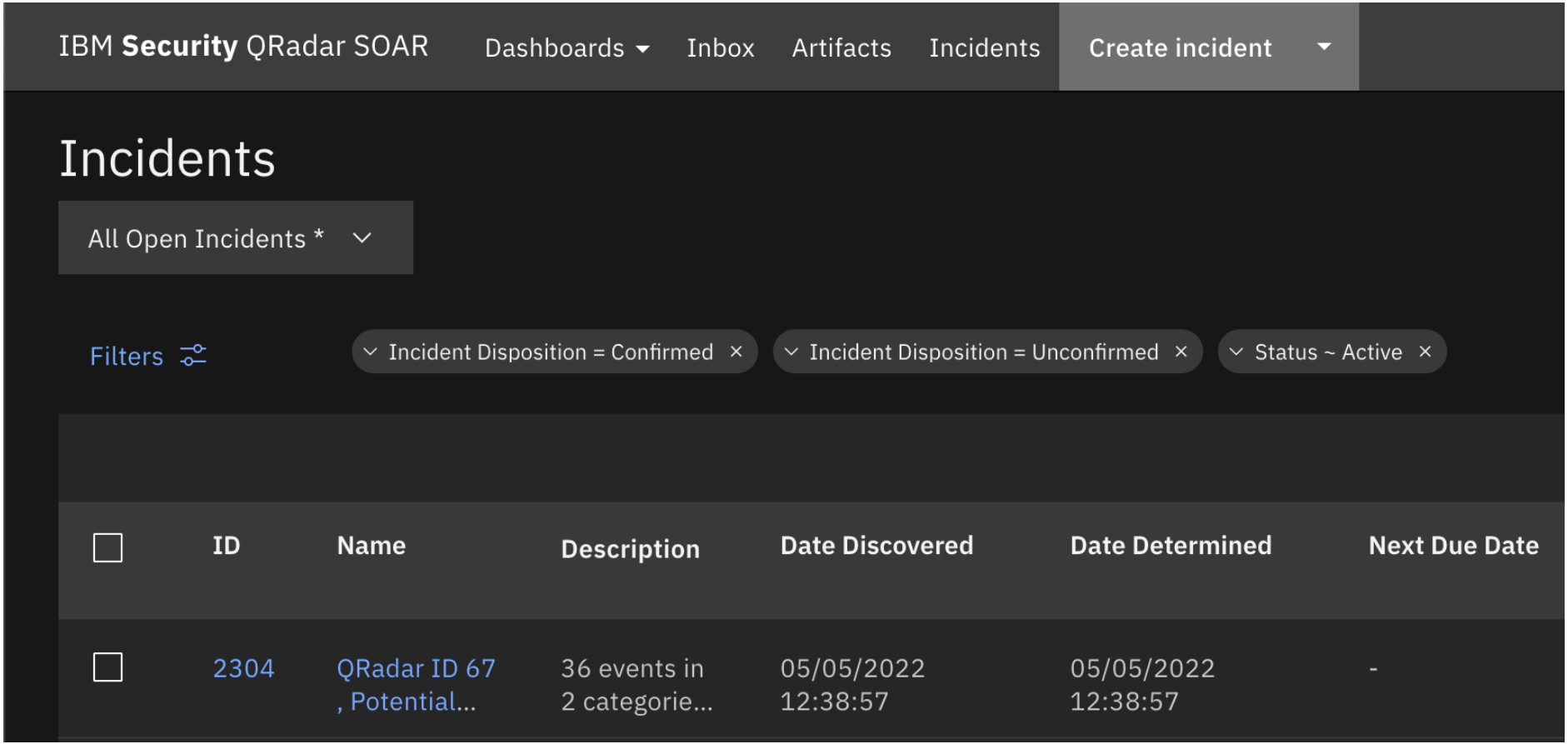


Wait a few minutes and it should appear in your QRadar Offense tab



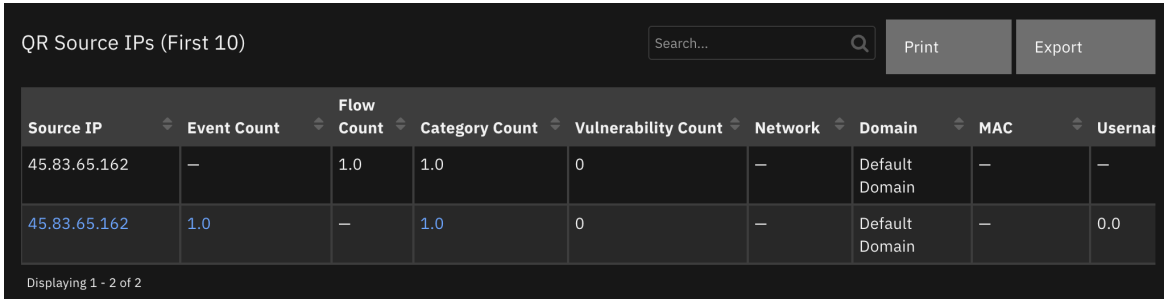
Test Connection

An incident should have been created in SOAR few minutes later



Test Connection

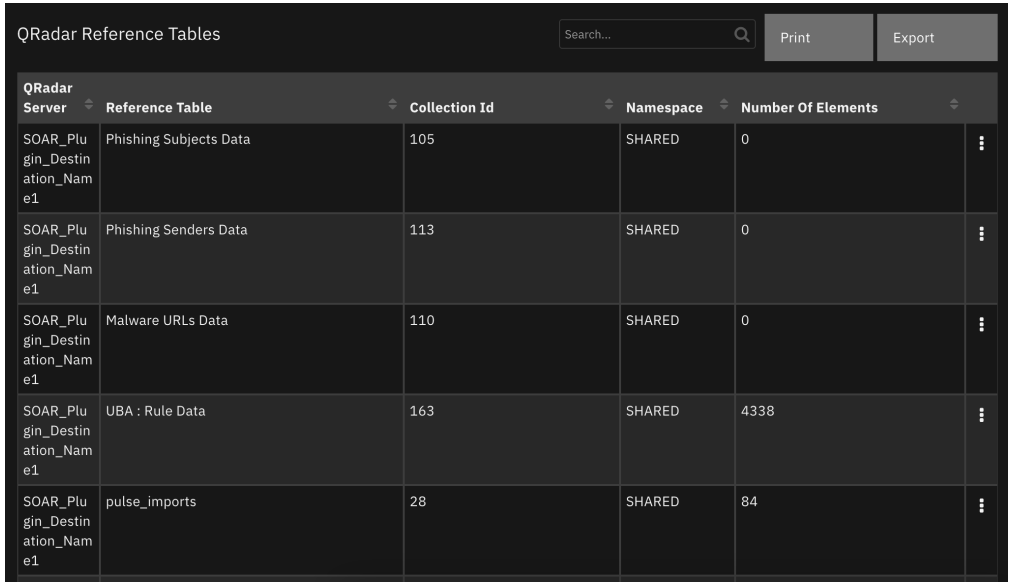
Short time later the tables from « QRadar Enhanced Data » should have some values



To fill the other tables you will have to launch the actions

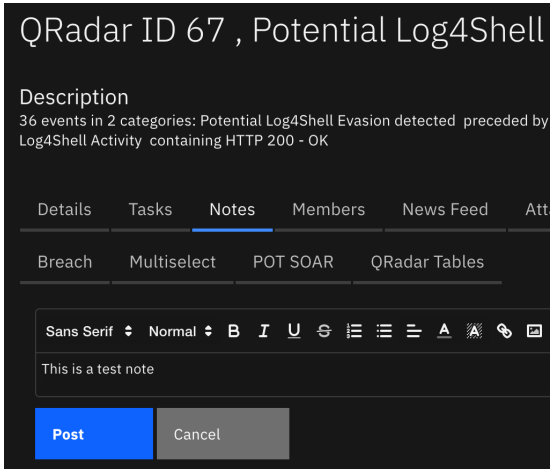
- Example: QRadar – Get all Reference Tables
- Search QRadar for Offense id

And refresh the page

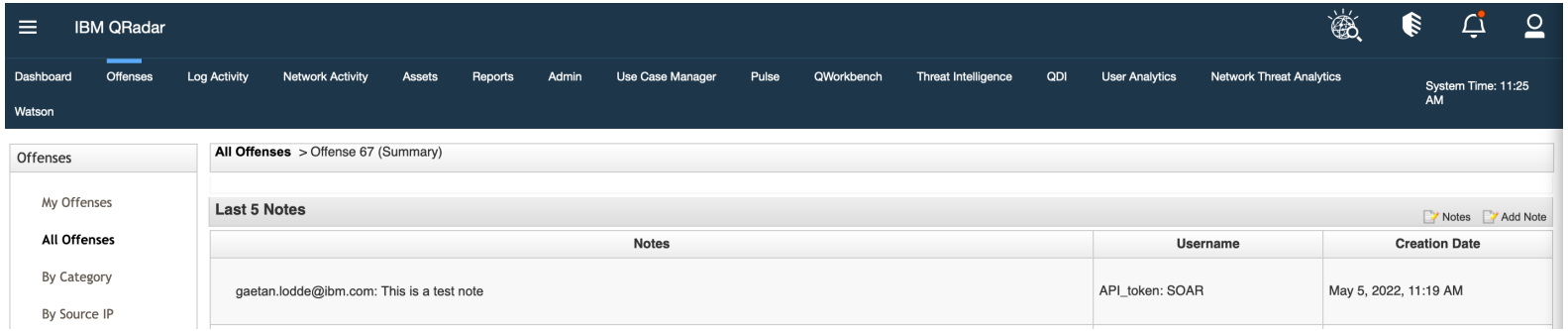


Add a note to SOAR and QRadar

Try adding a note to SOAR an post it

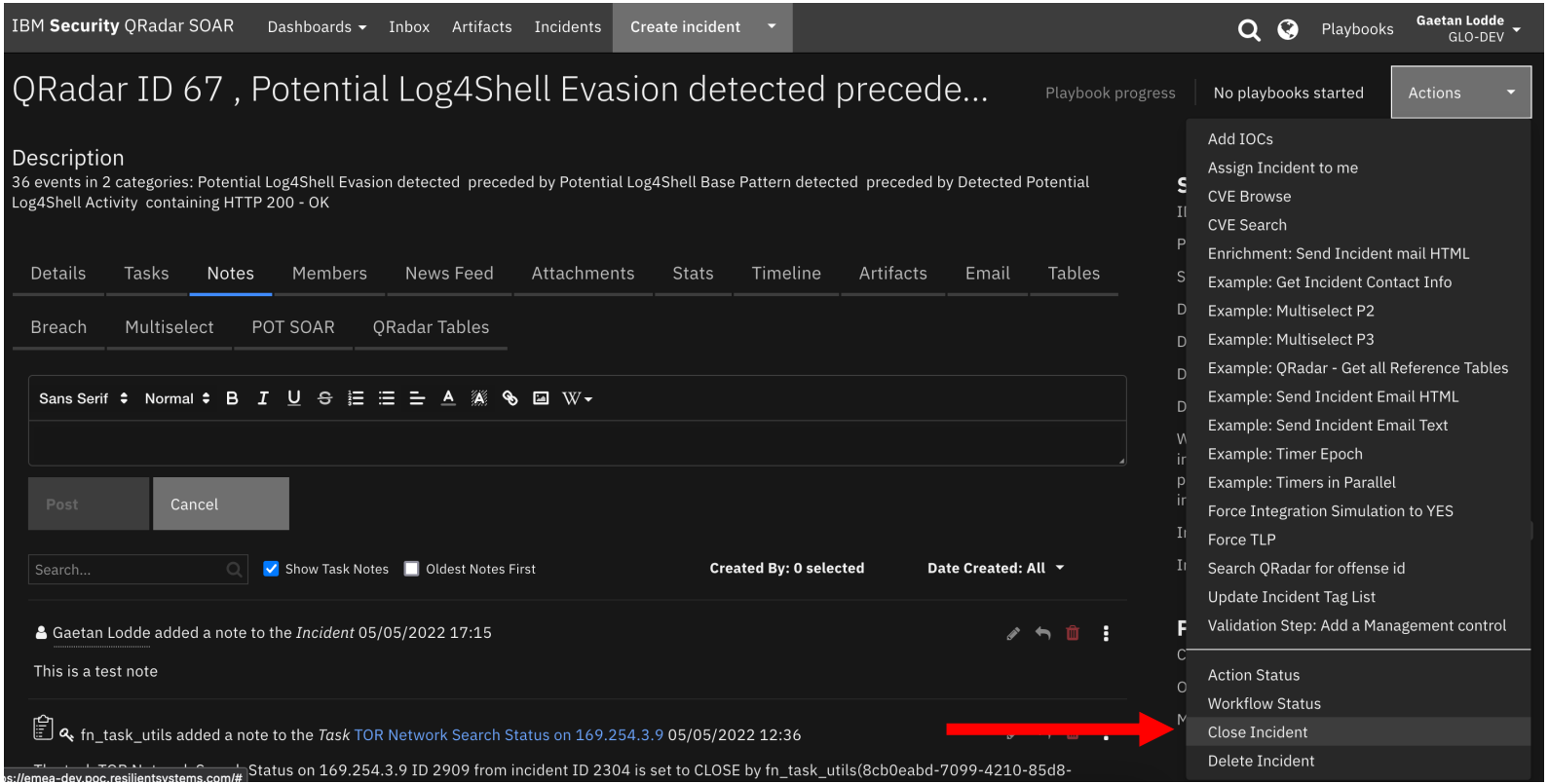


Then go to the offense with the same id in Qradar and open it
You will notice that the notes are shared between QRadar and SOAR

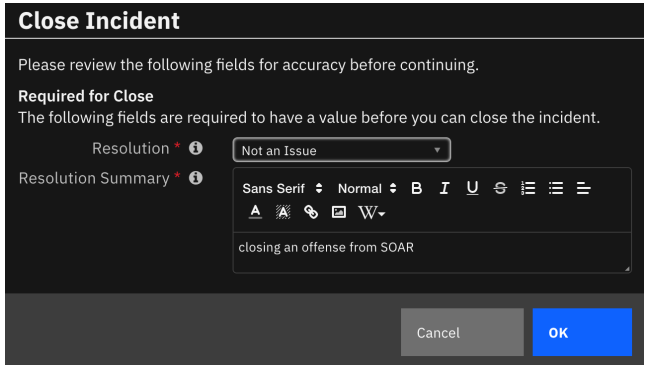


Close an Offense from SOAR

The same way, notice that closing an incident in SOAR will close the Offense related in QRadar



Close an Offense from SOAR



Offense with id 67 has been closed

All Offenses View Offenses with: Select An Option:

Current Search Parameters:

Exclude Hidden Offenses (Clear Filter), Exclude Closed Offenses (Clear Filter)

	Id	Description	Offense Type	Offense Source	Magnitude	Source IPs	Destination IPs	Users
	68	Detected Potential Log4Shell Activity containing Web.Web	Source IP	45.83.65.162		45.83.65.162	192.168.56.101	N/A
	69	Flow Source/Interface Stopped Sending Flows	Rule	Flow Source Stoppe…		Multiple (3)	192.168.56.101	N/A

Showing closed offense

	Id	Description	Offense Type	Offense Source	Magnitude	Source IPs	Destination IPs	Users	Log Sources
	67	Potential Log4Shell Evasion detected preceded by Potential Log4…	Source IP	169.254.3.9		169.254.3.9	Multiple (2)	N/A	Multiple (3)

IBM Security

THANK YOU

FOLLOW US ON:

- ibm.com/security
- securityintelligence.com
- xforce.ibmcloud.com
- @ibmsecurity
- youtube/user/ibmsecuritysolutions

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective.

IBM DOES NOT WARRANT THAT ANYSYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

IBM