# SOAR Lab

## Scenario:

ES has generated a finding that has been sent over to SOAR for additional contextualization and remediation. Utilizing the best practices and methods discussed in this course, follow the lab guide to build a SOAR playbook to automate the SOC's response to this finding.

This lab will operate entirely within the SOAR instance that has been assigned to you. All relevant actions have been implemented and finding containers to work from have been provided as well.

Note: Please ensure that you are testing against the **All Artifacts** scope when using the debugger.

## Task 2: Examine the finding Container

1. Log into your provided SOAR instance and just under the logo in the click on the drop down menu and select the ***Events*** option.
2. There will be 4 findings in SOAR containers, open any one of them.
3. In the upper right corner, toggle your SOAR view over to ***Analyst*** and then click on ***Artifacts*** tab.
4. Select the artifact named "ESCU - Suspicious Process DNS…" to expand it to view the full finding from Enterprise Security.
5. Examine the details of the finding, there are multiple fields that can be utilized within the scope of a playbook. Important fields such as:
    a. asset_channel - Slack channel for the owners of the asset
    b. destinationAddress - the short hostname for the asset
    c. domain - the destination DNS domain the asset requested
    d. instance_id - the AWS instance ID of the asset
    e. user - the username that made the request.
6. Keep this tab open for reference and then clone/replicate it to begin building the playbook.

## Task 3: Build the playbook

1. In the upper left hand corner, right click on the SOAR icon and open that in a new tab.
2. Using the dropdown menu, click on Playbooks.
3. In the upper right corner, click on the **+Playbook** button and in the popup modal select **Automation** for playbook type.

4. In your new playbook, name it **XX SOAR Workshop** (with XX being replaced with your initials) and expand the settings menu on the right hand side.
    a. *You may name the playbook whatever you want so long as you remember the name in the coming steps.*
5. Click on the blue Save button, it will pop up a menu on the right hand side where you will need to set some options:
    a. Under **Operates on** type in or select the * to allow it to operate on all tags.
6. Click on the **Save** button and in the modal popup type in any comment that you would like and click **Save**.
    a. *If you select the **Reuse this comment until the editor is reloaded** option, you will not need to add a comment every time you save.*

You are now ready to begin building your playbook, the following instructions will walk you through what needs to be done to set up your playbook.

***Initial Triage***

1. Off the **Start** action, drag and drop an action block and select the *AWS EC2* App and the *describe instance* action.
    a. The instance ID can be found in the *artifact:*.cef.instance_id* data path.
    b. Click on the **i** with a circle around it to go into the info tab and set the Custom Name to *ec2 describe instance*
2. Off the **Start** action, drag and drop an action block and select the Cisco Talos Intelligence App and the domain reputation action.
    a. The domain name can be found in the *artifact:*.cef.domain* data path.
    b. Click on the **i** with a circle around it to go into the info tab, set the Custom Name to *talos domain reputation*
3. Off the **Start** action, drag and drop an action block and select the *Virus Total v3* App and the *domain reputation* action.
    a. The domain name can be found in the *artifact:*.cef.domain* data path.
    b. Click on the **i** with a circle around it to go into the info tab, set the Custom Name to *vt domain reputation*

***Disposition Check***

4. Drag and drop in a decision block and connect it back to all blocks configured in the ***Initial Triage*** steps.
5. In the Info tab, set the Custom Name to *Disposition Check*
6. In the Configure tab, click the **+** symbol to add another data path block and have them operate on a boolean AND.
7. Examining the action results, the following examinations must be made:
    a. Is the Talos Threat Level Untrusted?
    talos_domain_reputation:action_result.data.*.Threat_Level == Untrusted
    b. Does the Virus Total domain reputation have greater than 5 malicious findings?
    vt_domain_reputation:action_result.summary.malicious > 5

8. Click the pencil icon next to the *Condition 1* name in the If statement and rename that condition to *Is Malicious* and click done.
9. Click on the + *Else* button to add an Else clause and then the pencil icon next to the *Condition 2* name. Rename this to *Is NOT Malicious* and click done.
10. Connect the *Disposition Check* decision block to the **End** block and on the line label select the *Is NOT Malicious* name and unselect the *Is Malicious* name.

***Check EC2 instance State***

11. Drag and drop in a decision block and connect it to the decision block just used in the ***Disposition Check*** and the *describe instance* action in ***Initial Triage***.
    a. Instance state can be found in the *describe_instance_1:action_result.data.*.Reservations.*.Instances.*.State.Name* datapath.
12. There are three outcomes to examine here based on the described EC2 instance state.
    a. If the EC2 instance is in a **terminated** state then this branch will connect to the **END** action block as no further action is needed.
    b. If the EC2 instance is in the **running** state then this branch will connect to the "**Notify and Shutdown**" section below.
    c. If the EC2 instance is in any state, other than **Terminated** or **Running**, then this branch will connect to the ***Quarantine EC2 Instance*** section below.

***Notify and Shutdown***

13. Drag and drop a *User Prompt* block in, connect it to the *Check EC2 Instance state* decision on the Running path.
14. Configure the user prompt as follows:
    a. Select **Others** for recipients and
    b. In the Distribution drop down, add a distribution method with the Slack asset to send a message.
        i. The destination can be found in the *artifact:*.cef.asset_channel* datapath
        ii. Select the *Formatted Input* option (denoted by the {0}) and fill it out with the message you wish to send to Slack.
            1. Example message: "EC2 instance {0} has been found to be communicating with a malicious domain. Please respond ASAP here: {1}"
            2. In the 0 field just after *message* select the instance ID datapath of *artifact:*.cef.instance_id*
            3. In the 1 field just after *message* select the instance ID datapath of *prompt1:action_result.parameter.secure_link*
    c. In the Introductory Message field, just replicate what you have put in the Slack message, without *Please respond…* section.
    d. Click on the **+ Question** and configure as follows:
        i. Question 1: "Can this host be shutdown?"
        ii. Response Type: Yes/No

15. Drag and drop a *decision block* in, connect it to the *User Prompt*, and the validation is going to have two paths:
    a. If the prompt is **positively** acknowledged (Yes) then proceed to the next step.
        i. Configure the If statement conditions, *prompt_1:action_result.summary.responses.0* == Yes
    b. If the prompt is **not positively** acknowledged (No/no/etc or no reply/timeout) then connect this path to the ***Quarantine EC2 instance*** section below.
16. Drag and drop an action block, connect it to the positively acknowledged path of the previous decision, and configure it with the *AWS EC2* app and the *snapshot instance* action.
    a. Volume ID can be found at *ec2_describe_instance:action_result.data.*.Reservations.*.Instances.*.BlockDeviceMappings.*.Ebs.VolumeId* datapath.
17. Drag and drop an action block, connect it with the previous *snapshot instance* action block, and configure it with the *AWS EC2* app and the *stop instance* action.
    a. Instance ID can be found at *artifact:*.cef.instance_id* datapath.
18. Connect the *stop instance* action block to **END**.

***Quarantine EC2 instance***

19. Drag and drop an action block from the decision block after the User Prompt and configure it with the *AWS EC2* App and the *list security group* action.
    a. You do NOT need to configure any of the settings for this action.
20. Drag and drop a filter block, connect this block with the previous *list security group* action, and have it filter the action results looking for **security_quarantine** as the name of the security group.
    a. Security Group names are found at *list_security_groups_1:action_result.data.*.SecurityGroups.*.GroupName* datapath.
21. Drag and drop an action block, connect this block with the previous filter, configure the AWS EC2 App and the *assign instance* action to the **security_quarantine** security group.
    a. Security Group ID is found at *filtered-data:filter_1:condition_1:list_security_groups_1:action_result.data.*.SecurityGroups.*.GroupId* datapath.
    b. Instance ID can be found at *artifact:*.cef.instance_id* datapath.
22. Drag and drop an action block, connect this with the *assign instance* action block, and configure the Splunk App and the *update event* action to update the finding in ES.
    a. Event_ids can be found at *artifact:*.cef.event_id* datapath.
    b. Leave a comment about the current status and what has been done

23. Drag and drop an action block, connect this with the *assign instance* action block, and configure the SNOW App and the *create ticket* action.
    a. The ticket will be a dummy, but using formatted inputs write out your reasoning behind the Quarantine.

# Task 4: Test and Run the playbook

1. Going back to your container in SOAR (found using the dropdown on the home screen and selecting ***Events***), look at the far left column in the container list and take note of all four of the container ID's.

# Task 5: Build the host examination playbook (Extra)

1. Create a new **input** playbook and name it **Windows Host Examination**
2. Off of Start drag and drop an action block, this block will be connected to the previous production tier decision block, and configure the Windows App to use the *list connections* action.
    a. ip_hostname can be found at *artifact:*.cef.dest*
3. Off of Start drag and drop an action block, this block will be connected to the previous production tier decision block, and configure the Windows App to use the *list processes* action.
    a. ip_hostname can be found at *artifact:*.cef.dest*
4. Off of Start drag and drop an action block, this block will be connected to the previous production tier decision block, and configure the Windows App to use the *list sessions* action.
    a. ip_hostname can be found at *artifact:*.cef.dest*
5. In the main automation playbook, during the **Check EC2 instance State** decision block, fork the **running** path to call a new decision block
6. It should examine if the production tier is **prod**. *ec2_describe_instance:action_result.data.*.Reservations.*.Instances.*.Tags.environment*
    a. If true, then continue.
    b. If false, then connect the block to End.
7. Call the **Windows Host Examination** playbook with the Input of dest being at the *artifact.*.cef.dest* datapath.