paloalto NETWORKS

SecOps Automation and Response: Cortex XSOAR Phishing Investigation

OPERATIONS GUIDE

AUGUST 2021

Table of Contents

Preface

GUIDE TYPES

Overview guides provide high-level introductions to technologies or concepts.

Reference architecture guides provide an architectural overview for using Palo Alto Networks® technologies to provide visibility, control, and protection to applications built in a specific environment. These guides are required reading prior to using their companion deployment guides.

Deployment guides provide decision criteria for deployment scenarios, as well as procedures for combining Palo Alto Networks technologies with third-party technologies in an integrated design.

DOCUMENT CONVENTIONS

Notes provide additional information.

Cautions warn about possible data loss, hardware damage, or compromise of security.

Blue text indicates a configuration variable for which you need to substitute the correct value for your environment.

In the **IP** box, enter `10.5.0.4/24`, and then click **OK**.

Bold text denotes:

- Command-line commands.

  ```
  # show device-group branch-offices
  ```

- User-interface elements.

  In the **Interface Type** list, choose **Layer 3**.

- Navigational paths.

  Navigate to **Network** > **Virtual Routers**.

- A value to be entered.

  Enter the password **admin**.

Italic text denotes the introduction of important terminology.

> An *external dynamic list* is a file hosted on an external web server so that the firewall can import objects.

Highlighted text denotes emphasis.

> Total valid entries: 755

ABOUT PROCEDURES

These guides sometimes describe other companies' products. Although steps and screen-shots were up-to-date at the time of publication, those companies might have since changed their user interface, processes, or requirements.

GETTING THE LATEST VERSION OF GUIDES

We continually update reference architecture and deployment guides. You can access the latest version of this and all guides at this location:

https://www.paloaltonetworks.com/referencearchitectures

WHAT'S NEW IN THIS RELEASE

Palo Alto Networks made the following changes since the last version of this guide:

- Changed the version of Cortex™ XSOAR to version 6.2.0
- Changed management of built-in incident types
- Changed phrasing, terminology, and diagrams for clarity

Related Guides

Cortex XSOAR is a security orchestration, automation, and response (SOAR) solution that manages alerts, standardizes processes, and automates responses.

The *SecOps Automation and Response—Cortex XSOAR* suite of guides details how to use Cortex XSOAR, from understanding its concepts and user interface through deployment and using playbooks to implement a structured and automated incident response.

SecOps: Reference Architecture Guide—Provides solutions for prevention, detection, investigation, and response to help security-operations teams prevent threats and efficiently manage alerts.

SecOps Automation and Response—Cortex XSOAR: Concepts Guide—Describes concepts and terminology essential to using Cortex XSOAR in order to automate responses to security incidents.

SecOps Automation and Response—Cortex XSOAR: User Interface Guide—Describes user interface components that are important when you use the operations guides.

SecOps Automation and Response—Cortex XSOAR: Deployment Guide—Provides detailed, step-by-step instructions for deploying Cortex XSOAR, including post-installation tasks such as the required integrations to external systems.

SecOps Automation and Response—Cortex XSOAR Phishing Investigation: Operations Guide—Suggests a method for automatically investigating and responding to an email-based phishing incident.

OTHER RESOURCES

Cortex XSOAR developer hub (https://xsoar.pan.dev)—Includes documentation and reference materials about all Cortex XSOAR components.

Cortex XSOAR Administrator's Guide—Serves as a comprehensive product reference and includes information about the numerous supported methods for installing Cortex XSOAR.

Purpose of This Guide

This guide describes an automation scenario that you can use to respond to an email-based phishing attack. The scenario is based on a manual process and uses Cortex XSOAR in order to automate the analysis and response to the attack.

This deployment guide:

- Requires that you first read the **SecOps Automation and Response—Cortex XSOAR: Concepts Guide**. This guide introduces you to concepts and terminology essential to using Cortex XSOAR for automating the response to security incidents.
- Requires that you first deploy Cortex XSOAR as described in the **SecOps Automation and Response—Cortex XSOAR: Deployment Guide**. The deployment guide provides step-by-step deployment details for deploying Cortex XSOAR, as well as post-installation tasks such as the required integrations to external systems.
- Provides step-by-step configuration details for how to build and run a Cortex XSOAR playbook for responding to an email-based phishing attack. This guide separates the playbook configuration process into multiple stages. In the first stage, you build a basic playbook that automatically analyzes the phishing email. In subsequent stages, you can enhance the basic playbook with automated response actions.
- Provides step-by-step configuration details for integrating Cortex XSOAR with a Microsoft 365 email system. This guide includes the configuration steps for assigning roles and permissions to the user account that Cortex XSOAR uses to access the email system, retrieve email messages from a monitored mailbox, and delete malicious email messages.
- Provides step-by-step configuration details for integrating Cortex XSOAR with various Palo Alto Networks services and platforms, including AutoFocus™, WildFire®, and Cortex Data Lake. You use these integrations within the playbook in order to perform automation scripts and commands as part of the analysis and response to the phishing attack.

OBJECTIVES

Completing the procedures in this guide, you can successfully build and run a Cortex XSOAR playbook to analyze and respond to an email-based phishing attack. The main objectives are to enable the following functionality:

- Building and running a Cortex XSOAR playbook
- Integrating Cortex XSOAR with platforms and services from Palo Alto Networks
- Integrating Cortex XSOAR with Microsoft 365

Another primary objective of this guide is to teach the generic concepts of how to properly use Cortex XSOAR automation scripts and commands and to successfully access and manipulate context data within an incident. You can then use these concepts in order to independently develop new playbooks or enhance existing playbooks.

AUDIENCE

This deployment guide is for technical readers, including solution architects, security engineers, and security support staff, who want to orchestrate and automate the prevention, investigation, and response to security threats. It assumes the reader is familiar with the basic concepts of threat prevention, networking, and security operations, as well as possessing a basic understanding of automation, machine learning, and analytics. Additionally, this guide assumes the reader is familiar with the basic concepts of JSON, flowcharts, conditional statements, and Boolean logic.

Automation Scenarios

There are many ways to use the concepts discussed in **SecOps Automation and Response–Cortex XSOAR: Concepts Guide** in order to identify automation scenarios and develop response playbooks for common security use-cases. Each of the automation scenarios in that guide provides an example security use-case and a description of the process used by a SecOps team when responding to the use case.

GOALS FOR AN AUTOMATION SCENARIO

When considering any automation scenario, you should ask the following questions:

- **Automating mundane tasks**—How cumbersome is the current manual process? How important is manual intervention? How much can be automated? Are current staffing levels sufficient for completing tasks in a reasonable timeframe? Are security analysts using their time efficiently? Are security analysts able to use their advanced skills or are they just performing rudimentary data collection?
- **Responding rapidly**—How critical is it that the organization can respond quickly? How much time can you save through automation? Are responses currently delayed due to lack of resources? Does the current manual process meet the required service-level agreement?
- **Integrating across multiple components**—How many different systems are involved? What level of expertise do you require to master individual systems? Do security analysts have the appropriate permissions to access the required systems? Is the data properly formatted and compatible across systems?
- **Reducing errors**—Is the current manual process error prone? Are processes followed consistently? Is all data accessible within a single system?

AUTOMATED PHISHING INVESTIGATION AND RESPONSE AUTOMATION SCENARIO

In this scenario, your SecOps team has set up a phishing mailbox on the email system, which they monitor using Cortex XSOAR, and they request that users forward all suspected phishing messages to that mailbox. A user has forwarded one such email.

You had configured Cortex XSOAR to integrate with the email system and periodically check for emails to the monitored mailbox. When the new email arrives, Cortex XSOAR retrieves it as an event and creates an incident. As part of the incident response, Cortex XSOAR automatically executes a playbook to analyze the email and, optionally, to automatically respond if the email contains phishing or malware content.

Cortex XSOAR starts the basic analysis by retrieving the original message that the end-user forwarded. In addition to extracting the email headers, which include domain and IP address indicators, the analysis extracts URL indicators. To determine if unknown URL indicators are malicious, Cortex XSOAR submits them to WildFire.

When the analysis completes, before taking any further action, the assigned SecOps analyst uses Cortex XSOAR in order to review the attack information that Cortex XSOAR has automatically gathered.

Automatic Response Options

If you want Cortex XSOAR to take actions based on the results, this automation scenario includes some advanced options. The first option is to search the email system for all recipients of the original message and, if the analysis determines that the message contained a malicious phishing URL, remove the message from all mailboxes systemwide. The second option assumes that users at your organization connect to the network through Prisma™ Access. Cortex XSOAR searches the Prisma Access logs stored in Cortex Data Lake to determine if any users tried to access the malicious URL in the original message. If the logs confirm that users did try and connect, then you can notify the users. Although it is beyond the scope of this guide, you could use this information to create new incidents to interact with users who might need further attention, such as attending a training session focused on phishing awareness or initiating a forensic examination of their computer system.

Assumptions and Prerequisites

This guide assumes:

Cortex XSOAR requirements:

- Your organization has already deployed Cortex XSOAR and completed the post-installation tasks as outlined in the **SecOps Automation and Response–Cortex XSOAR: Deployment Guide**.
- The tested Cortex XSOAR software version in this operations guide is 6.2.0.

Email system requirements:

- Your organization uses Microsoft 365 email and has an active subscription.
- For the phishing mailbox, you have created a user account in the Microsoft 365 email system. This user may be an unlicensed account.
- You have a user account for Cortex XSOAR on the Microsoft 365 email system. This user account needs specific roles and permissions to execute automation commands.
- To assign these role and permissions for the Cortex XSOAR account requires access to a user account with the roles of Microsoft 365 global administrator and Exchange administrator.

Palo Alto Networks software and licensing requirements:

- Your organization has an active WildFire license.
- Your organization has an active AutoFocus license.

For additional analysis and response capabilities, this guide also assumes:

- Remote users use the GlobalProtect™ app to connect to Prisma Access in order to access internal and external resources.
- You have already created security policy rules for Prisma Access, which include a logging profile that sends logs to Cortex Data Lake.

Panorama™ and Prisma Access requirements:

- You have an existing Panorama management system in operation that you can use to provision and control Prisma Access.
- The tested Panorama PAN-OS® version used in this guide is 10.0.2 with the Cloud Services plugin version 1.8.0.
- You have purchased and activated a Prisma Access for Users license.
- You have an existing Cortex Data Lake instance operational and associated to the Panorama system.

Cortex XSOAR Automation Scenario: Automated Phishing Investigation

This section covers the complete workflow required for implementing an automated process for investigating and responding to a phishing attack. In this scenario, any email subscriber within your organization can trigger an incident by forwarding a suspected phishing email to a monitored mailbox, such as phishing@example.com. After creating an incident, a Cortex XSOAR playbook performs an automated investigation and response.

As you work through this operations guide, you do the following required tasks:

- Complete the prerequisite steps for building your playbook, which includes the installation of content packs and configuring integrations with the external systems that you later use in your playbook.
- Build a basic playbook, which you use to create an incident and then perform an analysis of the suspected phishing email. In this basic playbook, you retrieve the email from your Microsoft 365 email system, access reputation data for the content from AutoFocus, and submit any URLs contained in the message body to WildFire for malware dynamic analysis.
- Configure Cortex XSOAR to automatically run the playbook.
- Review the results of the investigation that the playbook captures throughout the incident.

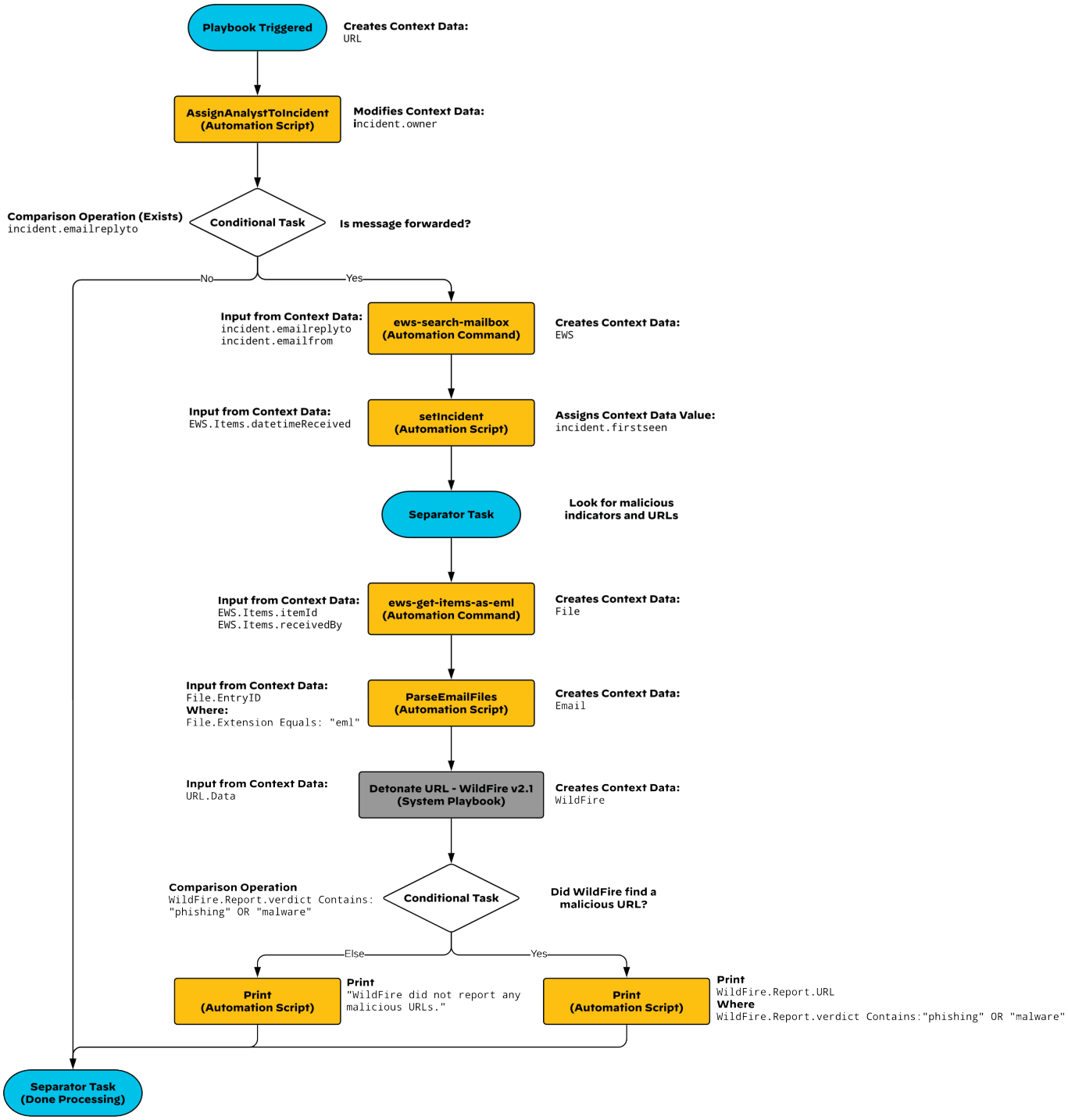
This guide also includes a variety of optional tasks that further demonstrate the capabilities of Cortex XSOAR. These optional tasks enhance the basic playbook:

- You can modify Cortex XSOAR to add custom incident fields that you use to summarize phishing information across multiple incidents. You then append tasks to the basic playbook, tasks that extract incident-specific data and populate the custom fields.
- You can create a custom report by using custom widgets that summarize information collected across all phishing incidents.
- You can modify your basic playbook to add a response action. In this option, if you have confirmed that the phishing email contains malicious content, your playbook searches the Microsoft 365 email system to identify users on the system that have received copies of the phishing email. If the case owner approves, the playbook deletes the email message systemwide.
- You can further modify the playbook to determine if any users tried to access any malicious content contained in the email. This option assumes that your organization uses Prisma Access for mobile user access. The playbook queries Cortex Data Lake for security-log entries that match malicious URLs in the email message. If Cortex Data Lake returns any matches, the playbook sends emails to notify affected users. When necessary, to limit the size and complexity of the parent playbook, you create sub-playbooks to group sets of related tasks or to enable advanced capabilities, such as looping.

PREPARING FOR, BUILDING, AND RUNNING THE BASIC PLAYBOOK

In this section, you complete the required tasks for the basic playbook. This section also includes the required tasks to build and generate custom reports.

Figure 1 Automated Phishing Investigation playbook (basic)



Procedures

Installing Content Packs and Configuring Integrations

In this section, you install and update content packs from the Cortex XSOAR Marketplace and then configure integrations to external systems.

1.1 Install or Update Content Packs

Before you can configure integration instances, you must first install the required content packs from the Cortex XSOAR Marketplace.

Some of the content packs come preinstalled with Cortex XSOAR. You may have installed a content pack while using the **SecOps Automation and Response Deployment Guide for Cortex XSOAR** or a different operations guide for Cortex XSOAR.

To cover all cases, this guide describes the process for installing content packs for the first time. If you have already installed the content pack, you update the content pack to the latest version.

Table 1 Cortex XSOAR content packs used in this guide

Content pack	Preinstalled	Integrations used	Playbook
AutoFocus	Yes	Palo Alto Networks AutoFocus v2	Automated Phishing Investigation
Cortex Data Lake	No	Cortex Data Lake	Check CDL Logs for URL
Exchange Web Services (EWS)	No	EWS v2	Automated Phishing Investigation
EWS Mail Sender	No	EWS Mail Sender	Email to List
Palo Alto Networks WildFire	No	Palo Alto Networks WildFire v2	Automated Phishing Investigation

Step 1: Log in to the Cortex XSOAR portal (example: https://xsoar.example.com).

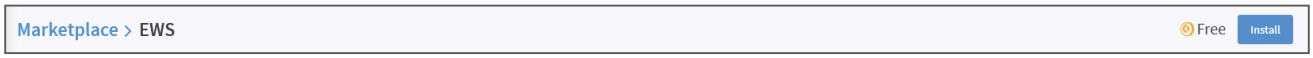
Step 2: In the navigation pane, click **Marketplace**.

Step 3: On the Browse tab, in the **Search in** list, choose **Content Packs**.

Step 4: In the search box, enter **EWS**.

Step 5: In the results pane, click **EWS**.

Step 6: If Cortex XSOAR does not have the content pack installed, click **Install**.



Step 7: If Cortex XSOAR has the content pack installed and there is an update, click **Update to**.

If Cortex XSOAR has the content pack installed and there are no updates available, then take no action.

If you select a content pack for installation or update, Cortex XSOAR adds it to the cart. If there are any additional required content packs that have updates available, Cortex XSOAR automatically adds them to your cart.

Step 8: In the cart pane, click **Update**.

Step 9: After Cortex XSOAR successfully updates a content pack, click **Refresh content**.

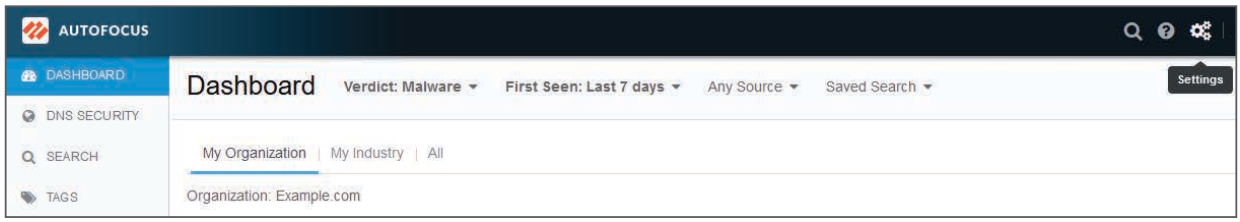
Step 10: Repeat Step 2 through Step 9 for any remaining content packs in Table 1.

1.2 Configure AutoFocus v2 Integration Instance

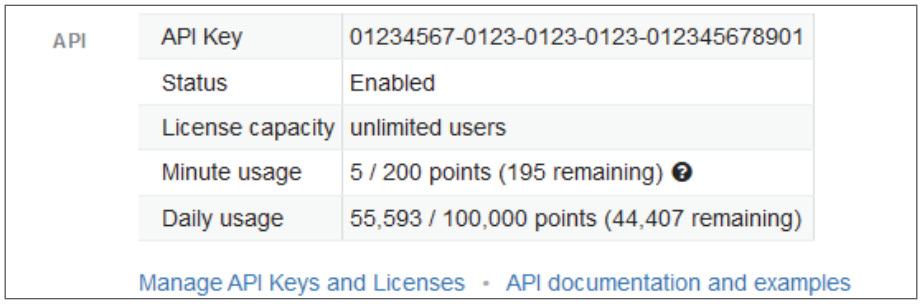
This procedure requires that your organization has an active AutoFocus subscription. As an active subscriber, you are entitled to access the AutoFocus API.

Step 1: Log in to the AutoFocus portal at **https://autofocus.paloaltonetworks.com**.

Step 2: Click **Settings**.



Step 3: To view the AutoFocus API key for your organization, scroll to the bottom of the Settings page, and then record the value of the API key.



Step 4: Log in to the Cortex XSOAR portal (example: https://xsoar.example.com).

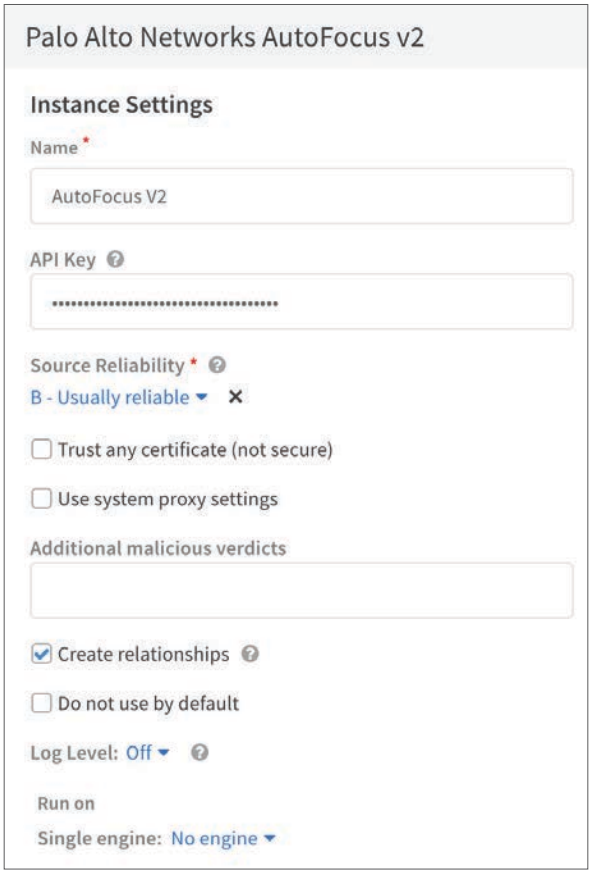
Step 5: In the navigation pane, click **Settings**.

Step 6: In **Integrations** > **Servers & Services**, in the search box, enter **Palo Alto Networks AutoFocus v2**.

Step 7: Click **Add instance**.

Step 8: In the **Name** box, enter **AutoFocus V2**.

Step 9: In the **API Key** box, enter the API key, and then click **Save & exit**.



1.3 Configure WildFire v2 Integration Instance

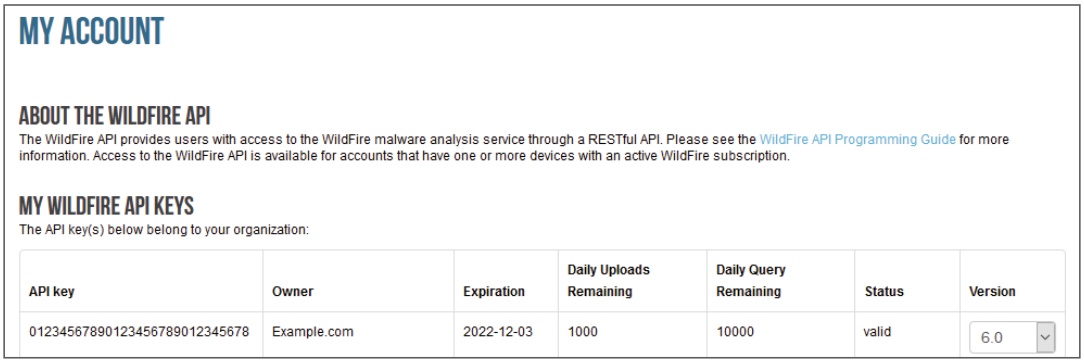
This procedure assumes that in Procedure 1.1, you installed the WildFire content pack from the Cortex XSOAR Marketplace.

If your organization has one or more devices with an active WildFire subscription, you are entitled to access the WildFire API.

Step 1: Log in to the WildFire portal at **https://wildfire.paloaltonetworks.com**.

Step 2: To view the WildFire API keys for your organization, click **Account**.

Step 3: Your organization may have multiple API keys. Select an API key with a status of *valid*. Record the value of the API key.



MY ACCOUNT

ABOUT THE WILDFIRE API

The WildFire API provides users with access to the WildFire malware analysis service through a RESTful API. Please see the WildFire API Programming Guide for more information. Access to the WildFire API is available for accounts that have one or more devices with an active WildFire subscription.

MY WILDFIRE API KEYS

The API key(s) below belong to your organization:

API key	Owner	Expiration	Daily Uploads Remaining	Daily Query Remaining	Status	Version
01234567890123456789012345678	Example.com	2022-12-03	1000	10000	valid	6.0

Step 4: Log in to the Cortex XSOAR portal (example: https://xsoar.example.com).

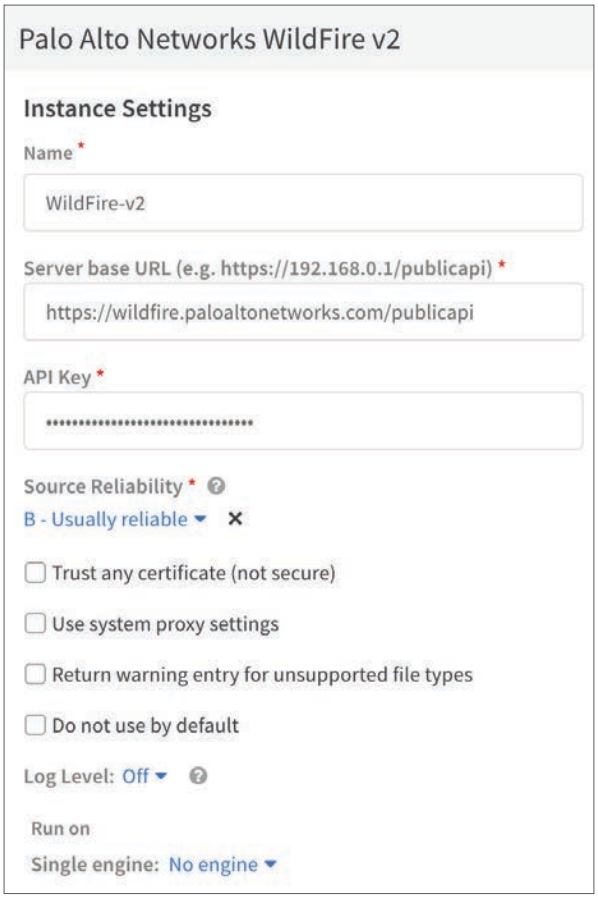
Step 5: In the navigation pane, click **Settings**.

Step 6: In **Integrations > Servers & Services**, in the search box, enter **Palo Alto Networks WildFire v2**.

Step 7: Click **Add instance**.

Step 8: In the **Name** box, enter **WildFire-v2**.

Step 9: In the **API Key** box, enter the API key, and then click **Save & exit**.



1.4 Configure Microsoft 365 System Permissions for EWS Integration

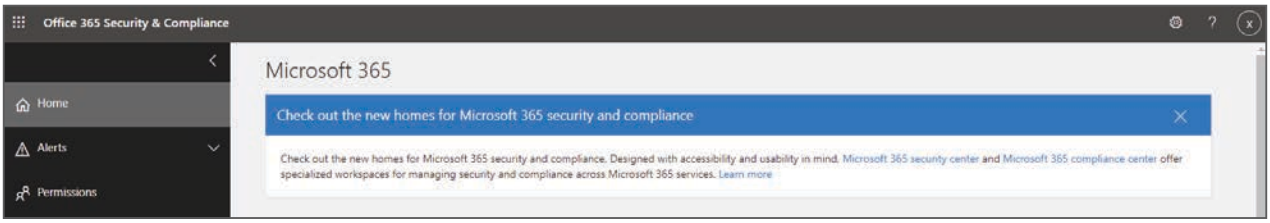
You use a Cortex XSOAR user account on your Microsoft 365 email system in order to retrieve emails from the monitored mailbox and perform searches and other operations on the email system.

The following procedures use xsoar@example.com as the Cortex XSOAR user account.

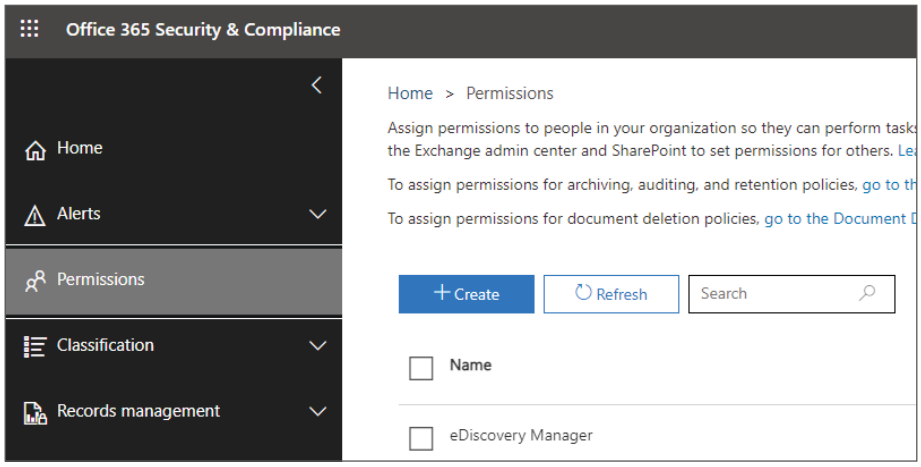
In this procedure, you assign permissions for the user xsoar@example.com.

Assigning these permissions for the Cortex XSOAR account requires access to a user account that you have assigned the role of Microsoft 365 global administrator and Exchange administrator.

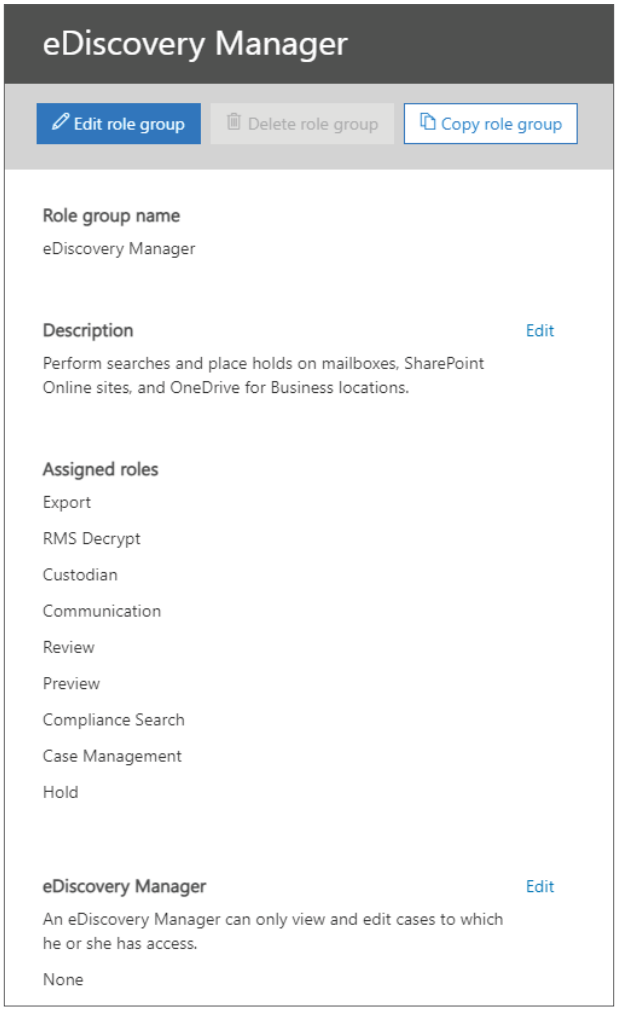
Step 1: Log in to the Microsoft 365 Security & Compliance Center at **https://protection.office.com**.



Step 2: In the navigation pane at left, click **Permissions**.

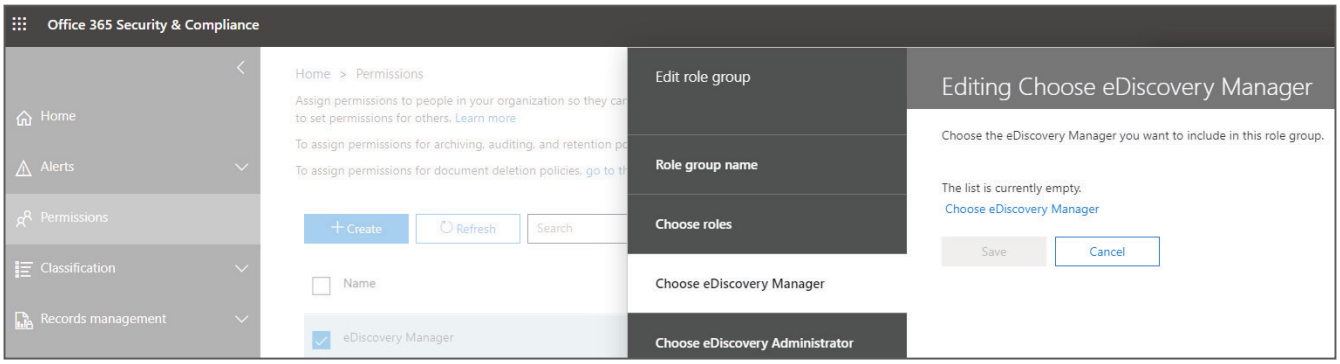


Step 3: In **Home > Permissions**, select **eDiscovery Manager**.

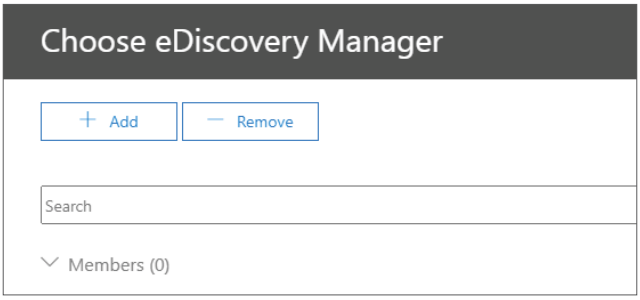


Step 4: In the row for eDiscovery Manager, click **Edit**.

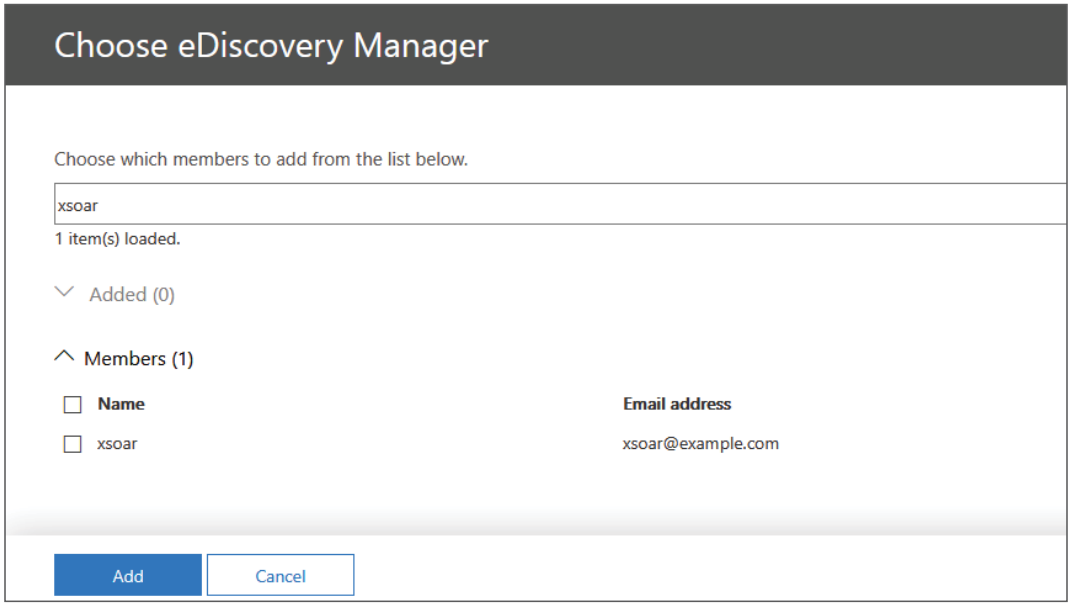
Step 5: On the Editing Choose eDiscovery Manager dialog box, click **Choose eDiscovery Manager**.



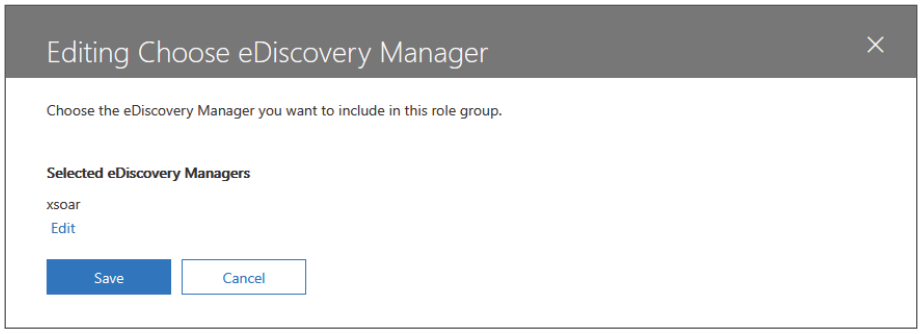
Step 6: On the Choose eDiscovery Manager page, click **Add**.



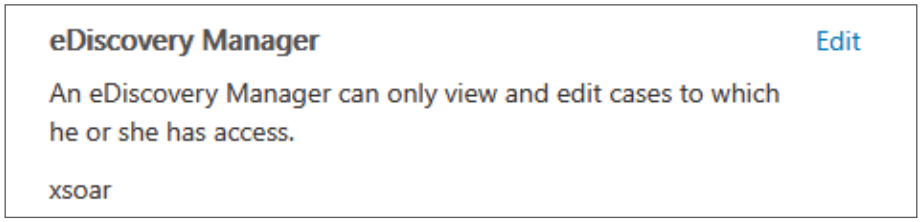
Step 7: In the **Members** list, select **xsoar**, click **Add**, and then click **Done**.



Step 8: Click **Save**.



Step 9: Verify that the **xsoar** user has been granted the eDiscovery Manager permission.



Step 10: Click **Close**.

1.5 Configure Microsoft 365 System Roles for EWS Integration

In this procedure, you create a new role group for Application Impersonation. You then assign the user xsoar@example.com to the new role group. You also assign the user xsoar@example.com to the existing Discovery Management role group.

As a result, assigning these roles grants these permissions to the user:

- ApplicationImpersonation (Application Impersonation role group)
- Legal Hold (Discovery Management role group)
- Mailbox Search (Discovery Management role group)

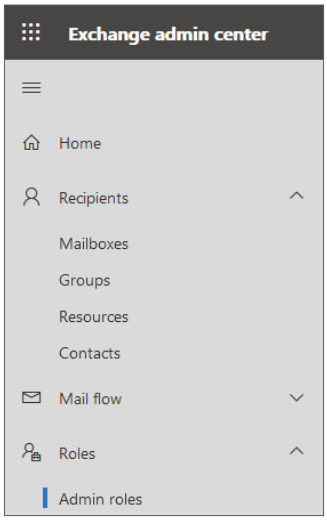
Assigning these roles for the Cortex XSOAR account requires access to a user account that is assigned the role of Microsoft 365 global administrator and Exchange administrator.

Note

This procedure uses the New Exchange admin center. You can complete the same tasks by using the Classic Exchange admin center, but this guide does not document that process.

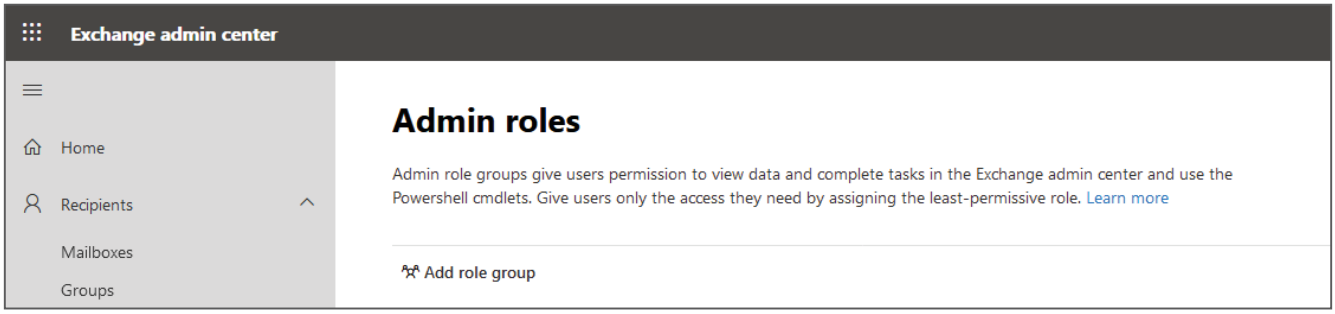
Step 1: Log in to the Exchange Admin Center at **https://admin.exchange.microsoft.com**.

Step 2: Click **Roles > Admin roles**.

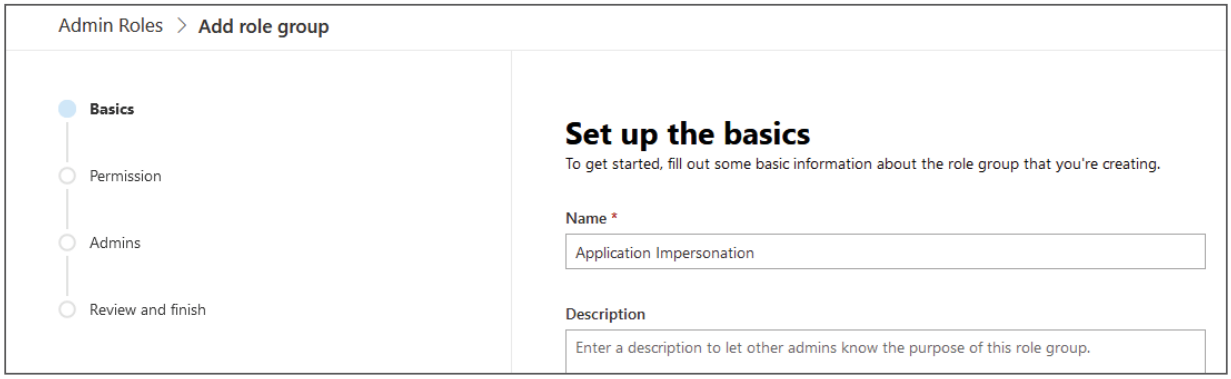


First, you create the Application Impersonation role group and assign the user to that group.

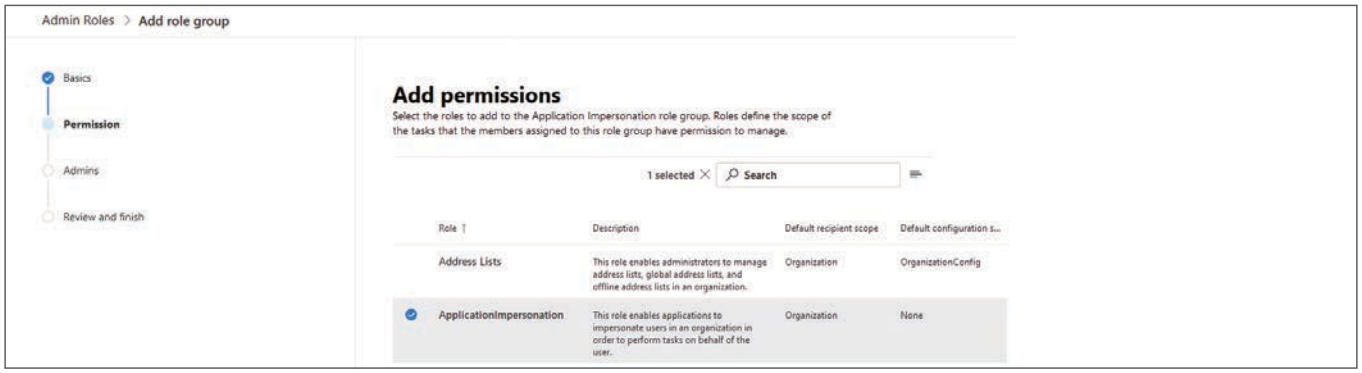
Step 3: On the Admin Roles page, click **Add role group**.



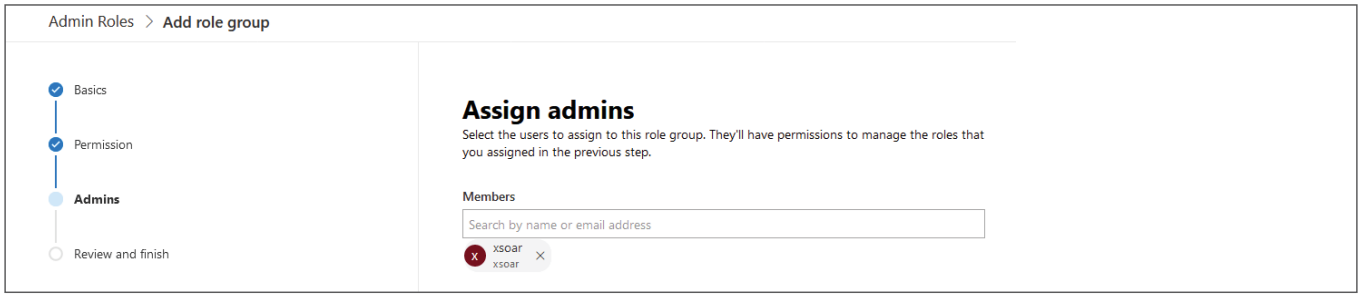
Step 4: On the Set Up the Basics page, in the **Name** box, enter **Application Impersonation**, and then click **Next**.



Step 5: On the Add Permissions page, select **ApplicationImpersonation**, and then click **Next**.



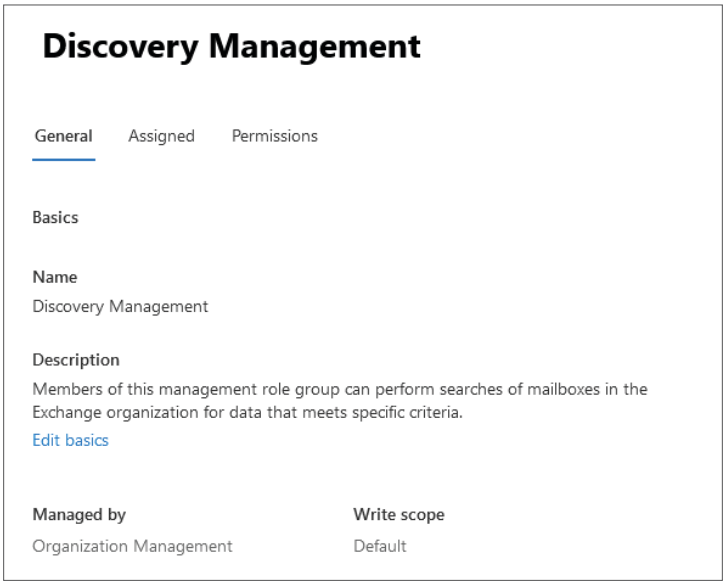
Step 6: On the Assign Admins page, in the **Members** box, enter **xsoar**, select **xsoar**, and then click **Next**.



Step 7: Review the role group settings. If they are correct, click **Add role group**, and then on the following page, click **Done**.

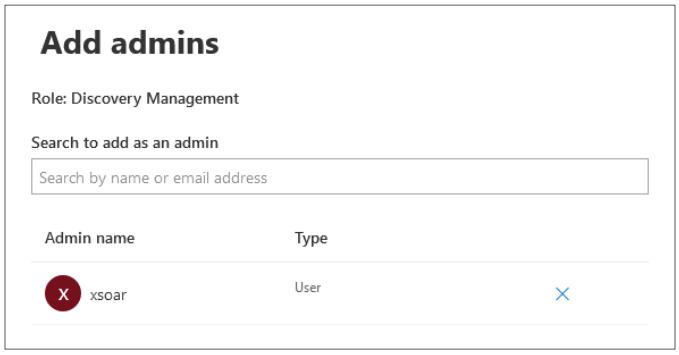
Next, you edit the existing Discovery Management role group and assign the user to that group.

Step 8: On the Admin Roles page, click **Discovery Management**.



Step 9: On the Assigned tab, click **Add**.

Step 10: On the Add Admins page, in the search box, enter **xsoar**, click **xsoar**, and then click **Add**.



Step 11: Click **X** to close the Discovery Management dialog box.

1.6 Configure EWS v2 Integration Instance

This procedure assumes that in Procedure 1.1, you installed the EWS content pack from the Cortex XSOAR Marketplace.

This procedure creates an instance of the EWS v2 integration. This integration fetches incidents by monitoring the phishing inbox. This integration also includes a variety of automation commands that you use within the playbook to retrieve suspected phishing emails for analysis and, if necessary, to delete the emails.

When using this integration to fetch an incident, Cortex XSOAR ingests event information and maps data within the email message to specific incident fields. The EWS v2 integration parses the email from the monitored mailbox by using the mapper contained in the phishing content pack EWS - Incoming Mapper.

Step 1: Log in to the Cortex XSOAR portal (example: https://xsoar.example.com).

Step 2: In the navigation pane, click **Settings**.

Step 3: In **Integrations > Servers & Services**, in the search box, enter **EWS v2**.

Step 4: Click **Add instance**.

Step 5: In the **Name** box, enter **EWSv2**.

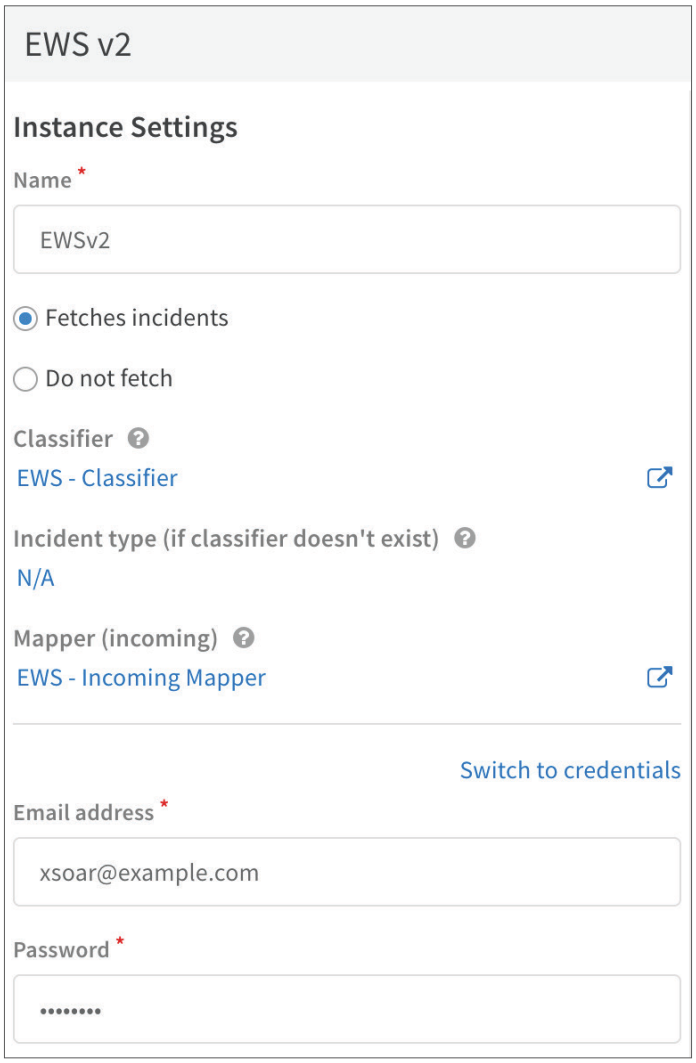
Step 6: Select **Fetches Incidents**.

Step 7: In the **Email address** box, enter **xsoar@example.com**.

Step 8: In the **Password** box, enter the password for **xsoar@example.com**.

Step 9: In the **Email address from which to fetch incidents** box, enter **phishing@example.com**.

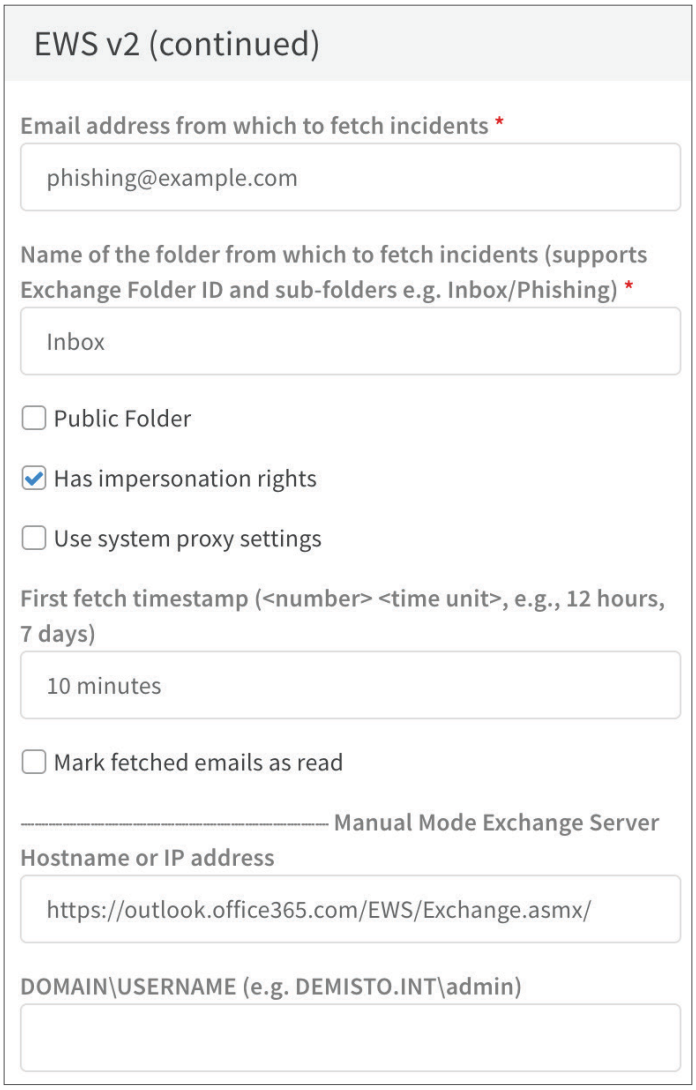
Step 10: In the **Name of the folder from which to fetch incidents** box, enter **Inbox**.



Step 11: Select **Has impersonation rights**.

Step 12: In the **Exchange Server Hostname or IP address** box, enter **https://outlook.office365.com/EWS/Exchange.asmx/**, and then click **Test**.

Step 13: Verify that you receive a success message, and then click **Save & exit**.



Procedures

Creating a Playbook to Investigate Phishing Emails

This playbook uses automation to retrieve a suspected phishing message from an organization’s Microsoft 365 email system and to analyze the message in order to identify malicious content.

2.1 Create the “Automated Phishing Investigation” Playbook

In this procedure, you create a playbook. Later, in Procedure 3.1, you assign this playbook to be the default playbook for phishing incidents and to run automatically.

When you assign this playbook later, you also configure Cortex XSOAR to automatically extract and enrich indicators before the playbook runs. By choosing this configuration option, Cortex XSOAR populates context data that is available for use by automation scripts and commands within the playbook.

Each new playbook automatically starts with a section-header task named Playbook Triggered.

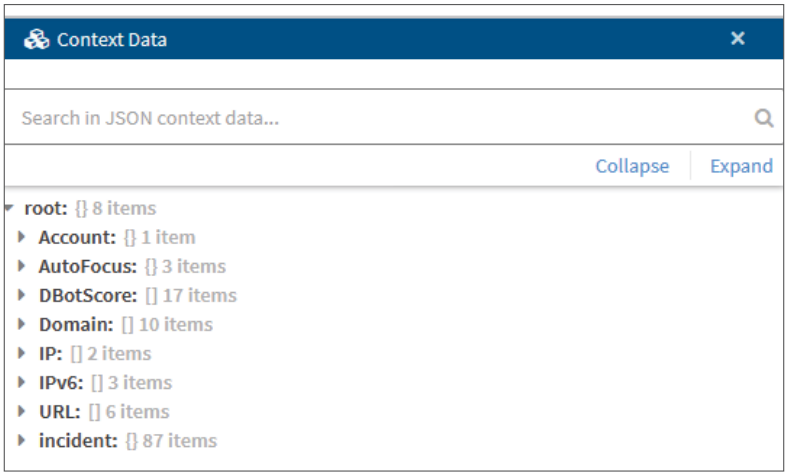
Step 1: In Cortex XSOAR, in the navigation pane, click **Playbooks**.

Step 2: Click **New Playbook**.

Step 3: In the New Playbook dialog box, in the **Playbook name** box, enter **Automated Phishing Investigation**, and then click **Save**. A playbook workspace with a Playbook Triggered section header appears.

Step 4: If the Task Library dialog box obscures your view of the playbook workspace, click **x** to close the dialog box.

If you were to run the playbook at this time, the context data available would include the objects shown.



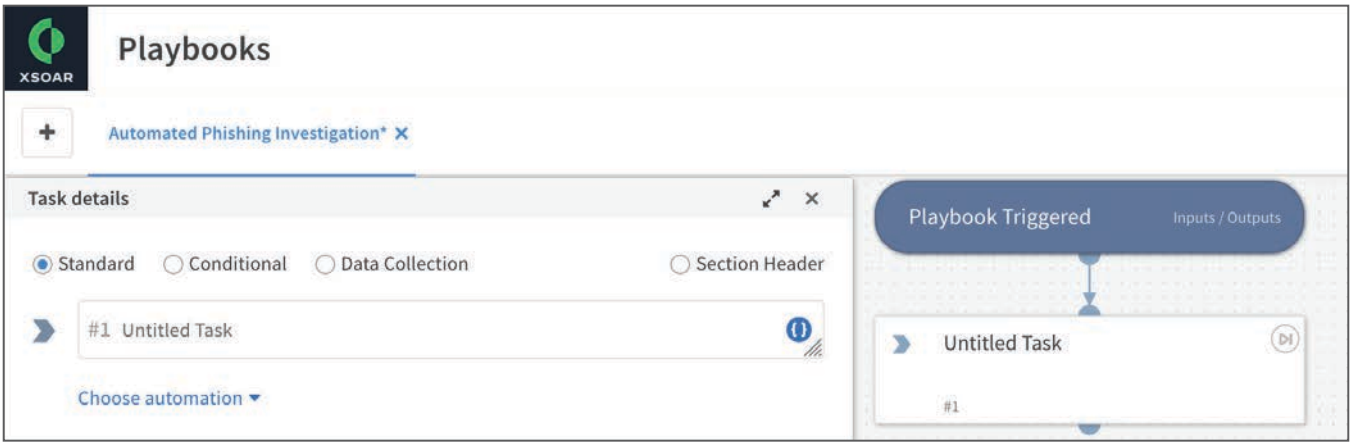
2.2 Create the "Assign Analyst" Task

When Cortex XSOAR creates the new incident, it does not assign an incident owner. Later in this guide, you add automation commands that require input from the case owner, so as your first task, you should assign a case owner.

This procedure assumes you have assigned users the analyst role.

In this procedure, you use the **AssignAnalystToIncident** automation script to select a Cortex XSOAR user with the Analyst role and set that user as the incident owner. By default, the automation script uses random selection from users with the role you specify. You can also choose from several other options when you configure the task.

Step 1: From the Playbook Triggered section header egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.



Step 2: In the box with the placeholder **Untitled Task**, enter **Assign analyst**.

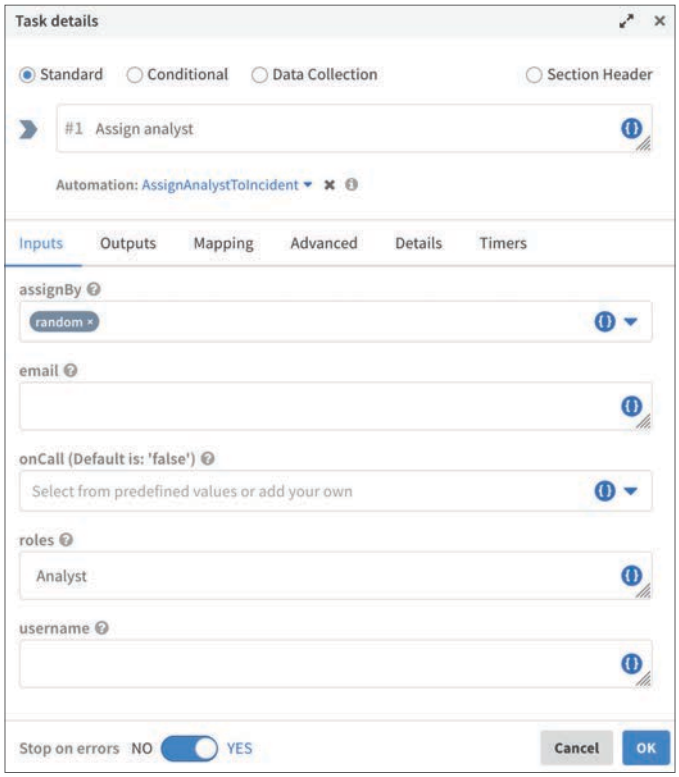
Step 3: In the Choose Automation section, click the down arrow. The search dialog box opens.

Step 4: In the **search** box, enter **AssignAnalystToIncident**, and then choose **AssignAnalystToIncident**. The task fields update.

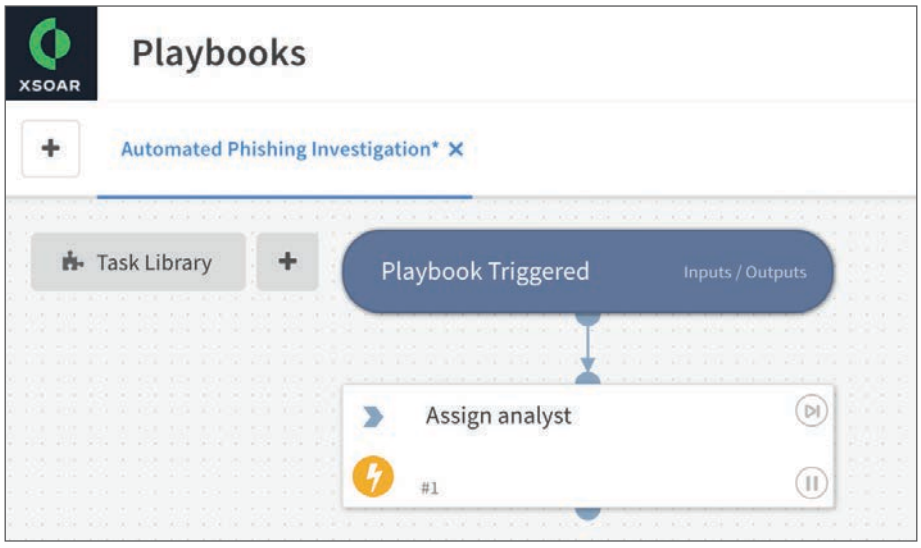
Step 5: In the **assignBy** list, select **random**.

Step 6: In the **roles** box, enter **Analyst**. This value is case sensitive.

Step 7: To add the completed task to the playbook, click **OK**.



Step 8: Verify that the task is now in your playbook.

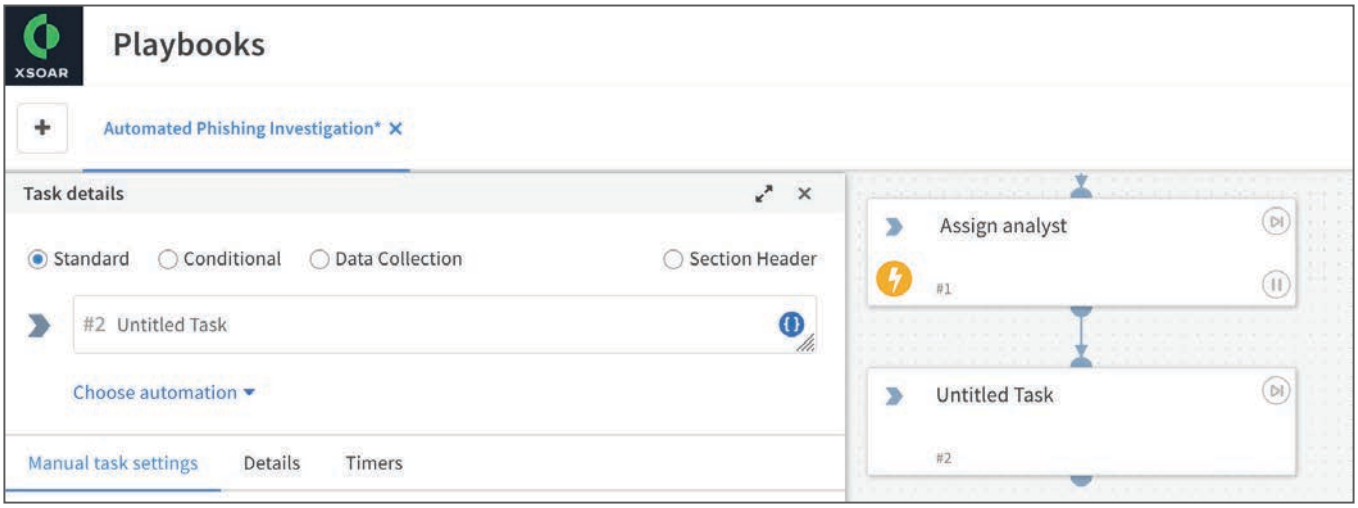


2.3 Create the "Is Message Forwarded?" Task

In this procedure, you perform a check to verify that the email that triggered the incident was actually forwarded to the monitored mailbox, as opposed to a message being composed and sent directly to the monitored mailbox. Your playbook executes different branches depending on the results of the check.

Forwarded emails typically contain the header field *In-Reply-To*, which contains a unique message identifier for the original message that was forwarded. If this field exists, then the email has been forwarded. When the incident that uses this playbook is triggered, the context data *incident.emailreplyto* object is assigned the value of the In-Reply-To field.

Step 1: From the Assign analyst task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.



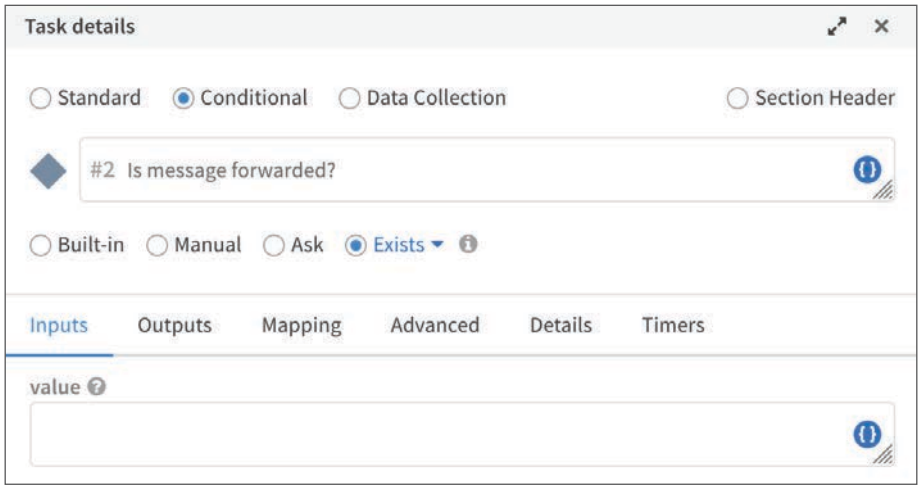
Step 2: Select **Conditional**.

Step 3: In the box with the placeholder **Untitled Task**, enter Is message forwarded?.

Step 4: Select **Choose automation**. The default automation, **AreValuesEqual**, is displayed.

Step 5: To the right of AreValuesEqual, click the down arrow to open the search dialog box.

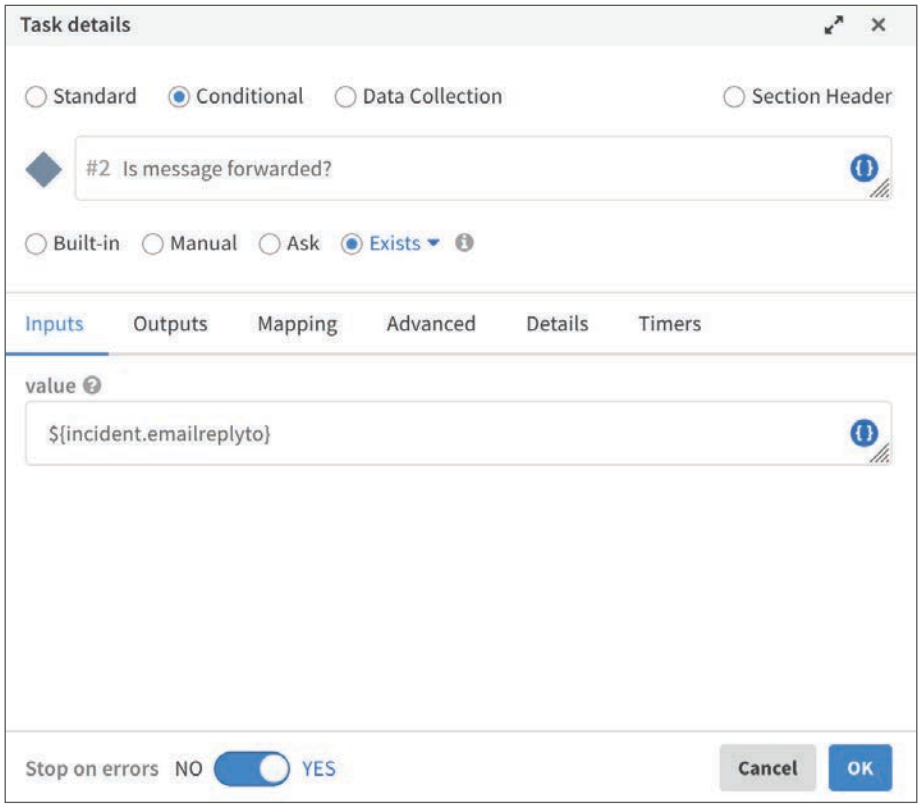
Step 6: In the search box, enter **Exists**, and then choose **Exists**. The task fields update.



Step 7: In the **value** box, click the {} button The Select Source For Value dialog box appears.

Step 8: In the search box, enter **Email Reply To**. In the Incident Details section, choose **Email Reply To**, and then click **Close**.

Step 9: Verify that the **value** box is now correctly populated with **${incident.emailreplyto}**, and then click **OK**.



Step 10: Verify that the task is now in your playbook.

2.4 Create the "Get Original Message" Task

This is the first task of a new branch of the playbook. The playbook selects this branch only if the email that triggered the incident is a forwarded message, as determined in Procedure 2.3.

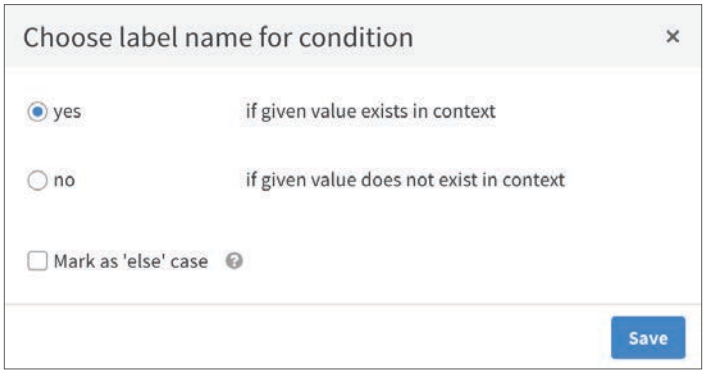
You use the **ews-search-mailbox** automation command from the EWS v2 integration in order to search the Microsoft 365 email system for the original email that was forwarded. In order to perform the search, you need to provide the message ID and recipient for the original email. This task assumes that the user that forwarded the email was also the original recipient.

In addition to the message ID information, the recipient information is also available from the context data. When the incident that uses this playbook is triggered, the context data *incident.emailfrom* object is assigned the value of the From field in the forwarded email header.

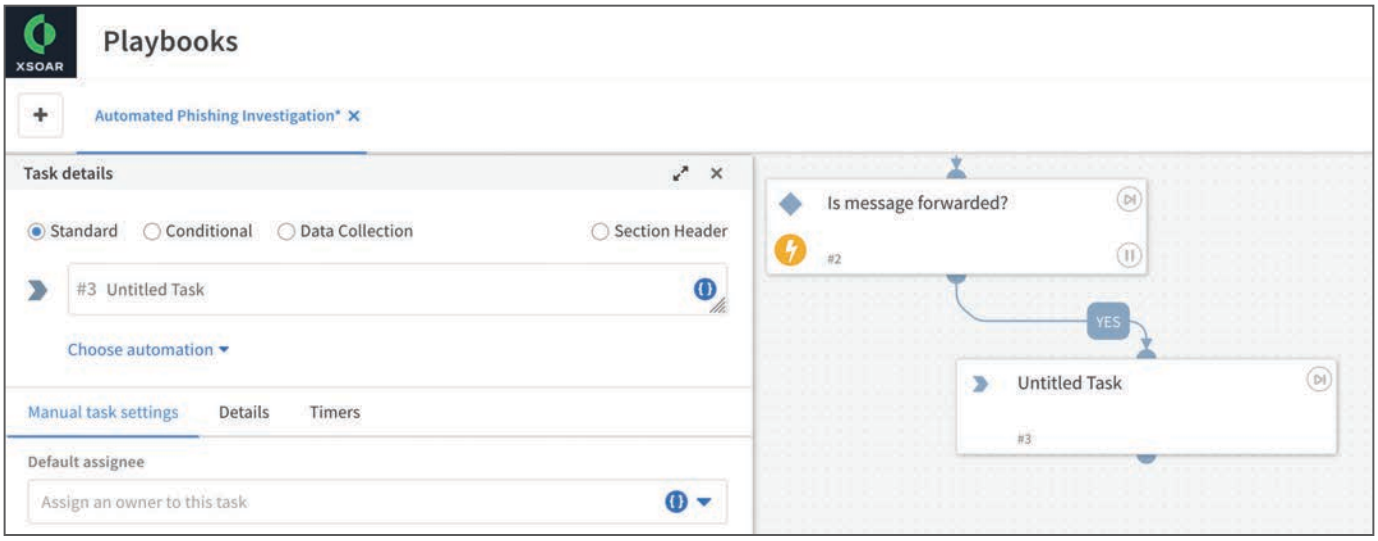
After you have run this automation command, Cortex XSOAR adds *EWS* context data.

Step 1: From the **Is message forwarded?** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below and to the right.

Step 2: In the Choose Label Name for Condition dialog box, select **yes**.



Step 3: Click **Save**. The Edit Task dialog box appears.



Step 4: In the box with the placeholder **Untitled Task**, enter **Get original message**.

Step 5: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 6: In the **search** box, enter **ews-search-mailbox (EWS v2)**, and then choose **ews-search-mailbox (EWS v2)**. The task fields update.

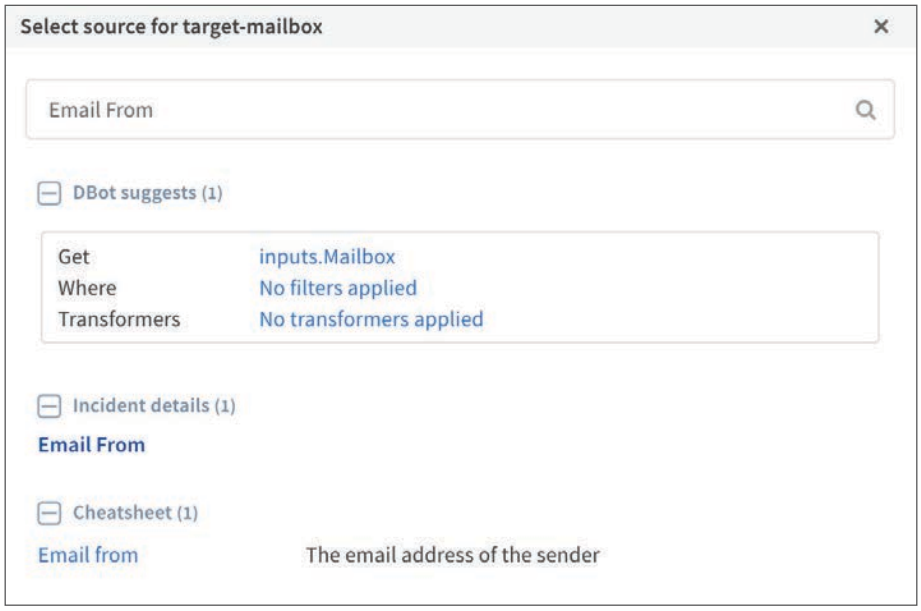
Step 7: In the **message-id** box, click the {} button. The Select Source for Message-id dialog box appears.

Step 8: In the search box, enter **Email Reply To**. In the Incident Details section, click **Email Reply To**, and then click **Close**.

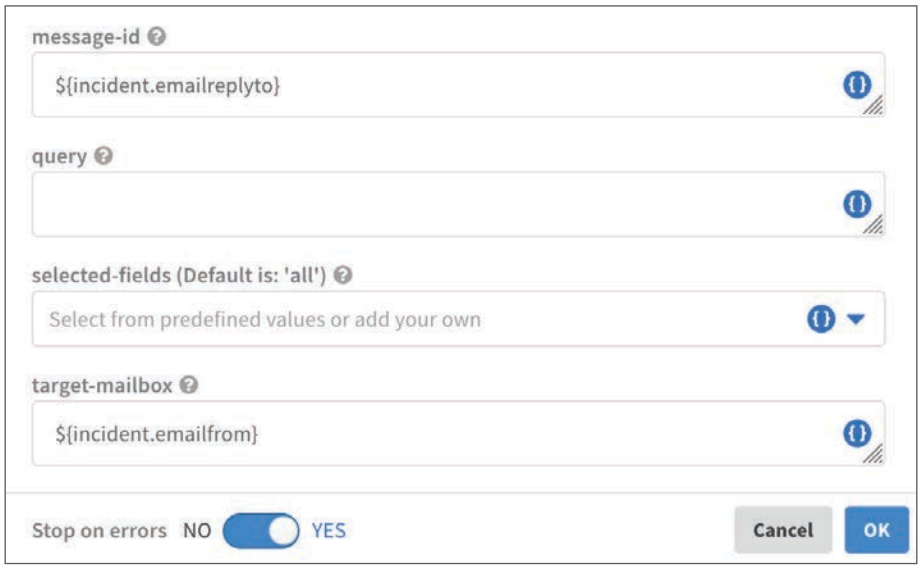
Step 9: Verify that the **message-id** box is now correctly populated with **${incident.emailreplyto}**.

Step 10: In the **target-mailbox** box, click the {} button. The Select Source for Target-Mailbox dialog box appears.

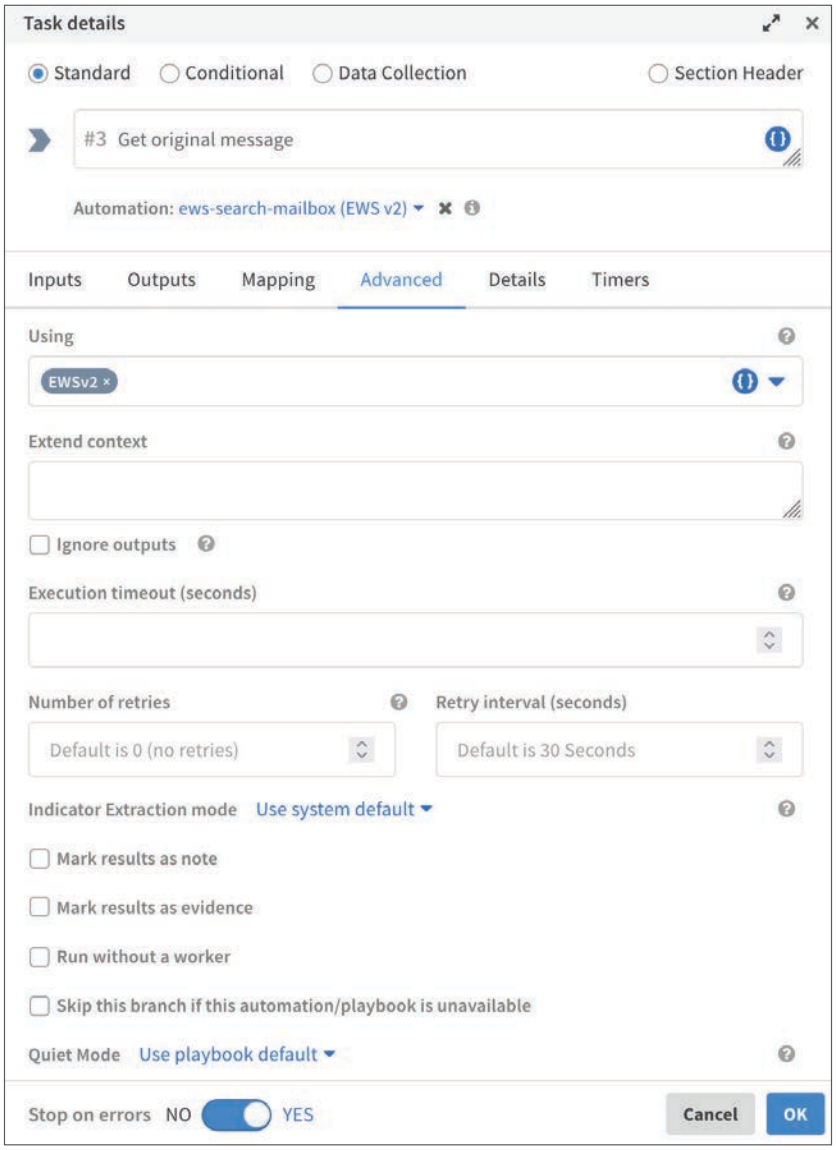
Step 11: In the search box, enter **Email From**. In the Incident Details section, click **Email From**, and then click **Close**.



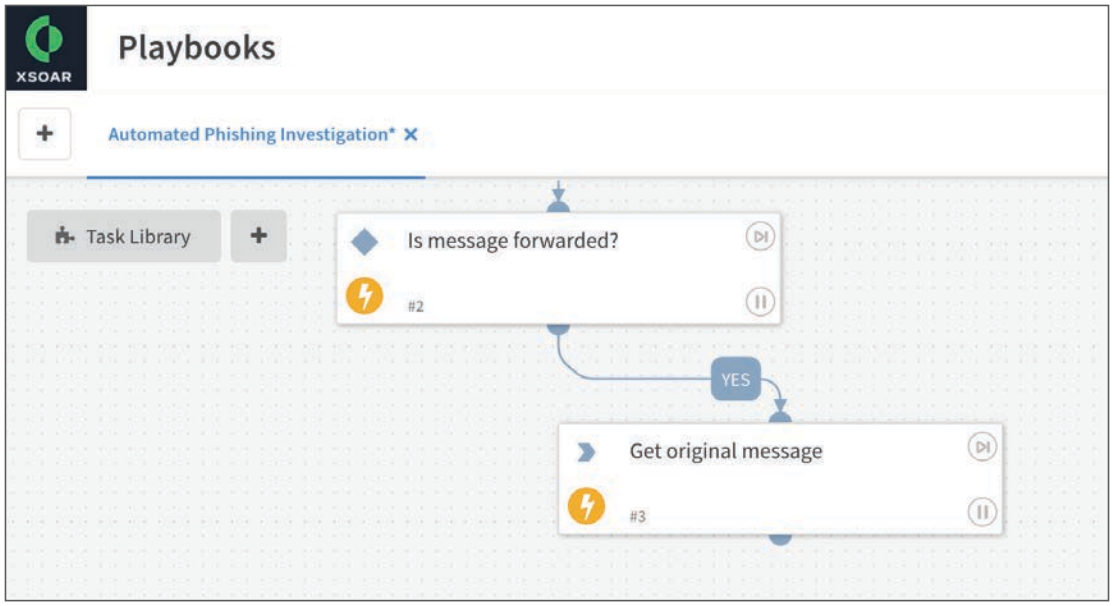
Step 12: Verify that the **target-mailbox** box is now correctly populated with **${incident.emailfrom}**.



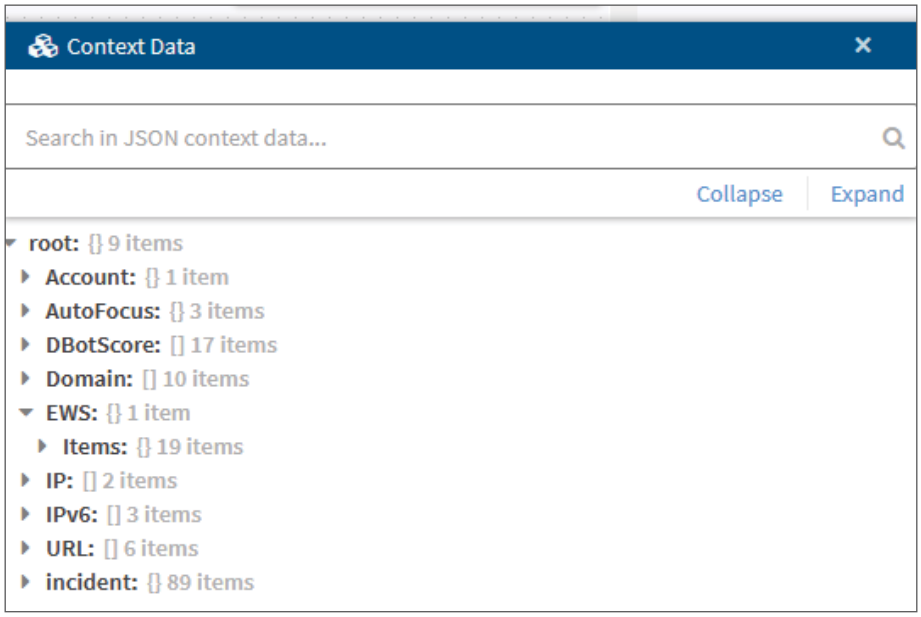
Step 13: On the Advanced tab, in the **Using** list, choose **EWSv2**, and then click **OK**.



Step 14: Verify that the task is now in your playbook.

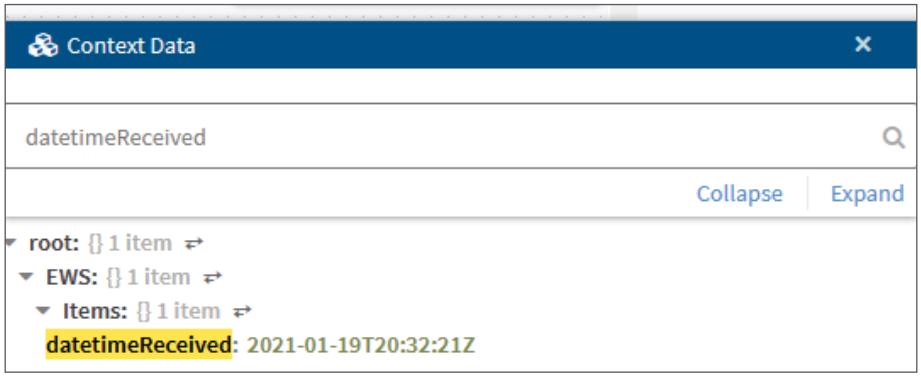


If you were to run the playbook at this time, the context data available would now include *EWS* objects, as shown.



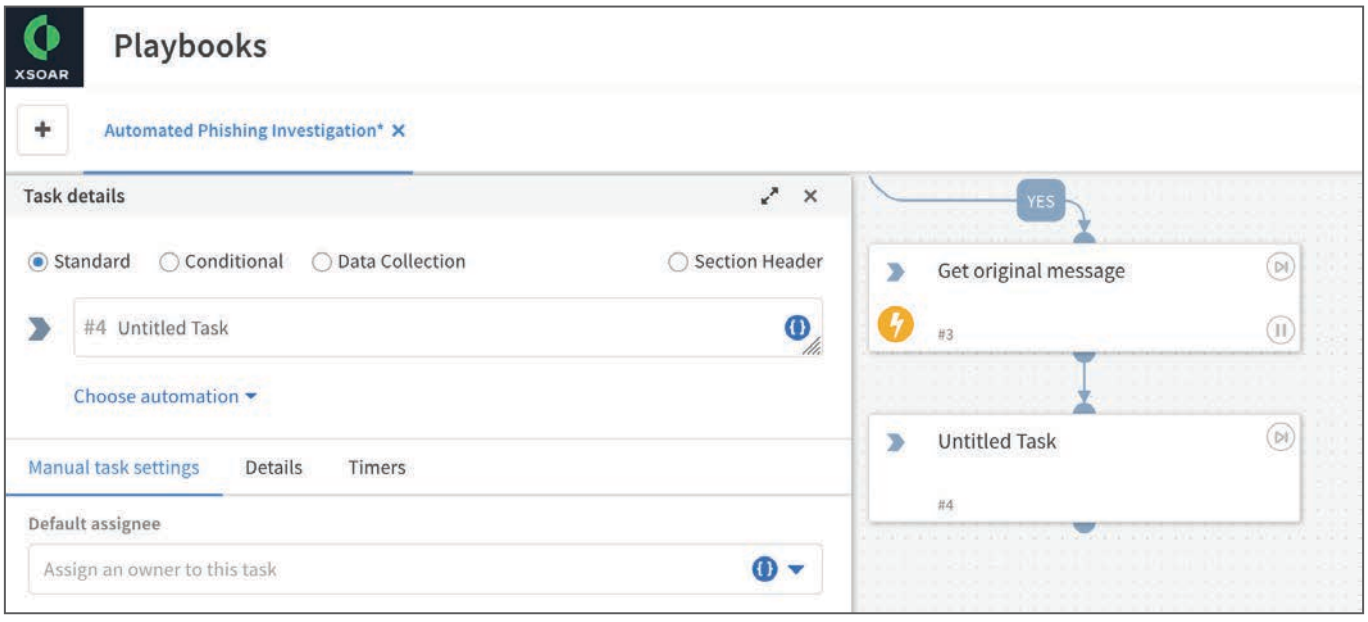
2.5 Create the "Set First Seen" Task

To optimize the search of system logs in future tasks, you can extract the time at which the original message was received and assign the time to the *incident.firstseen* value of the context data. This task uses the **setIncident** built-in automation command, which uses the *EWS.Items.datetimeReceived* context data as an input.



Later in this playbook, you reference the *incident.firstseen* value as the start time for other search automation tasks.

Step 1: From the Get original message task egress node, drag the task connector line to the playbook workspace below, and then release to create an untitled task. The Edit Task dialog box appears.



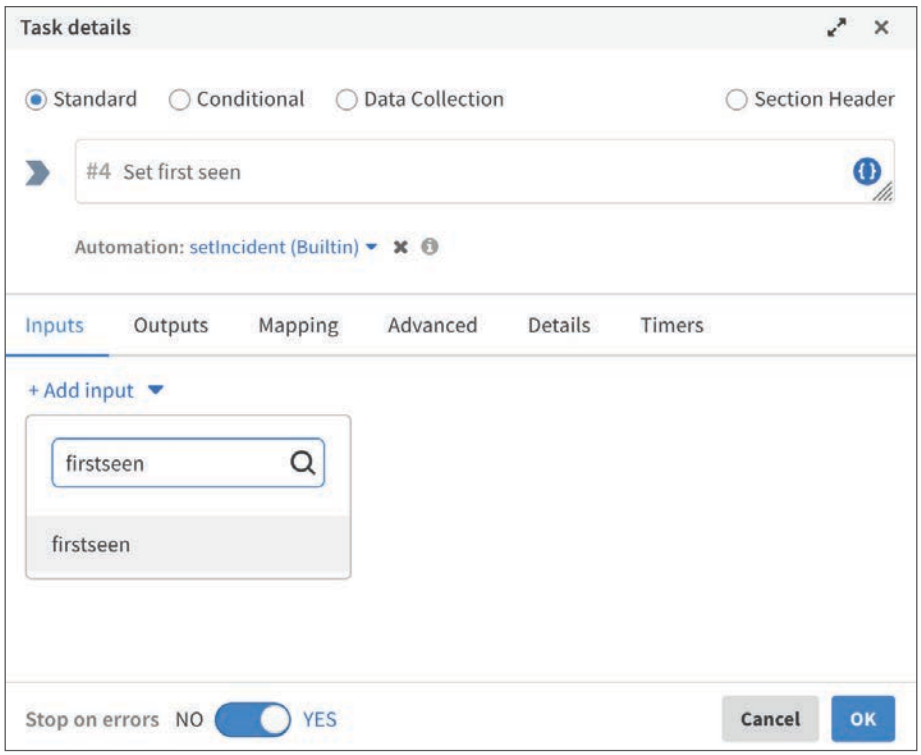
Step 2: In the box with the placeholder **Untitled Task**, enter Set first seen.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 4: In the **search** box, enter **setIncident (Builtin)**, and then choose **setIncident (Builtin)**. The task fields update.

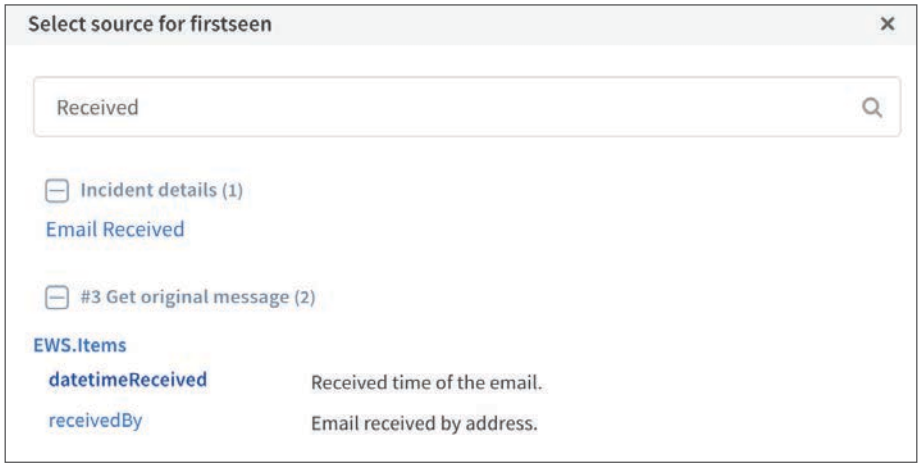
Step 5: In the **Add input** section, click the down arrow to open the search dialog box.

Step 6: In the **search** box, enter **firstseen**, and then choose **firstseen**.



Step 7: In the **firstseen** box, click the {} button. The Select Source for Firstseen dialog box appears.

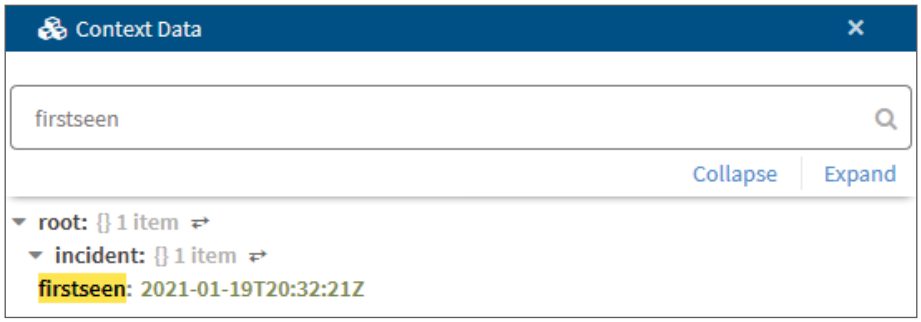
Step 8: In the search box, enter **Received**. In the EWS.Items section, click **datetimeReceived**, and then click **Close**.



Step 9: Verify that the **firstseen** box is correctly populated with the value from Step 8, and then click **OK**.

Step 10: Verify that the task is now in your playbook.

If you were to run the playbook at this time, the context data would now include the *incident.firstseen* object, as shown.



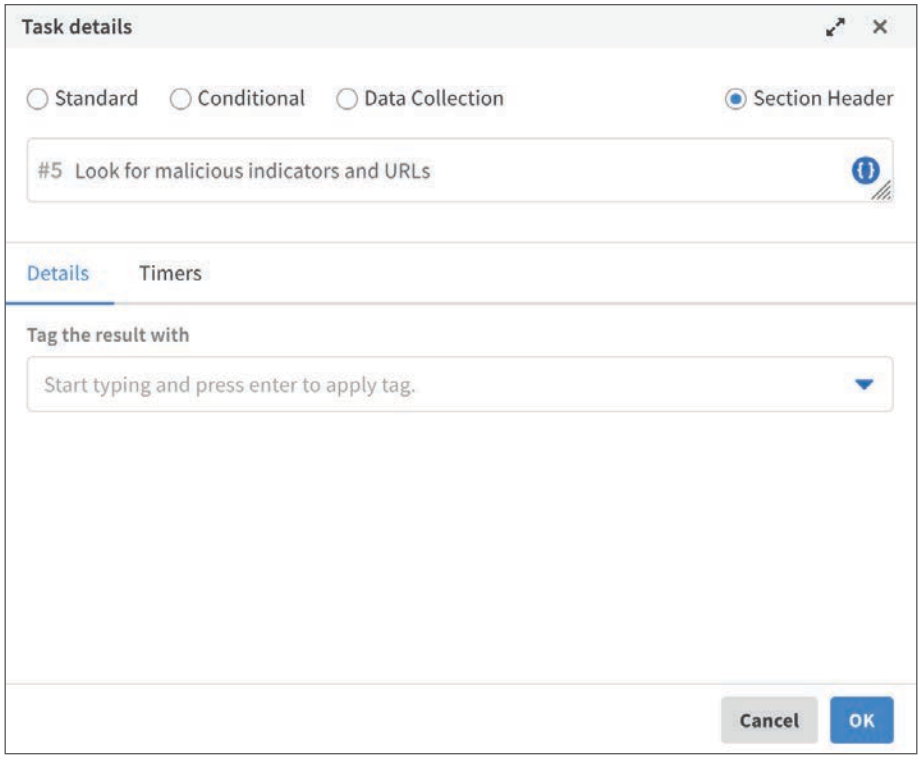
2.6 Create a Separator Task

As a best practice, you should logically separate longer playbooks by using section headers. You use this procedure to create a section header that demarcates the playbook tasks that analyze the contents of the suspected phishing email.

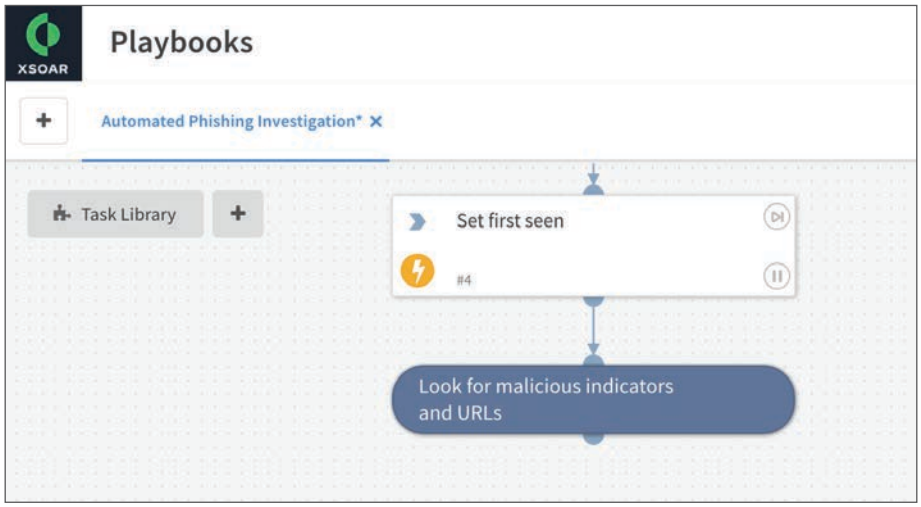
Step 1: From the **Set first seen** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled blank task below. The Edit Task dialog box appears.

Step 2: Select **Section Header**.

Step 3: In the box with the placeholder **Untitled Task**, enter **Look for malicious indicators and URLs**, and then click **OK**.



Step 4: Verify that the task is now in your playbook.

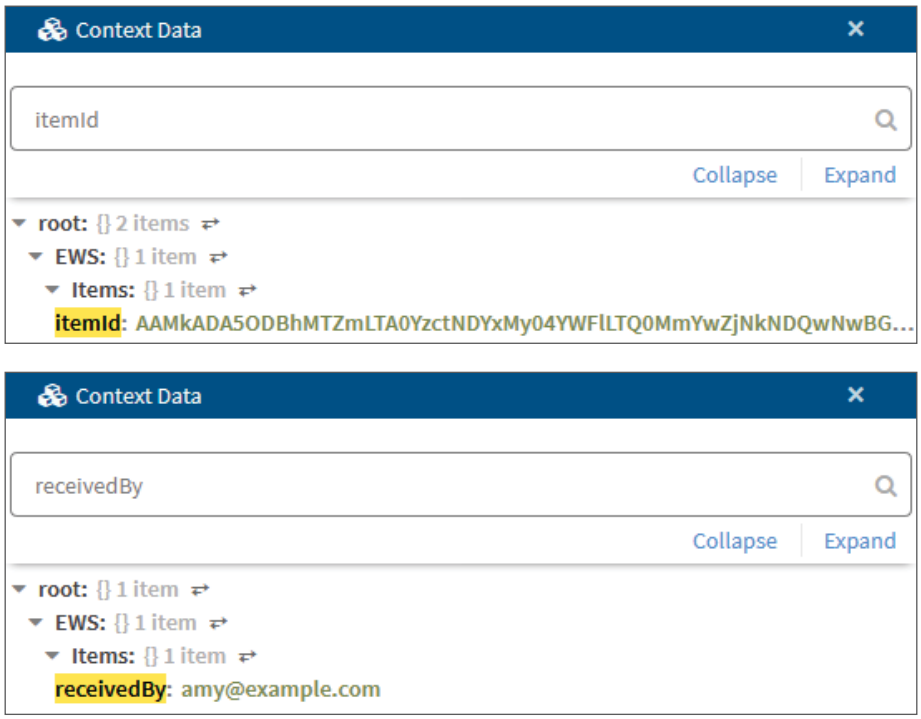


2.7 Create the "Retrieve EML Version of Message" Task

You use the **ews-get-items-as-eml** automation command from the EWS v2 integration to retrieve the suspected phishing email in .eml file format from the Microsoft 365 email system. Later in this playbook, you use automation commands that require this file format.

In order to perform the search, you need to provide the message item ID and recipient for the original email.

This task uses *EWS.Items.itemId* and *EWS.Items.receivedBy* context data and assumes that the **ews-search-mailbox** automation command has been run previously in this playbook.



After you have run this automation command, Cortex XSOAR adds *File* context data to the incident and adds the actual email file to the incident files as an artifact.

Step 1: From the **Look for malicious indicators and URLs** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

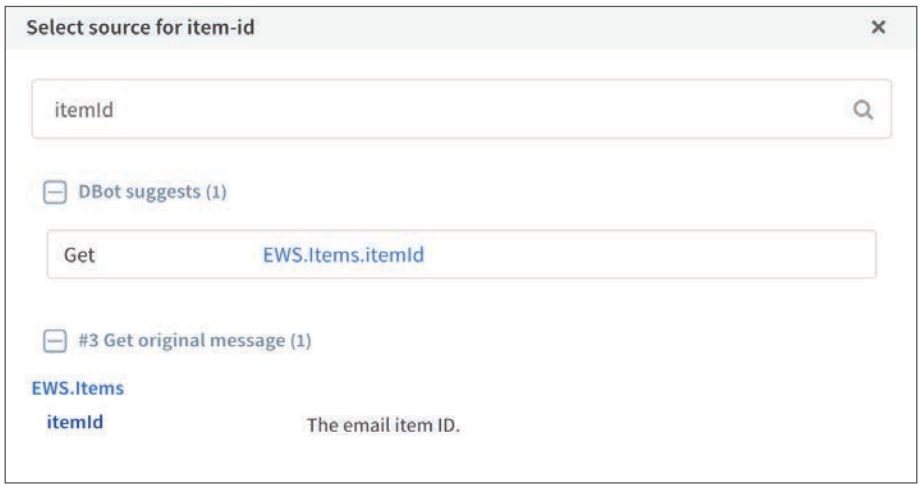
Step 2: In the box with the placeholder **Untitled Task**, enter **Retrieve EML version of message**.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 4: In the **search** box, enter **ews-get-items-as-eml (EWS v2)**, and then choose **ews-get-items-as-eml (EWS v2)**. The task fields update.

Step 5: In the **item-id** box, click the {} button. The Select Source for Item-id dialog box appears.

Step 6: In the search box, enter **itemId**. In the Get Original Message section, click **itemID**, and then click **Close**.

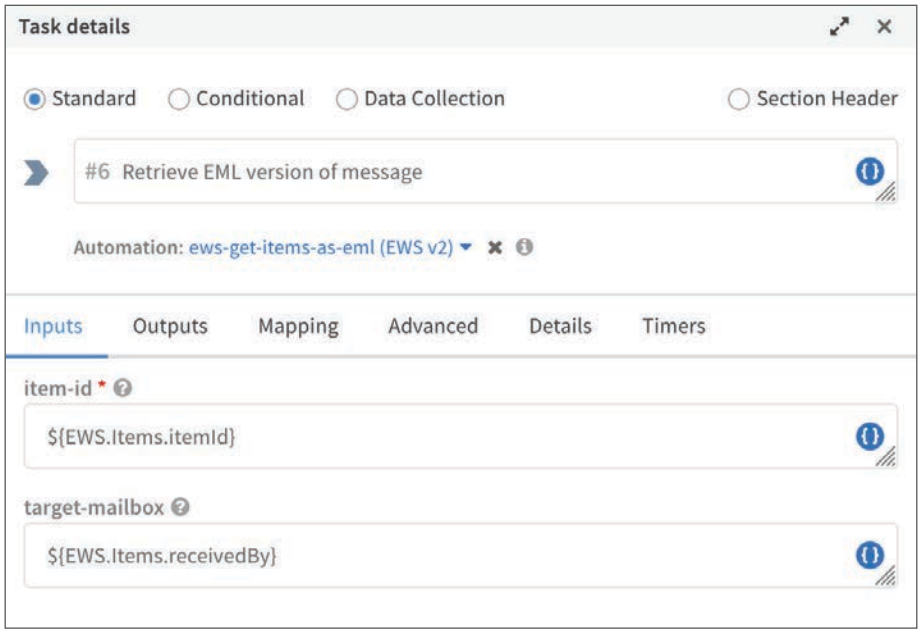


Step 7: Verify that the **item-id** box is now correctly populated with **${EWS.Items.itemId}**.

Step 8: In the **target-mailbox** box, click the {} button. The Select Source for Target-Mailbox dialog box appears.

Step 9: In the search box, enter **receivedBy**. In the EWS.Items section, click **receivedBy**, and then click **Close**.

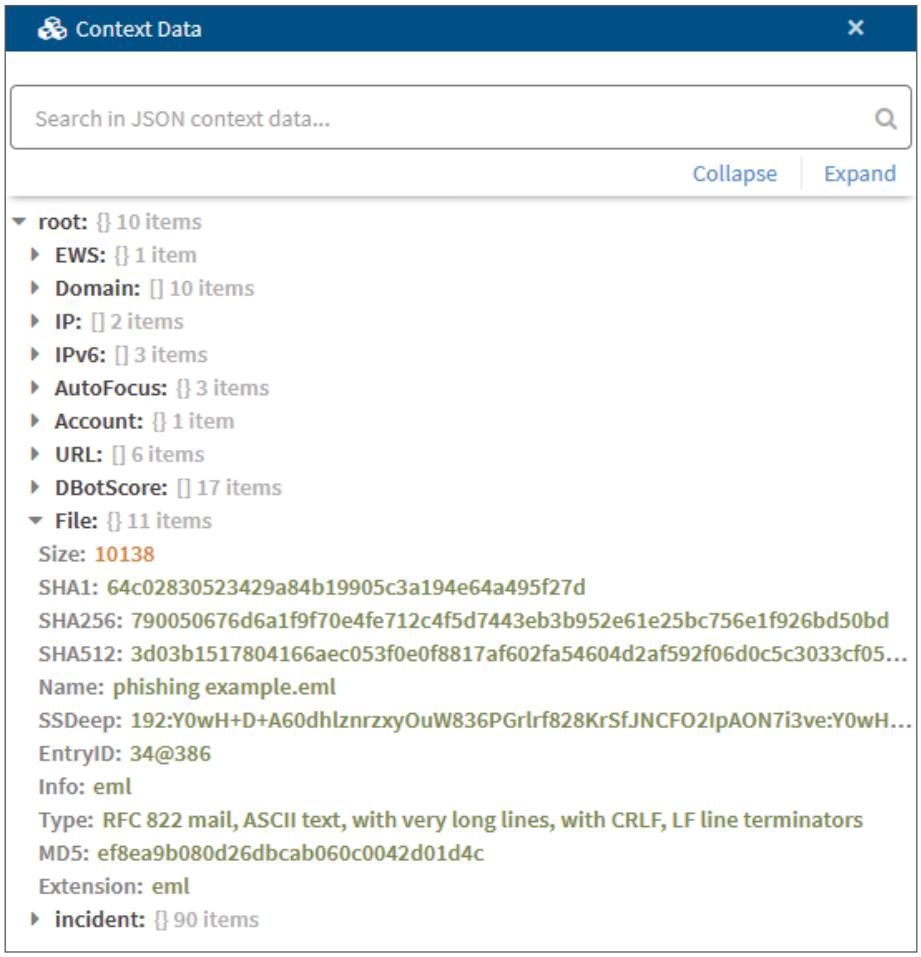
Step 10: Verify that the **target-mailbox** box is now populated with **${EWS.Items.receivedBy}**.



Step 11: On the Advanced tab, in the **Using** list, choose **EWSv2**, and then click **OK**.

Step 12: Verify that the task is now in your playbook.

If you were to run the playbook at this time, the context data available would now include *File* objects, as shown.



2.8 Create the "Parse Email Files" Task

You use the **ParseEmailFiles** automation script to extract complete header details from the email message.

In order to use this automation, you need to provide the file entry ID for the email.

This automation script uses *File.entryId* context data and assumes that you ran the **ews-get-items-as-eml** automation command previously in this playbook. In case the email includes attachments with non-.eml formats, when choosing the file entry ID, make sure that the file type is .eml.

After you have run this automation script, Cortex XSOAR adds *Email* context data to the incident.

Step 1: From the **Retrieve EML version of message** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

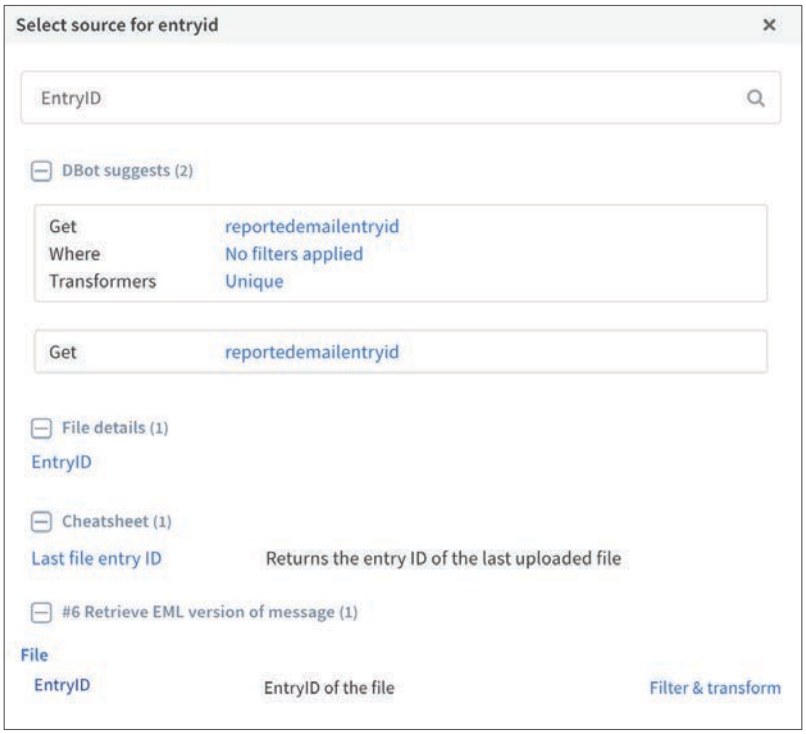
Step 2: In the box with the placeholder **Untitled Task**, enter **Parse email files**.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 4: In the **search** box, enter **ParseEmailFiles**, and then choose **ParseEmailFiles**. The task fields update.

Step 5: In the **entryid** box, click the {} button. The Select Source for Entryid dialog box appears.

Step 6: In the search box, enter **EntryID.** In the File section, hover over **EntryID**, and then in the EntryID row, click **Filter & Transform**. The dialog box name changes to Filters & Transformers for Entryid.

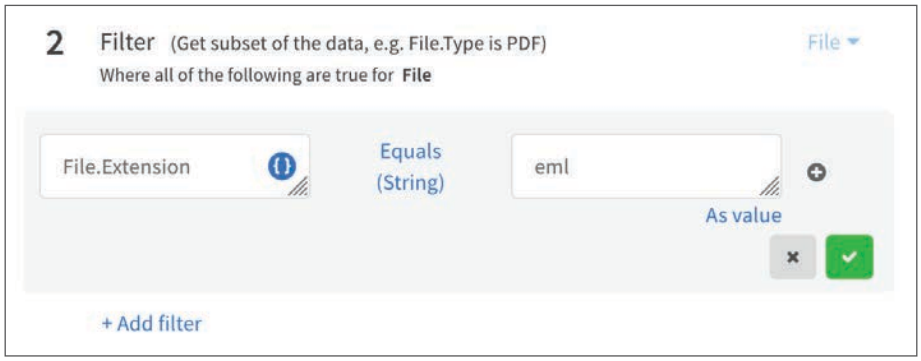


Step 7: On the Filters & Transformers for Entryid dialog box, click **Add filter**.

Step 8: In the conditional statement left-side box, click the {} button. The Choose Field of File dialog box appears.

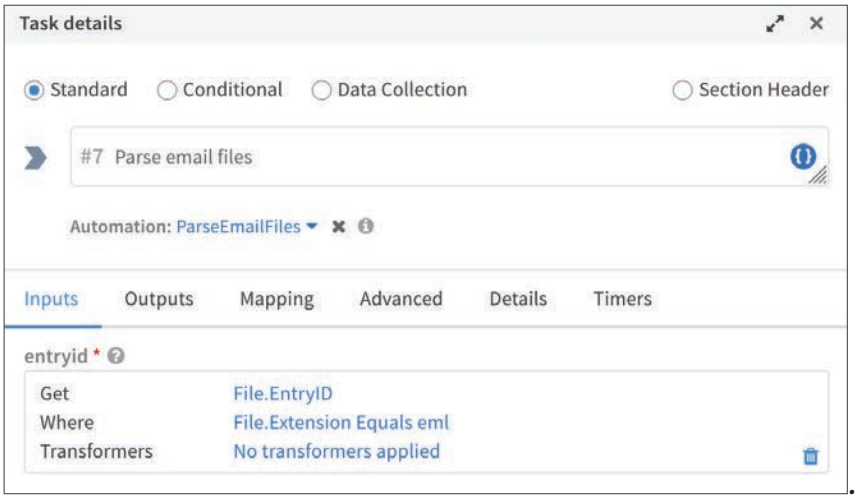
Step 9: In the search box, enter **Extension,** and then in the File section, click **Extension**.

Step 10: In the conditional statement right-side box, enter **eml**, and then click the check.

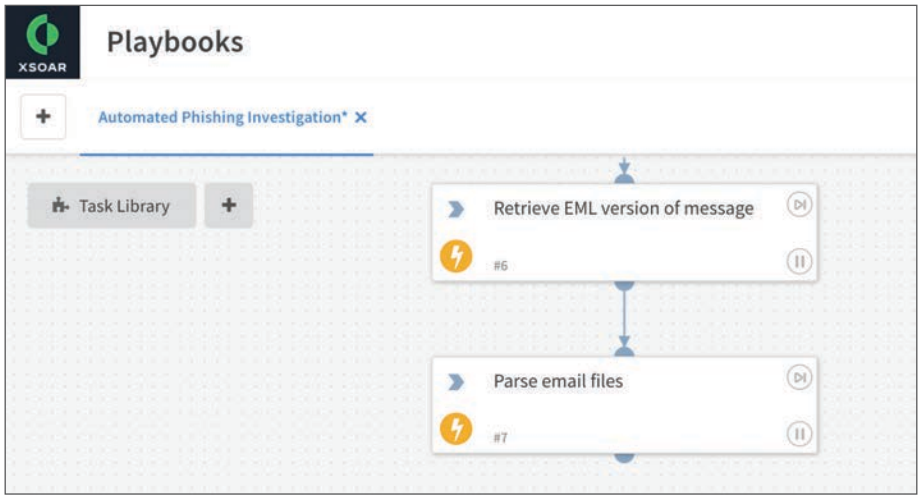


Step 11: Click **OK** to close the Filters & Transformers for Entryid dialog box.

Step 12: Verify that the **entry-id** box is now correctly populated with the correct GWT structure, and then click **OK**.



Step 13: Verify that the task is now in your playbook.



2.9 Add the "Detonate URL - WildFire v2.1" Sub-Playbook Task

If the suspected phishing email contains any URLs, to determine their reputation, you submit them to WildFire for analysis.

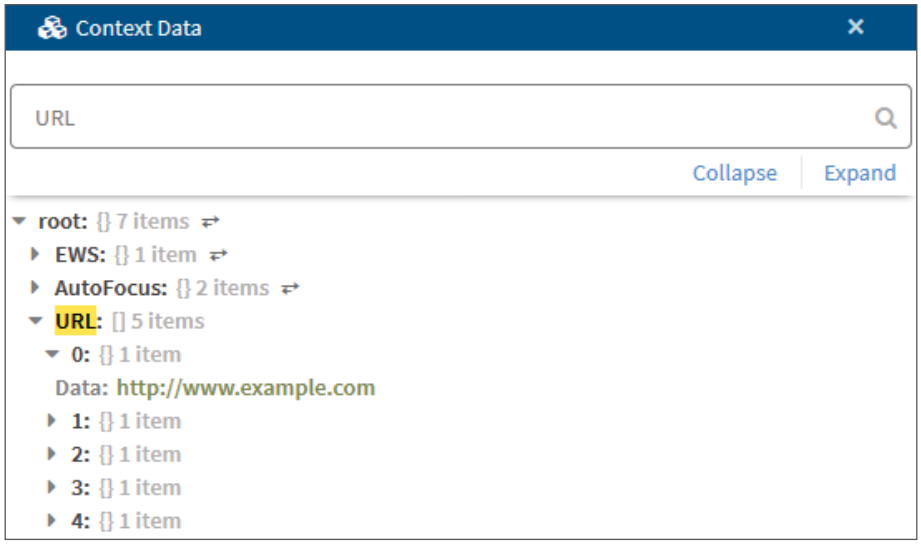
You use a system playbook as a sub-playbook in this task. The sub-playbook uses the **wildfire-upload-file-url**, **wildfire-upload-url**, and **wildfire-report** automation commands from the Palo Alto Networks WildFire v2 integration.

Note

The run time for this sub-playbook depends on the WildFire processing time. The sub-playbook submits each URL to WildFire and then checks for an existing report. If a report already exists for the submitted URL, then WildFire returns the verdict immediately.

If a report does not exist, then the sub-playbook starts a 1-minute polling timer. After the timer expires, the sub-playbook then checks again for a report. The sub-playbook repeats this process until the report is available.

This playbook uses the *URL.Data* context data as an input. If the suspected phishing email contains multiple URLs, then *URL.Data* contains an array of URLs. The sub-playbook automatically processes all URLs, and you do not need to configure the task to loop.

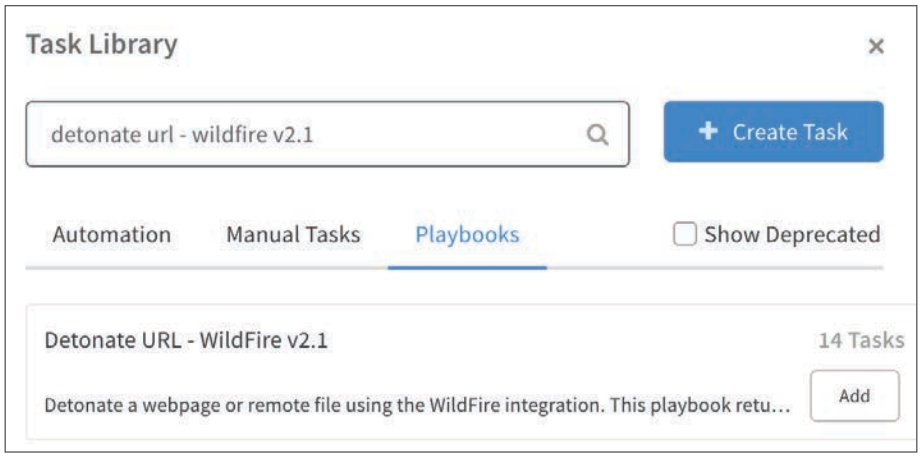


After you have run this sub-playbook task, Cortex XSOAR adds *WildFire* context data, which contains the verdict for each URL, to the incident.

Step 1: On the playbook workspace, click **Task Library**.

Step 2: On the Playbooks tab, in the search box, enter **detonate url - wildfire v2.1**. The search box accepts only lowercase characters.

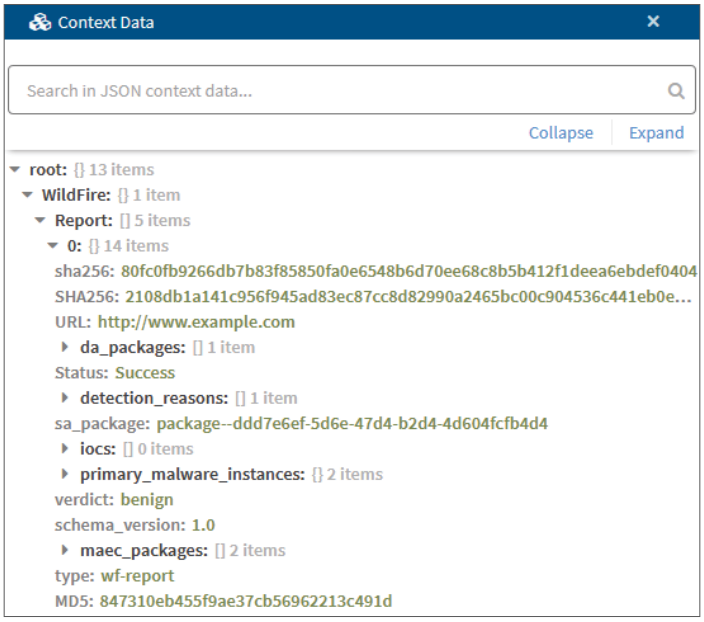
Step 3: In the search results, for the playbook **Detonate URL - WildFire v2.1**, click **Add**. Cortex XSOAR adds the new sub-playbook task to the playbook workspace.



Step 4: From the **Parse email files** task egress node, drag the task connector line to the **Detonate URL - WildFire v2.1** task ingress node and release. Cortex XSOAR adds a connector line between the tasks.

Step 5: Verify that the task is now in your playbook.

If you were to run the playbook at this time, the context data available would now include *WildFire* objects, as shown.



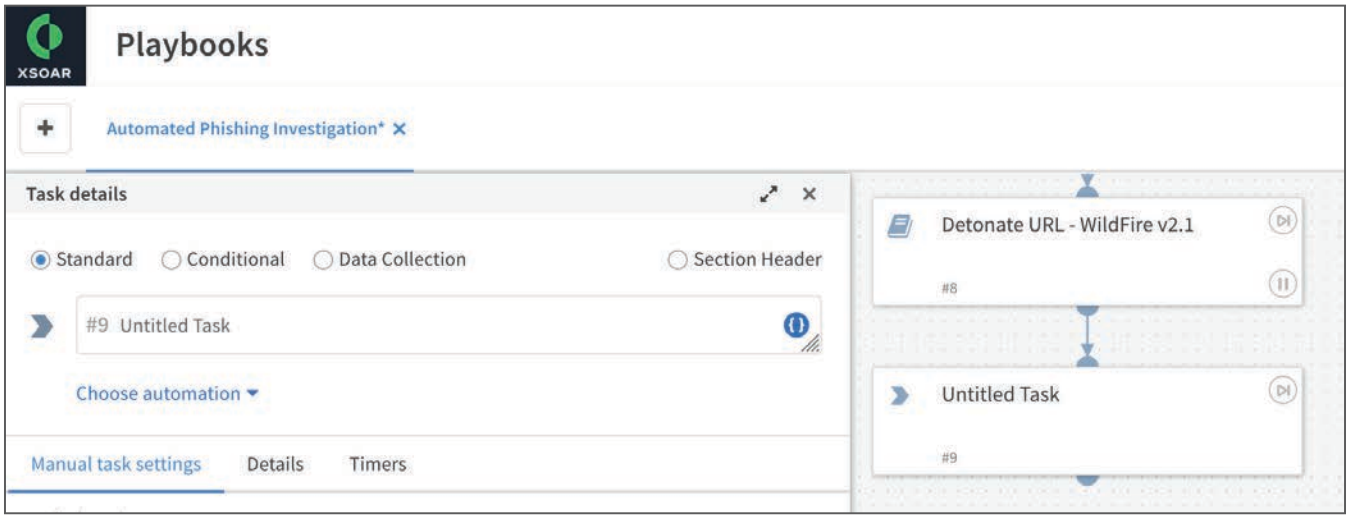
2.10 Create the "Did WildFire Find a Malicious URL?" Task

In this procedure, you perform a check to see if WildFire returned a "Malicious" verdict for any of the URLs that you submitted for analysis in Procedure 2.9. Your playbook executes different branches depending on the results of the check.

To perform the check, your conditional statement uses the *WildFire.Report* context data. For each URL that you submitted for analysis, the *WildFire.Report.verdict* object contains the verdict value. You need to use a complex conditional statement in this task.

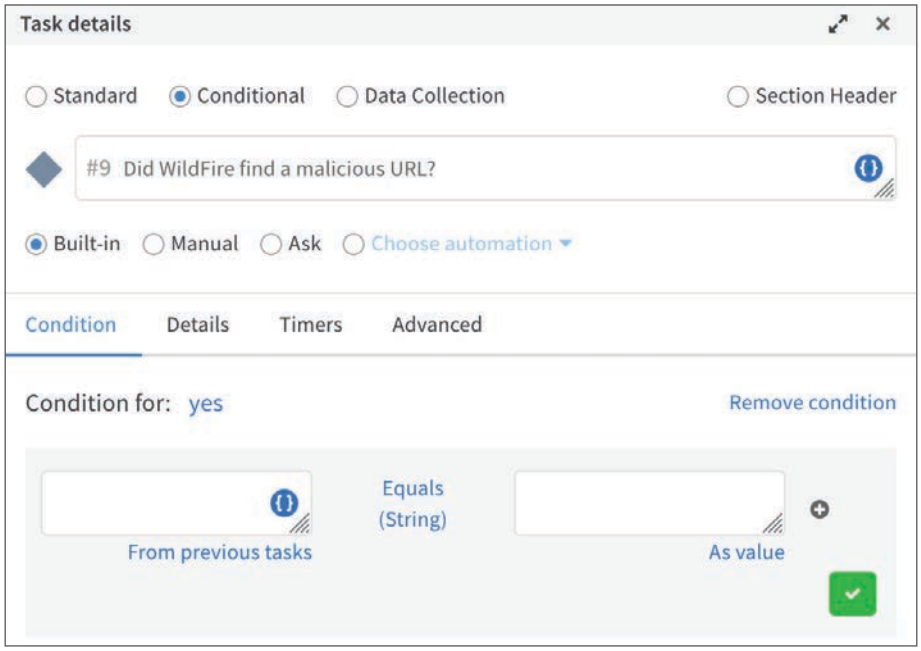
First, you check each of the verdicts to see if there is a match for "phishing" or "malware." You accomplish this by filtering the verdicts to eliminate the non-matching verdicts. Next, you check the list of remaining verdicts. If the list of verdicts is not empty, then one or more verdicts was "phishing" or "malware." If the list of verdicts is empty, then none of the verdicts were "phishing" or "malware."

Step 1: From the **Detonate URL - WildFire v2.1** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.



Step 2: Select **Conditional**.

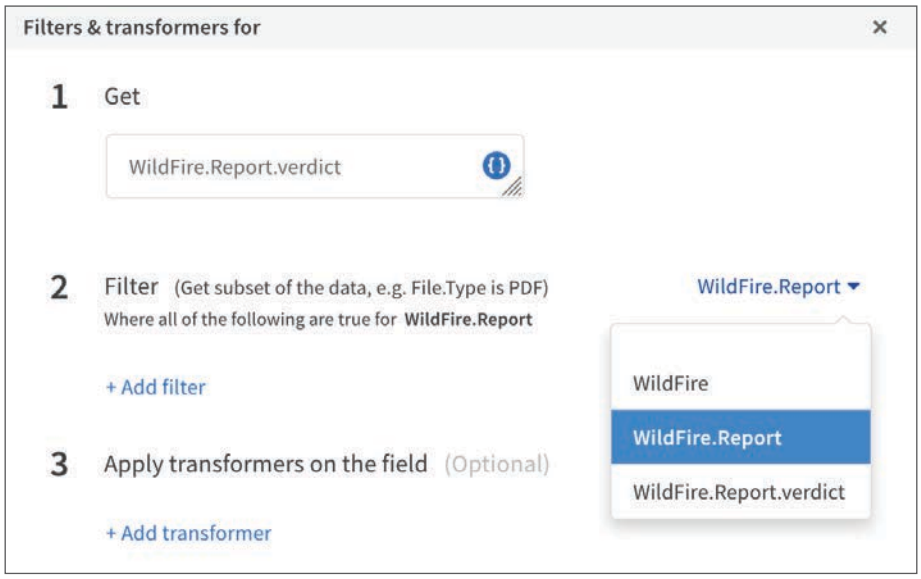
Step 3: In the box with the placeholder **Untitled Task**, enter **Did WildFire find a malicious URL?**.



Step 4: In the conditional statement section left-side box, click the {} button. The Select Source For dialog box appears.

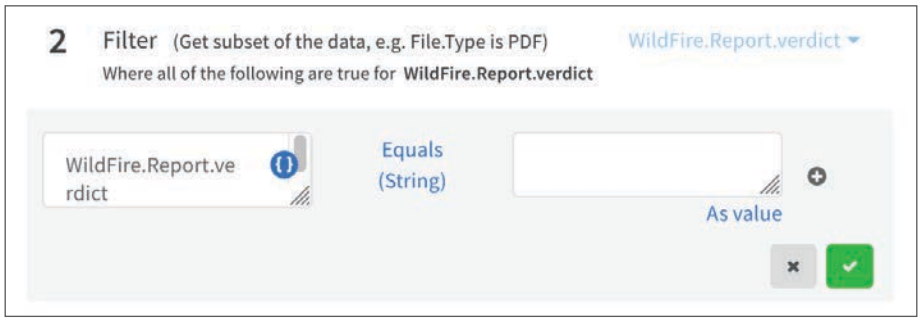
Step 5: Click **Filters And Transformers**. The dialog box name changes to Filters & Transformers For.

Step 6: In the **Get** box, enter **WildFire.Report.verdict**. This value is case sensitive.



Step 7: In the Filter section, click **WildFire.Report**, and then choose **WildFire.Report.verdict**.

Step 8: Click **Add Filter**.



Next, you choose the comparison operator for the filter.

Step 9: Click **Equals**.

Step 10: In the search box, enter **Contains**, and then click **Contains**.

Step 11: In the right-side box, enter **phishing**. This value is case sensitive.

Step 12: To add a second filter condition, click the **+**.

Step 13: In the conditional statement left-side box, enter **WildFire.Report.verdict**. This value is case sensitive.

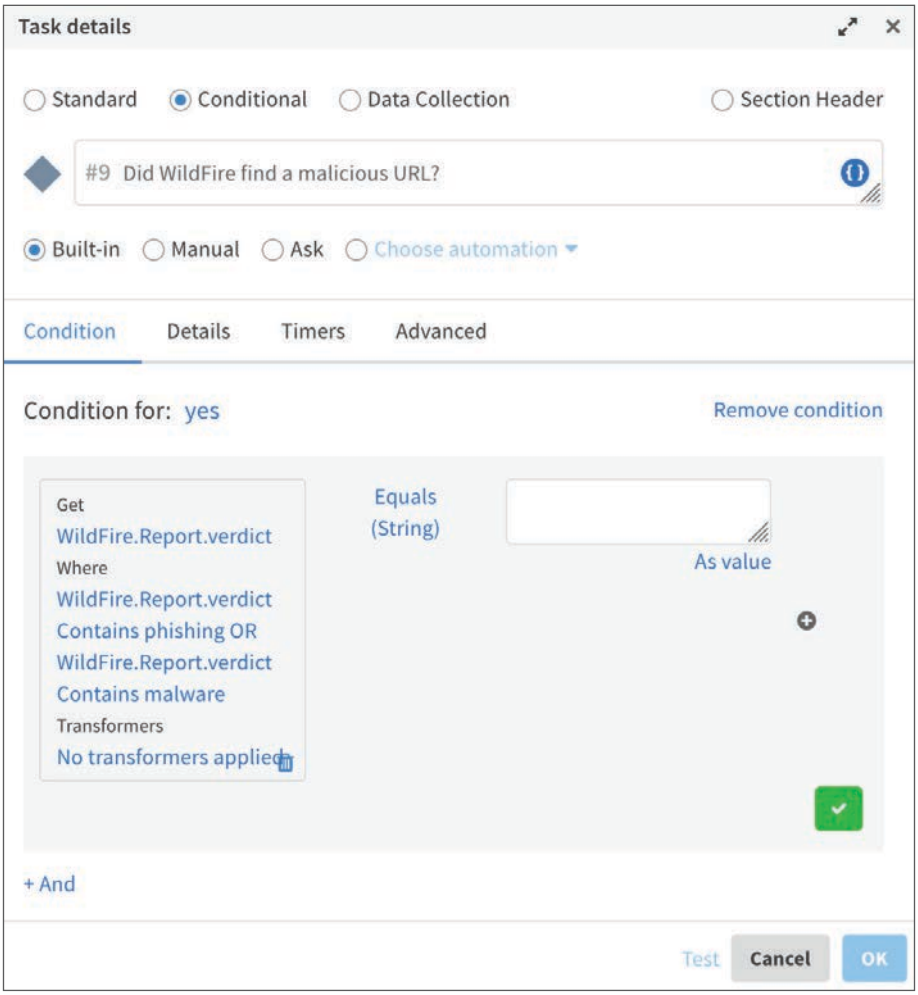
Next, you choose the comparison operator for the filter.

Step 14: Click **Equals**.

Step 15: In the search box, enter **Contains**, and then click **Contains**.

Step 16: In the right-side box, enter **malware**. This value is case sensitive.

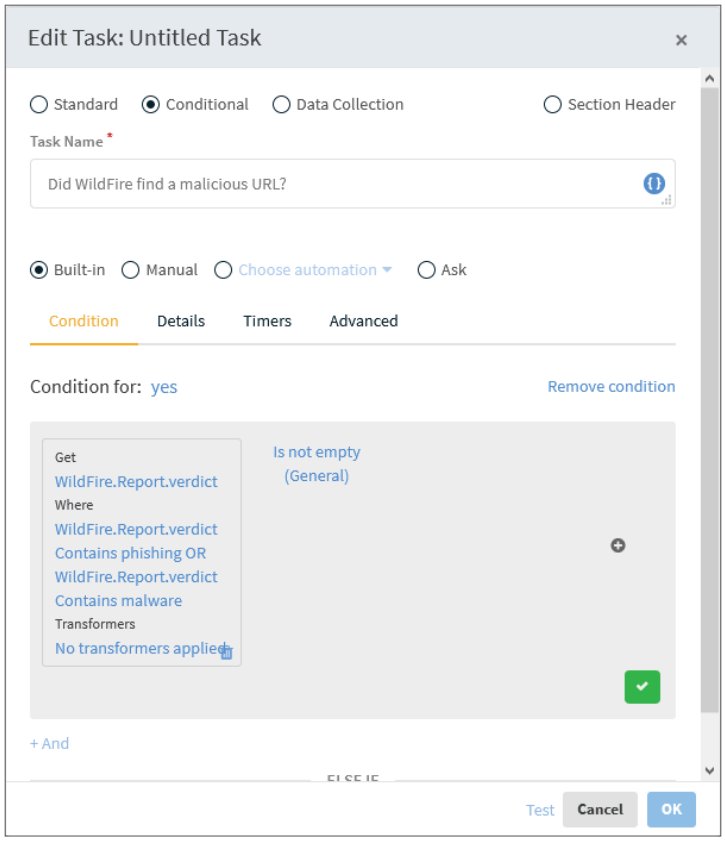
Step 17: Click the check, and then click **OK**.



Next, you choose the comparison operator for the condition.

Step 18: Click **Equals**.

Step 19: In the search box, enter **Is not empty**, and then click **Is not empty**.



Step 20: Click the check, and then click **OK**.

Step 21: Verify that the task is now in your playbook.

2.11 Create the "Mark as Note - Malicious URL Detected" Task

This is the first task of a new branch of the playbook. The playbook selects this branch only if the WildFire report includes any phishing or malware verdicts, as determined by Procedure 2.10.

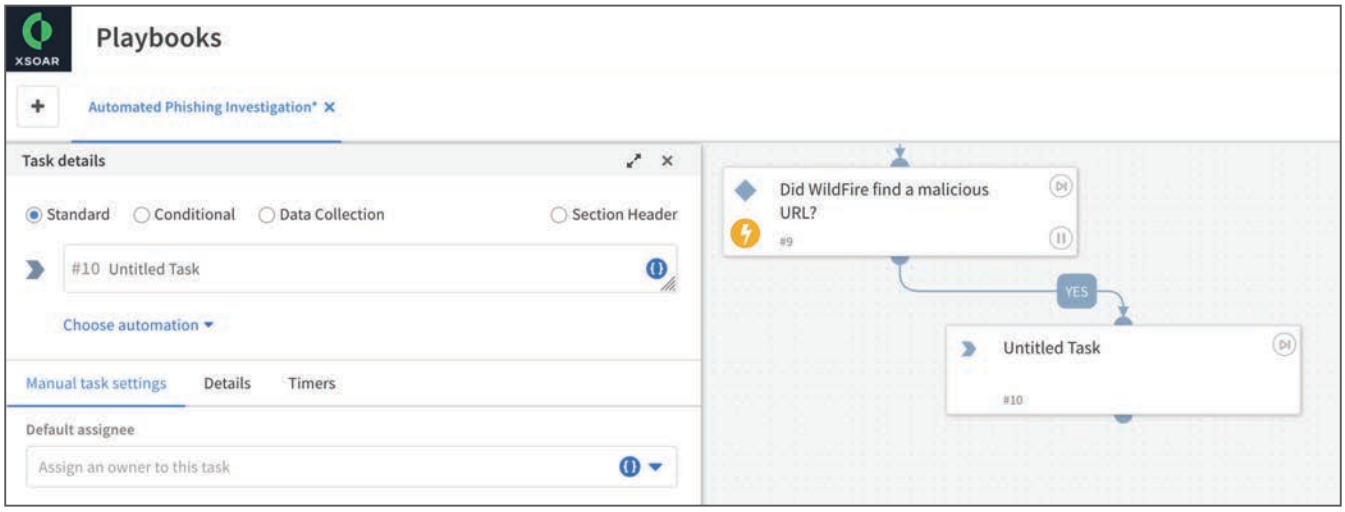
In this task, you parse the WildFire report to identify the URLs that returned a WildFire verdict of "phishing" or "malware." As in Procedure 2.10, this task uses the WildFire.Report context data. You first filter the verdicts to only include those with a "phishing" or "malware" match, and then you return the corresponding WildFire.Report.URL.

This task uses the **Print** automation script. You configure advanced settings for this task to mark the results as an incident note.

Step 1: From the Did WildFire find a malicious URL? task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below and to the right.

Step 2: In the Choose Label Name for Condition dialog box, select **yes**.

Step 3: Click **Save**. The Edit Task dialog box appears.



Step 4: In the box with the placeholder **Untitled Task**, enter Mark as note - malicious URL detected.

Step 5: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 6: In the **search** box, enter **Print**, and then choose **Print**. The task fields update.

Step 7: In the **value** box, click the {} button. The Select Source for Value dialog box appears.

Step 8: In the search box, enter **WildFire.Report.URL**.

Step 9: In the row for URL, click **Filter & transform**. The dialog box name changes to Filter & Transformers for Value.

Step 10: In the Filter section, click **Add filter**.

Step 11: In the conditional statement left-side box, enter **WildFire.Report.verdict**. This value is case sensitive.

Next, you choose the comparison operator for the filter.

Step 12: Click **Equals**.

Step 13: In the search box, enter **Contains**, and then click **Contains**.

Step 14: In the right-side box, enter **phishing**. This value is case sensitive.

Step 15: To add a second filter condition, click the +.

Step 16: In the conditional statement left-side box, enter **WildFire.Report.verdict**. This value is case sensitive.

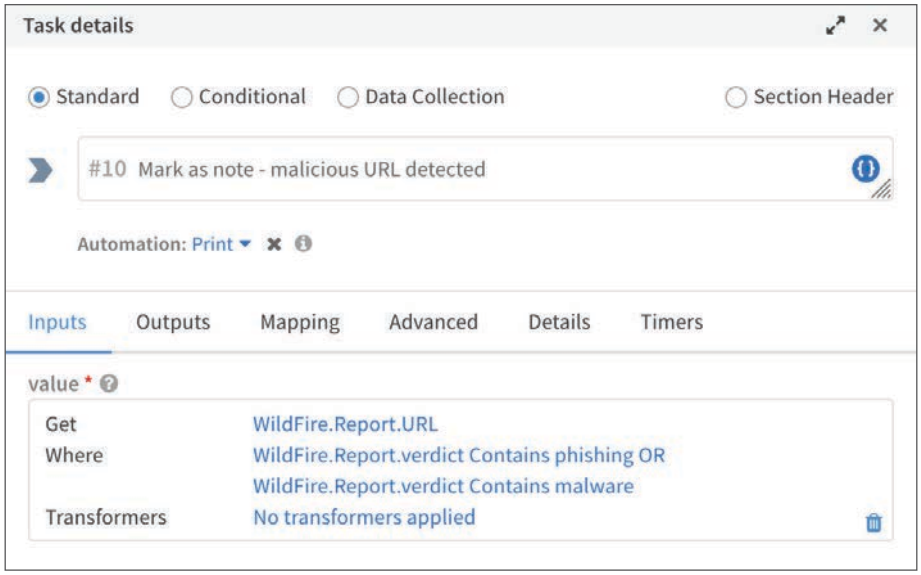
Next, you choose the comparison operator for the filter.

Step 17: Click **Equals**.

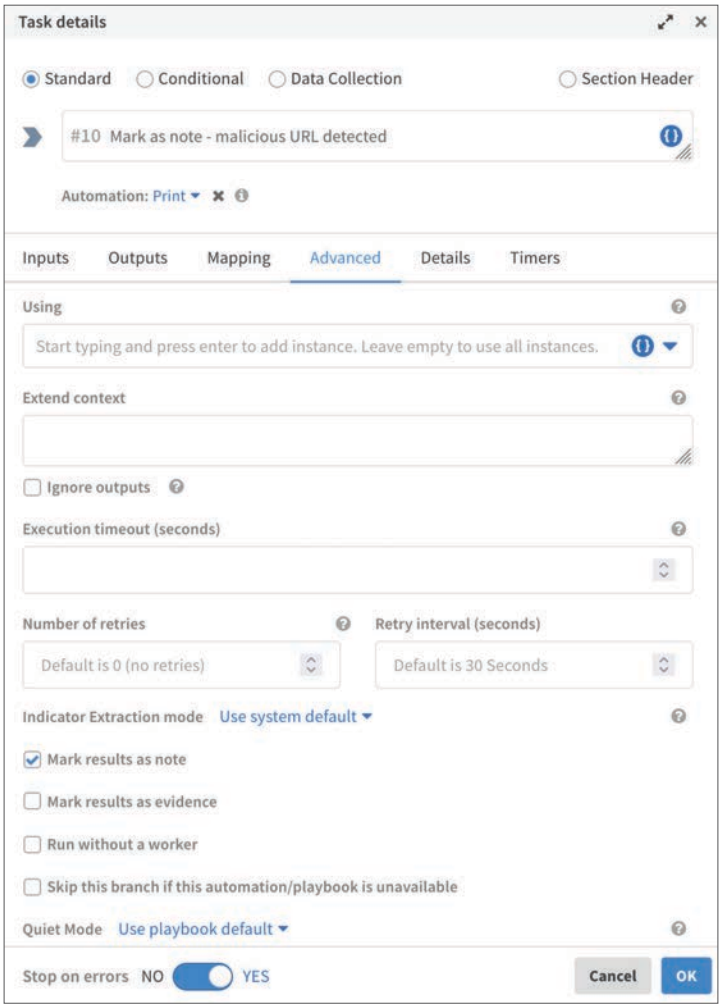
Step 18: In the search box, enter **Contains**, and then click **Contains**.

Step 19: In the right-side box, enter **malware**. This value is case sensitive.

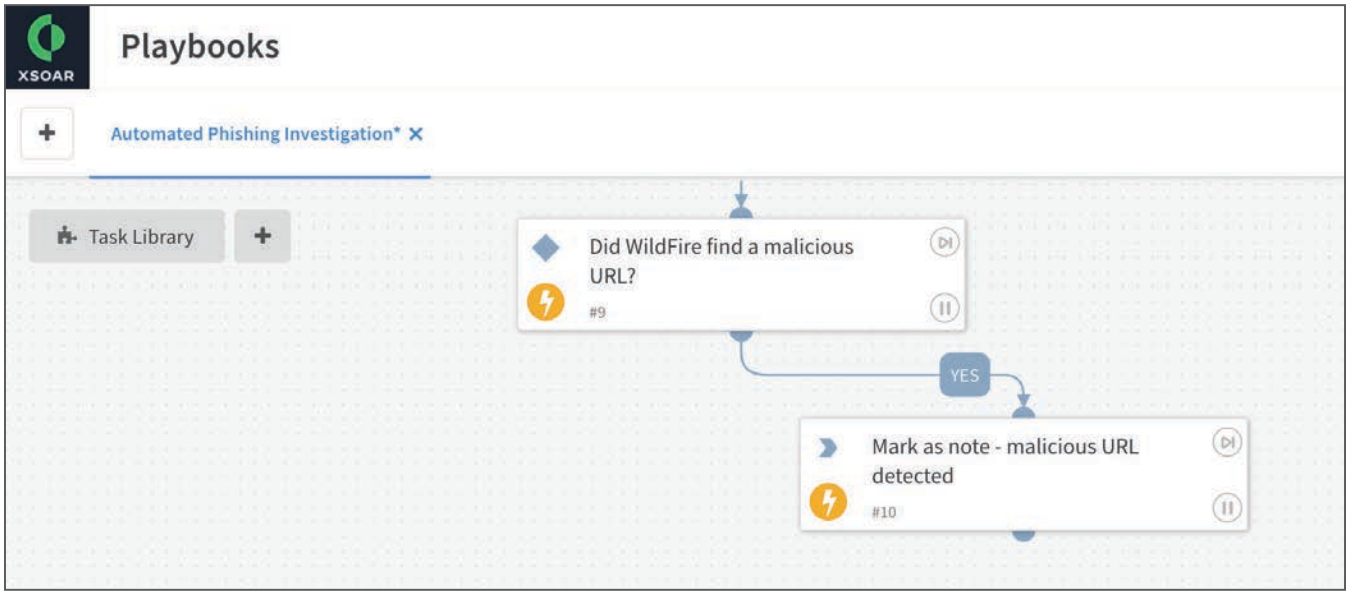
Step 20: Click the check, and then click **OK**.



Step 21: On the Advanced tab, select **Mark results as note**, and then click **OK**.



Step 22: Verify that the task is now in your playbook.



2.12 Create the "Mark as Note - No Malicious URLs Detected" Task

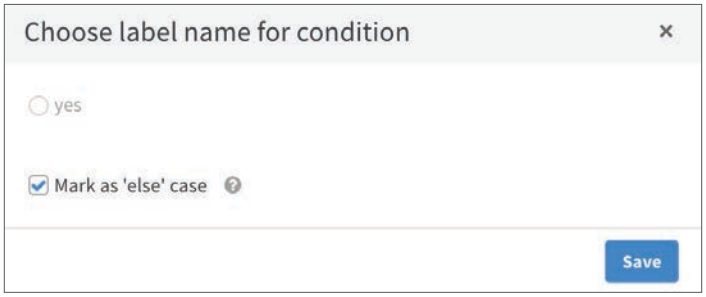
This is the first task of a new branch of the playbook. The playbook selects this branch only if the WildFire report does not include any phishing or malware verdicts, as determined by Procedure 2.10.

You do not have to further examine any context data for this task.

This task uses the **Print** automation script. You configure advanced settings for this task to mark the results as an incident note.

Step 1: From the **Did WildFire find a malicious URL?** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below and to the left.

Step 2: In the Choose Label Name for Condition dialog box, select **Mark as 'else' case**.



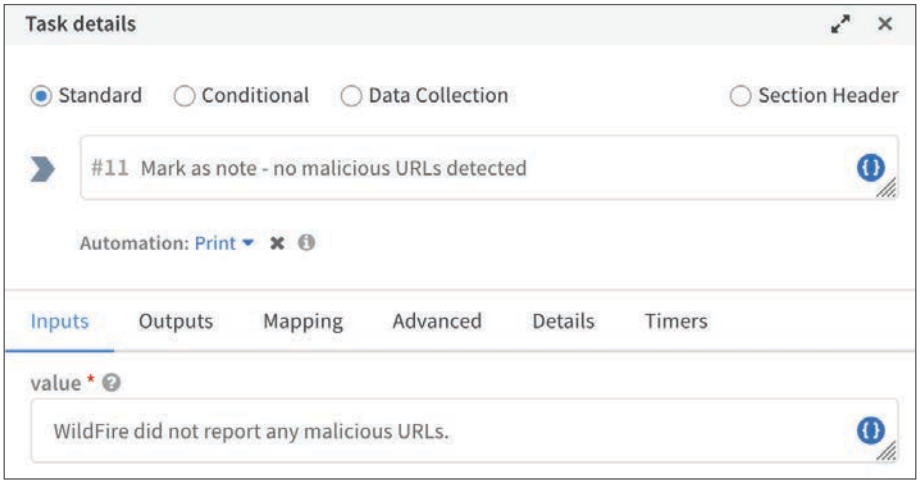
Step 3: Click **Save**. The Edit Task dialog box appears.

Step 4: In the box with the placeholder **Untitled Task**, enter **Mark as note - no malicious URLs detected**.

Step 5: In the Choose Automation section, click the down arrow to open the search dialog box.

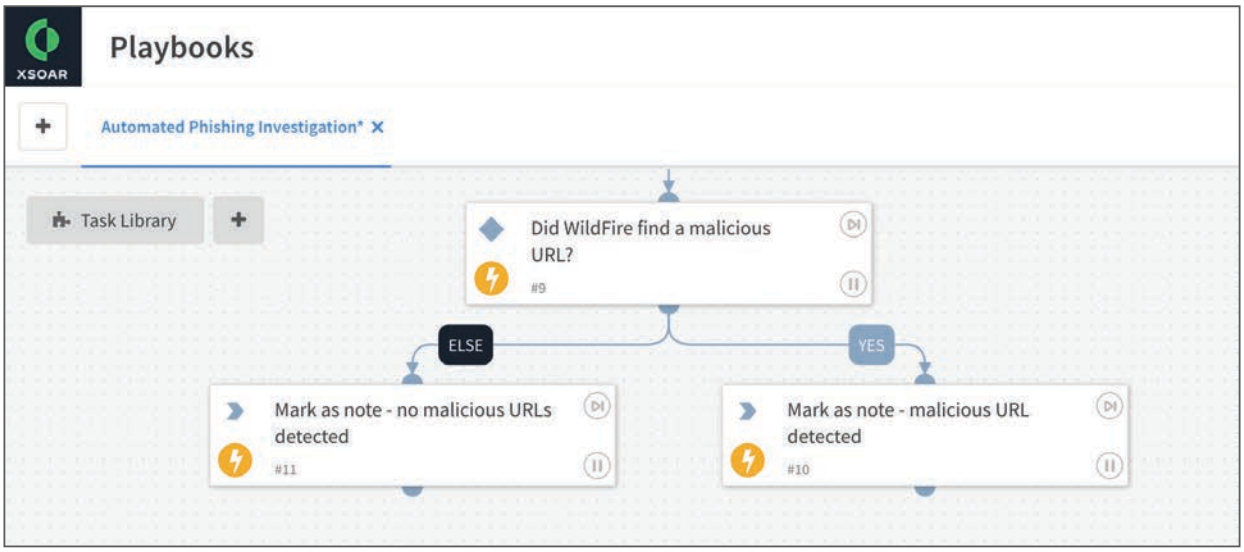
Step 6: In the **search** box, enter **Print**, and then choose **Print**. The task fields update.

Step 7: In the **value** box, enter **WildFire did not report any malicious URLs.**.



Step 8: On the Advanced tab, select **Mark results as note**, and then click **OK**.

Step 9: Verify that the task is now in your playbook.



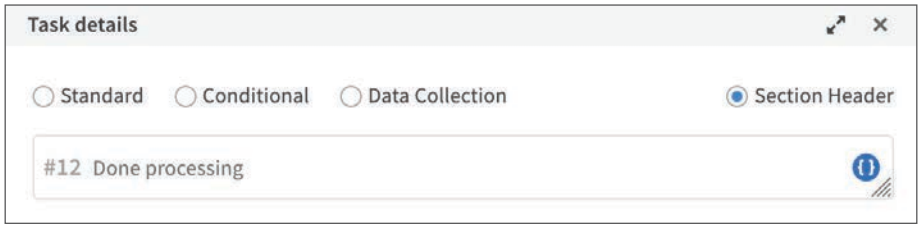
2.13 Create a Separator Task

As a best practice, you should create a Done Processing section header at the end of this section of the playbook.

Step 1: From the Mark as note - malicious URL detected task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled blank task below and to the left. The Edit Task dialog box appears.

Step 2: Select **Section Header**.

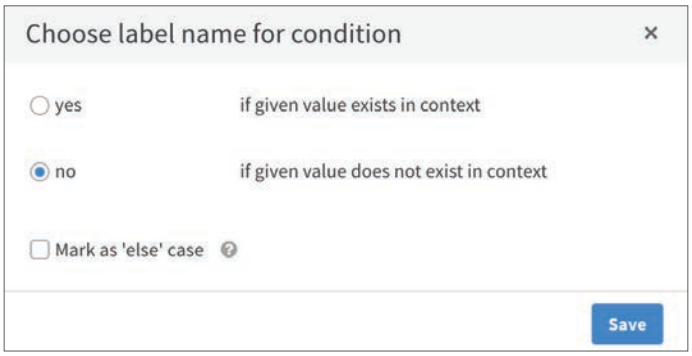
Step 3: In the box with the placeholder **Untitled Task**, enter Done processing, and then click **OK**.



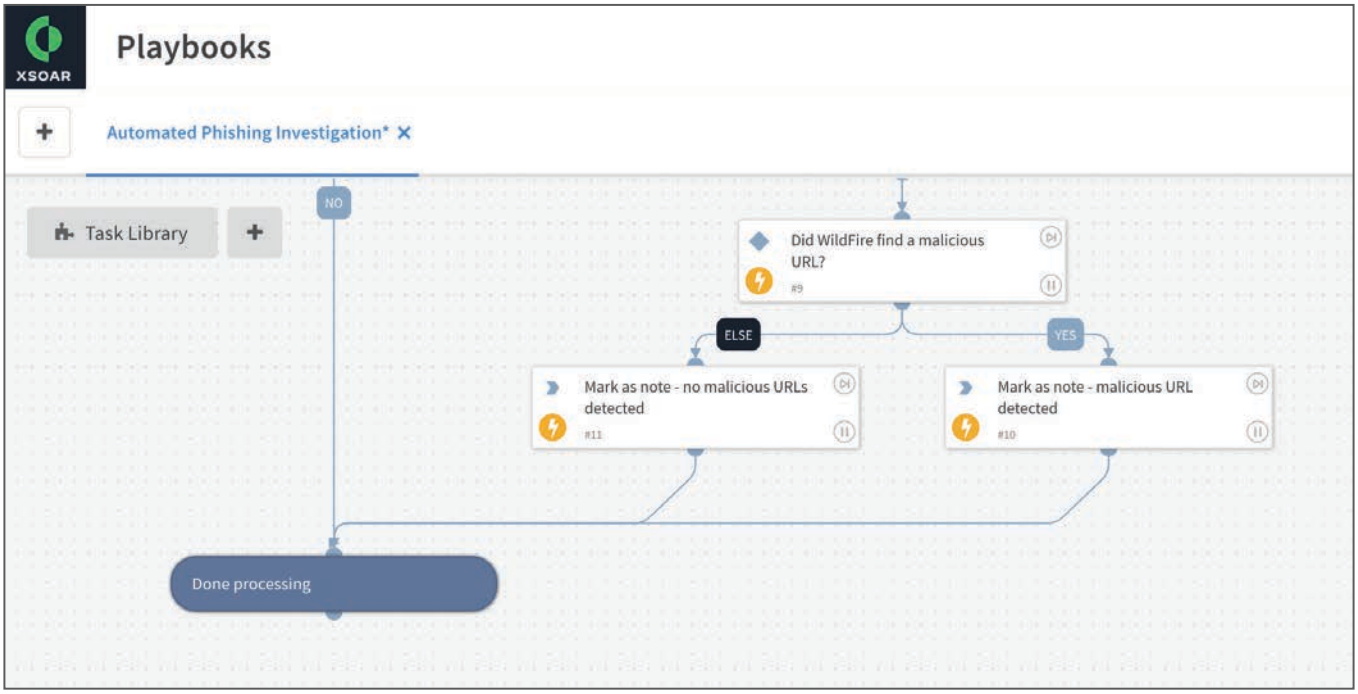
Step 4: From the Mark as note - no malicious URLs detected task egress node, drag the task connector line to the Done Processing task ingress node, and then release to create an additional connection to the Done Processing task.

Step 5: From the **Is message forwarded?** task egress node, drag the task connector line to the **Done Processing** task ingress node, and then release to create an additional connection to the **Done Processing** task.

Step 6: In the Choose Label name for Condition dialog box, select **no**, and then click **Save**.



Step 7: Verify that the task is now in your playbook.



Step 8: To save the playbook, click **Save Playbook**.

Procedures

Running the Playbook and Managing an Incident

To execute the Automated Phishing Investigation playbook, you need to forward a suspected phishing email to your monitored mailbox (example: phishing@example.com).

Earlier in this guide, you configured an EWS v2 integration instance and selected the option to fetch incidents. Based on that configuration, Cortex XSOAR creates a new incident for every email sent to the monitored mailbox, and the playbook you assign for the phishing incident type runs automatically.

In the following procedures, you first configure the default playbook for the phishing incident type, so that Cortex XSOAR is ready to process events, create incidents, and analyze the incident data. After Cortex XSOAR creates the incident, you can review the incident details and the results for individual tasks.

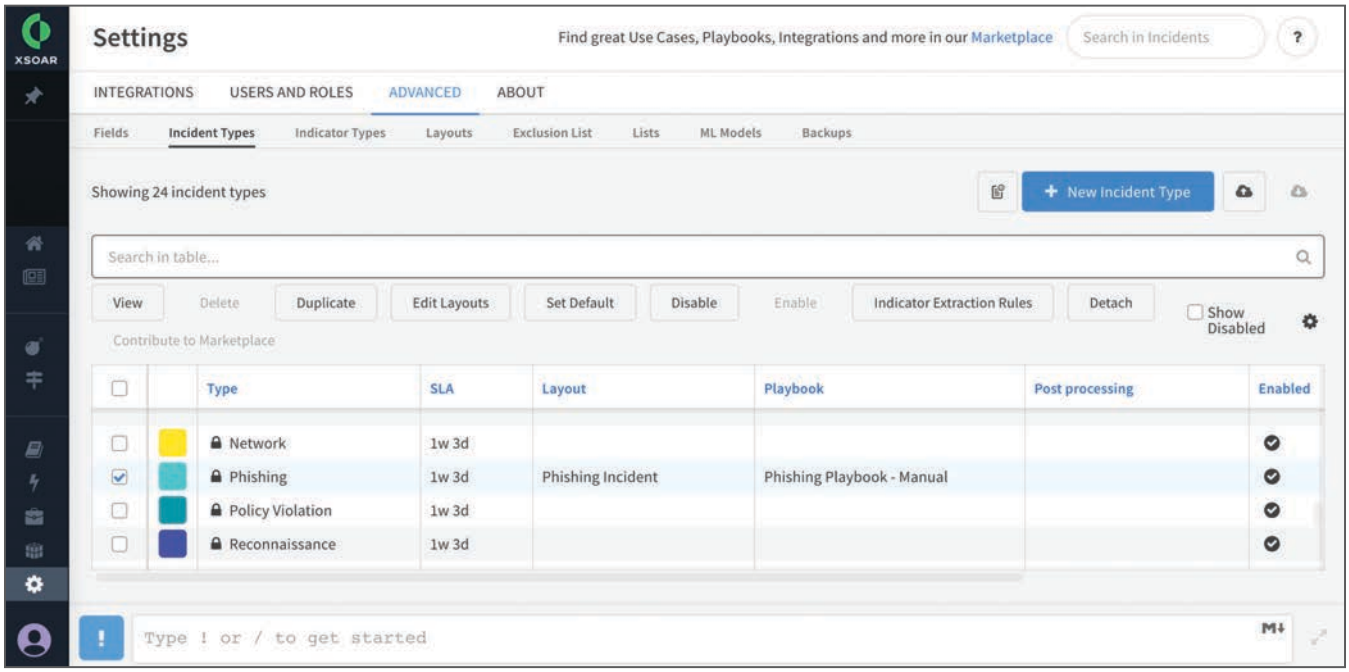
3.1 Assign Playbook for the Incident Type

For each incident type, you may assign a default playbook for Cortex XSOAR to use when Cortex XSOAR creates new incidents. In this procedure, you assign the **Automated Phishing Incident** playbook as the default playbook for phishing incidents and configure the playbook to run automatically.

When you assign this playbook, you also configure Cortex XSOAR to automatically extract and enrich indicators before the playbook runs. When you choose this configuration option, Cortex XSOAR populates context data objects that are available for use by automation scripts and commands within the playbook.

Step 1: In the navigation pane, click **Settings**.

Step 2: In **Advanced > Incident Types,** select the **Phishing** row, and then click **Detach**.



Step 3: In the Detach incident type dialog box, click **Detach**.

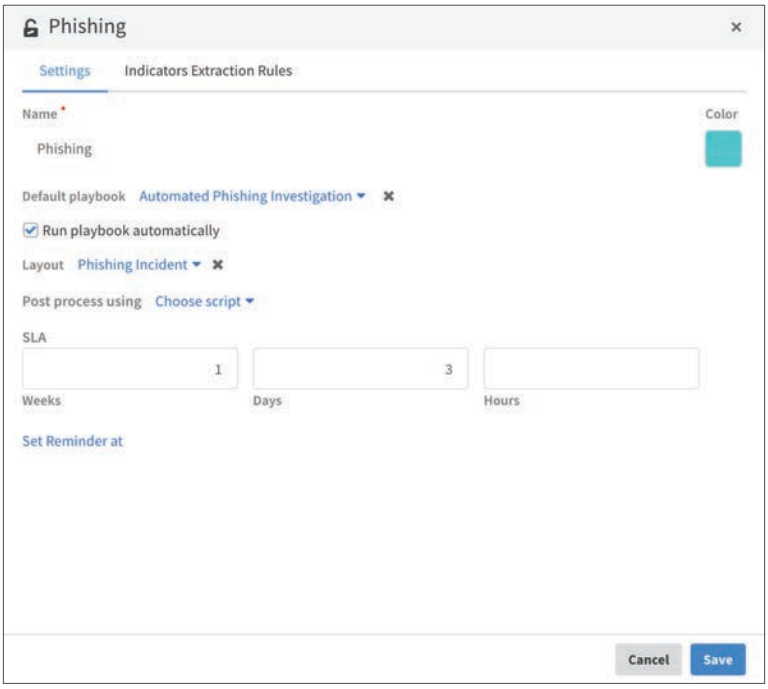
> **Note**
>
> You cannot directly modify a built-in incident type. To make changes, you must detach it first. While an incident type is detached, updates are not applied to the incident type. If you reattach the incident type after making changes, any subsequent updates applied to the incident type undo your changes.

Step 4: Select the **Phishing** row, and then click **Edit**. The Edit Incident Type dialog box appears.

Step 5: In the **Default playbook** box, click the down arrow to open the search dialog box.

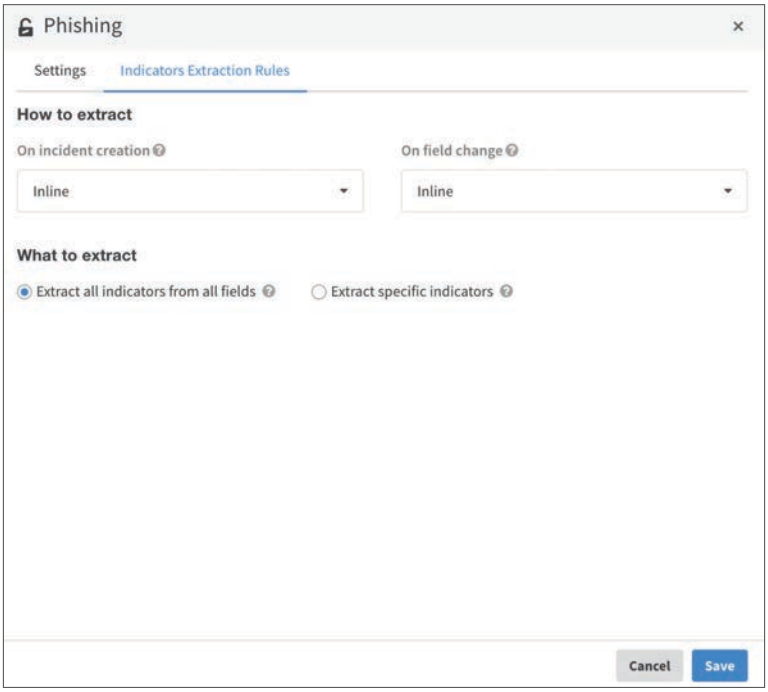
Step 6: In the **search** box, enter **Automated Phishing Investigation**, and then choose **Automated Phishing Investigation**.

Step 7: Select **Run playbook automatically**, and then click **Save**.



Step 8: On the Indicators Extraction Rules tab, in the On Incident Creation section, choose **Inline**.

Step 9: Select **Extract all indicators from all fields**, and then click **Save**.



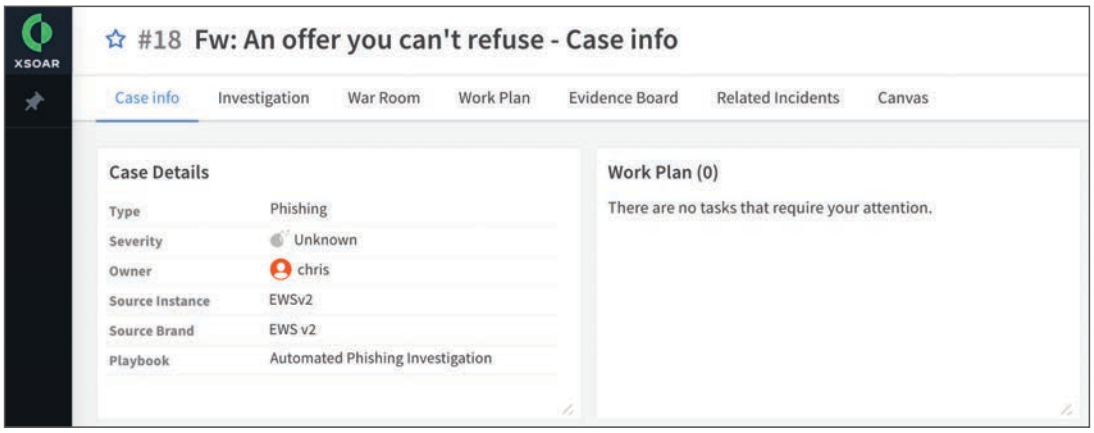
3.2 Running the Playbook and Reviewing an Incident

Based on the EWS v2 Integration instance that you created in Procedure 1.6, Cortex XSOAR checks the monitored mailbox every minute. Cortex XSOAR creates a new incident for each email message that it retrieves. Using the playbook settings you configured in Procedure 3.1, Cortex XSOAR automatically runs the Automated Phishing Investigation playbook after Cortex XSOAR creates the incident.

After Cortex XSOAR creates the new incident, your first action as an analyst is to review the incident details.

Step 1: In the navigation pane, click **Incidents**. By default, Cortex XSOAR displays open incidents created within the last 7 days. You may need to expand the time interval.

Step 2: In the incident list, in the row for the incident you choose to review, click the ID (example: #18). The incident dialog box appears.

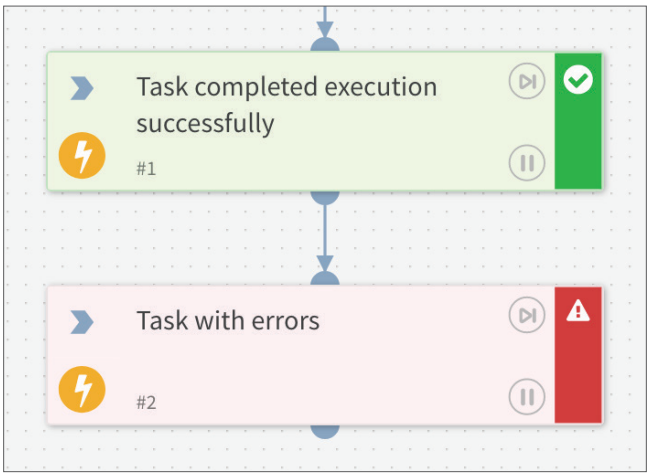


Step 3: If your incident includes notes or evidence, review them.

Step 4: On the Investigation tab, review the indicators that Cortex XSOAR has observed while analyzing the incident.

Type	Value	Verdict	First Seen		Last Seen		Source Time Stamp	Related Incidents
Domain	crushus.com	Malicious	August 9, 2021 12:12 PM	1@12	August 9, 2021 4:49 PM	3@25	August 9, 2021, 12:12 PM	6
Email	amy@example.com	Unknown	August 9, 2021 12:12 PM	1@12	August 9, 2021 4:49 PM	3@25	August 9, 2021, 12:12 PM	13
Domain	sn6pr08mb4653.namprd08.prod.outlook.com	Benign	August 9, 2021 12:12 PM	1@12	August 9, 2021 4:49 PM	3@25	August 9, 2021, 12:12 PM	13
Domain	dm6pr08mb5804.namprd08.prod.outlook.com	Benign	August 9, 2021 12:12 PM	1@12	August 9, 2021 4:49 PM	3@25	August 9, 2021, 12:12 PM	13
Domain	tpb.crushus.com	Malicious	August 9, 2021 12:12 PM	1@12	August 9, 2021 4:49 PM	3@25	August 9, 2021, 12:12 PM	6

Step 5: On the workplan tab, verify that all playbook tasks completed execution without errors. Successful tasks display a green check. Tasks with errors display a red caution symbol.



Step 6: On the War Room tab, review task results. The war room provides a complete record of all activities related to the incident.

Procedures

Preparing and Generating Custom Reports

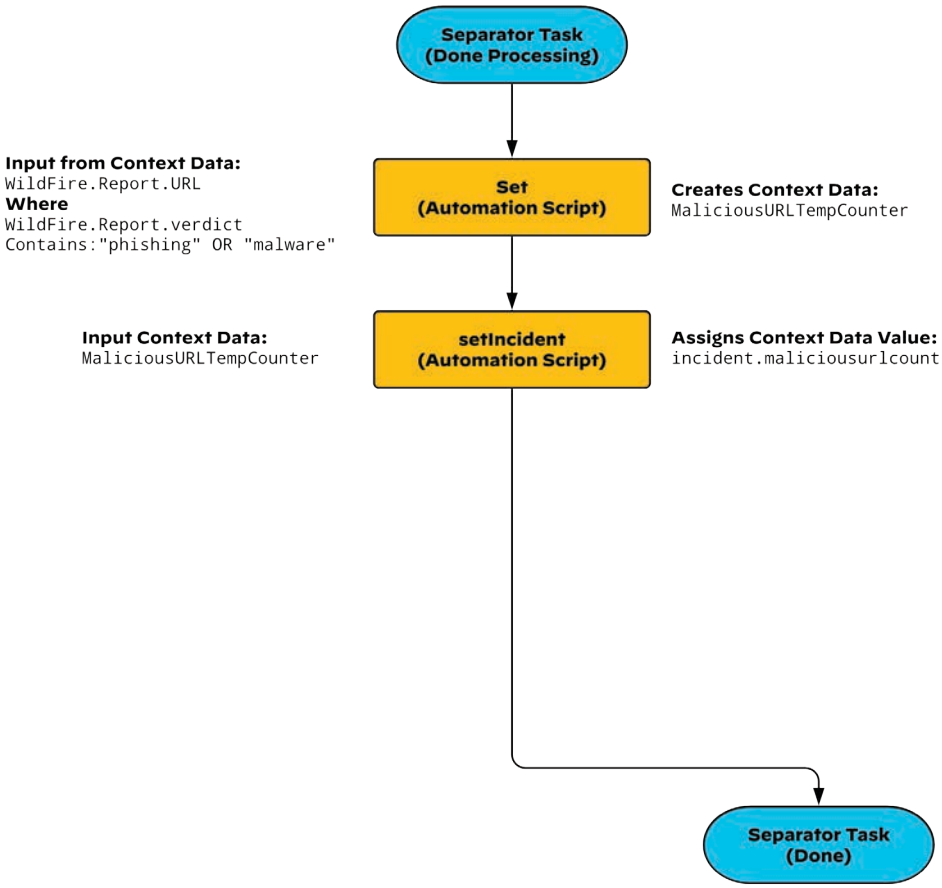
You use these procedures to create a Cortex XSOAR custom report that summarizes the results of multiple phishing incidents. The report includes two summary tables.

The first table includes a list of phishing incidents, with the status of each incident, the end user who originally forwarded the suspected phishing email, and the total number of malicious URLs identified by Cortex XSOAR. The second table includes all of the indicators with a “Malicious” verdict that are associated with phishing incidents.

To include data values from your incident in an incident summary, you need to assign the data values to an incident field. In the following procedures, you define a Cortex XSOAR custom field that applies to your incident type and then in your playbook, you assign a value to the field. After you define the custom field, you can include it in a custom widget for your summary report.

You modify your existing playbook by creating additional tasks that set the incident fields that Cortex XSOAR uses to generate the custom reports.

Figure 2 Modifications to the Automated Phishing Investigation playbook



4.1 Create a New Incident Field

In this procedure, you create a new incident field that you use in phishing incidents to track the number of malicious URLs reported by Cortex XSOAR.

Step 1: Log in to the Cortex XSOAR portal (example: https://xsoar.example.com).

Step 2: In the navigation pane, click **Settings**.

Step 3: In **Advanced > Fields**, click **New Field**. The New Incident Field dialog box appears.

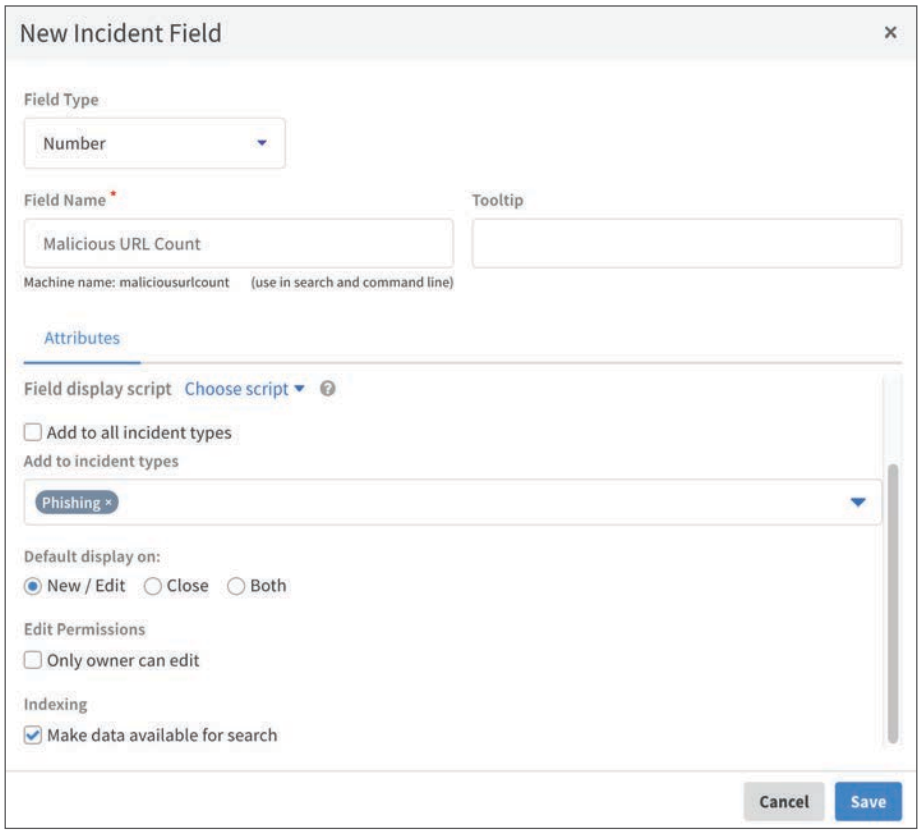
Step 4: In the **Field Type** box, choose **Number**.

Step 5: In the Field Name box, enter **Malicious URL Count**.

Step 6: Note the Machine Name value that Cortex XSOAR assigns (example: **maliciousurlcount**).

Step 7: Clear **Add to all incident types**. The Add to Incident Types section appears.

Step 8: In the **Add to incident types** list, choose **Phishing**, and then click **Save**.



4.2 Create the "Set Malicious URL Temporary Counter" Task

In this procedure, you choose the playbook you previously created in Procedure 2.1 and switch to edit mode. You can then proceed with creating additional tasks.

Before you can set the value for the incident field Malicious URL Count, you must filter and transform the WildFire report data from Procedure 2.9 and store this value using the temporary variable *MaliciousURLTempCounter*.

This task uses the **Set** automation script.

Step 1: In Cortex XSOAR, in the navigation pane, click **Playbooks**.

Step 2: In the search box, enter **Automated Phishing Investigation**, and then press ENTER.

Step 3: In the search results, click **Automated Phishing Investigation**.

Step 4: Click **Edit**.

Step 5: From the **Done Processing** section header egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

Step 6: In the box with the placeholder **Untitled Task**, enter **Set malicious URL temporary counter**.

Step 7: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 8: In the **search** box, enter **Set**, and then choose **Set**. The task fields update.

Step 9: In the **key** box, enter **MaliciousURLTempCounter**.

Step 10: In the value box, click the {} button. The Select Source for Value dialog box appears.

Step 11: In the search box, enter **WildFire.Report.URL**.

Step 12: In the row for URL, click **Filter & transform**. The dialog box name changes to Filter & Transformers for Value.

Step 13: In the Filter section, click **Add filter**.

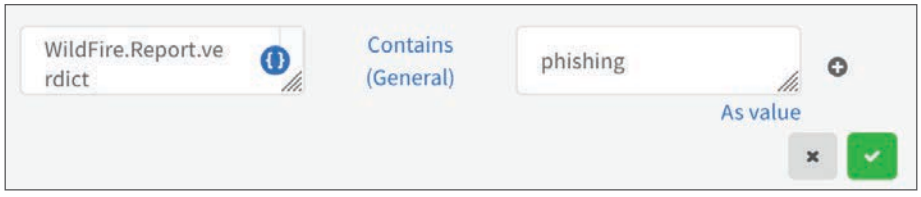
Step 14: In conditional statement left-side box, enter **WildFire.Report.verdict**. This value is case sensitive.

Next, you choose the comparison operator for the filter.

Step 15: Click **Equals**.

Step 16: In the search box, enter **Contains**, and then click **Contains**.

Step 17: In the right-side box, enter **phishing**. This value is case sensitive.



Step 18: To add a second filter condition, Click the +.

Step 19: In the conditional statement left-side box, enter **WildFire.Report.verdict**. This value is case sensitive.

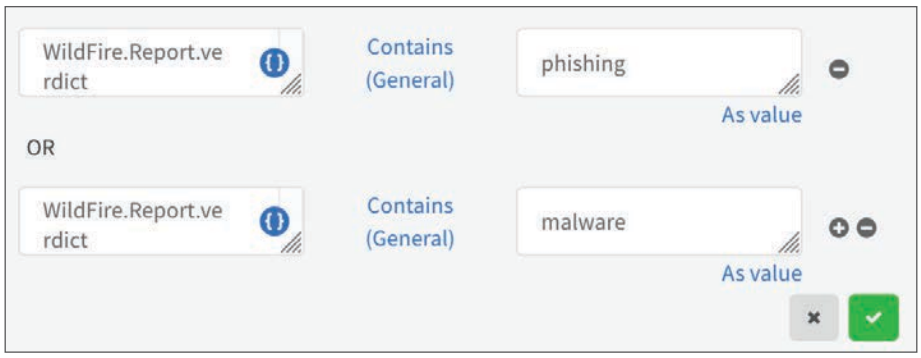
Next, you choose the comparison operator for the filter.

Step 20: Click **Equals**.

Step 21: In the search box, enter **Contains**, and then click **Contains**.

Step 22: In the right-side box, enter **malware**. This value is case sensitive.

Step 23: Click the check.



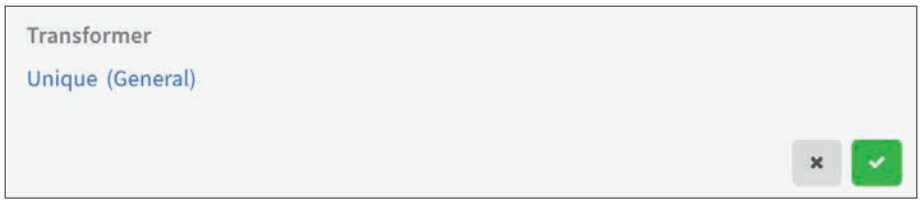
Now you transform the filtered URLs. First, you eliminate duplicates using the *unique* transformer.

Step 24: In the Apply Transformers on the Field section, click **Add Transformer**.

Step 25: Click **To upper case**.

Step 26: In the search box, enter **Unique** and then click **Unique**.

Step 27: Click the check.



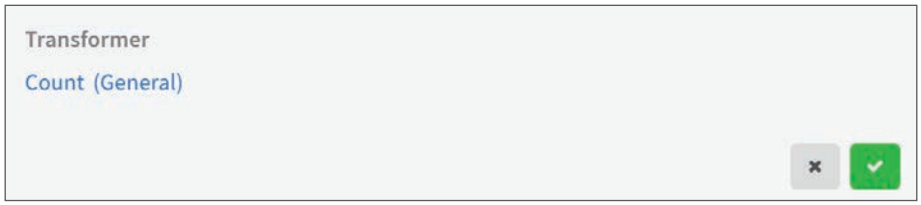
Next, you count the number of unique URLs using the *count* transformer.

Step 28: In the Apply Transformers on the Field section, click **Add Transformer**.

Step 29: Click **To upper case**.

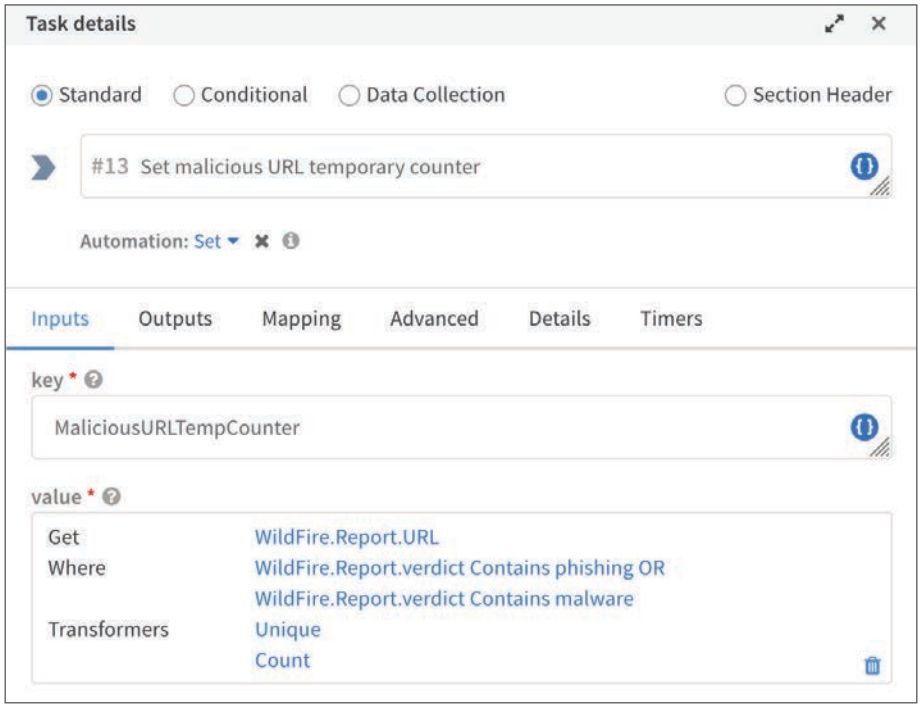
Step 30: In the search box, enter **Count** and then click **Count**.

Step 31: Click the check.

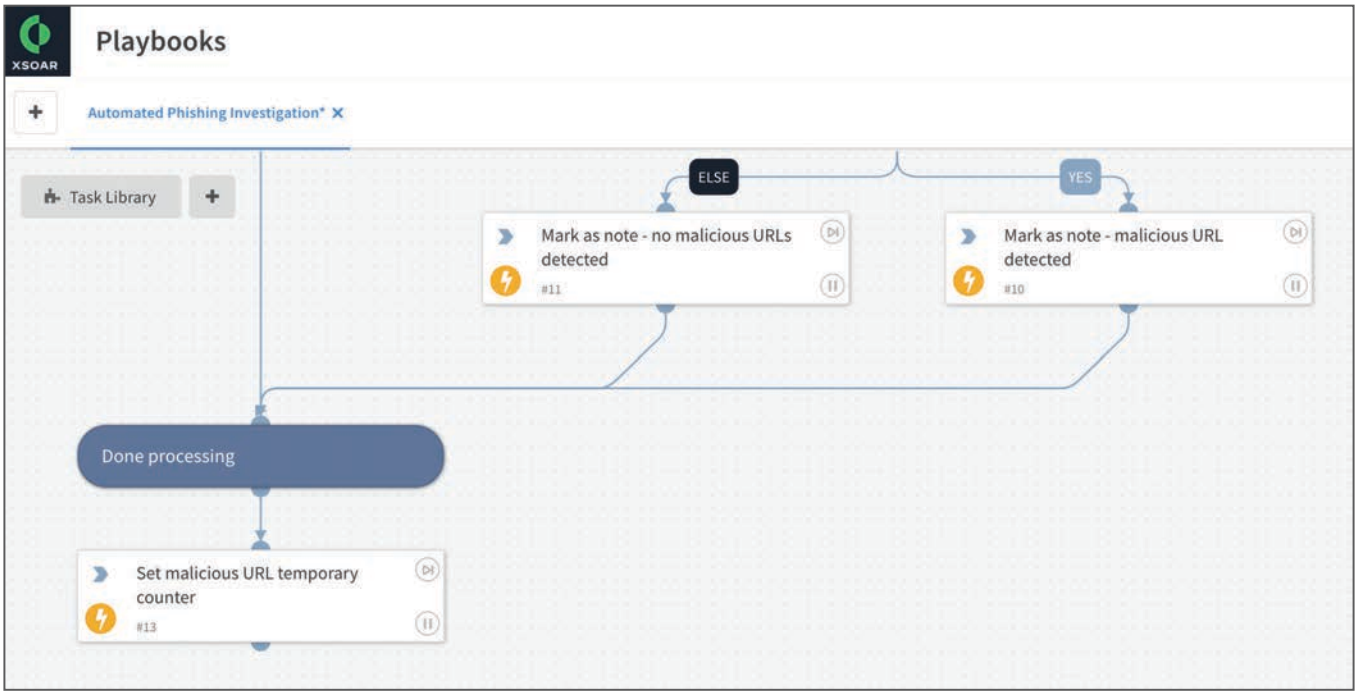


Step 32: Click **OK** to close the Filters & Transformers for Value dialog box.

Step 33: Verify that the **value** box is properly populated, and then click **OK**.



Step 34: Verify that the task is now in your playbook.



4.3 Create the "Set Malicious URL Count" Task

This task uses the **setIncident** built-in automation command to set the value for the custom incident field that you created in Procedure 4.1.

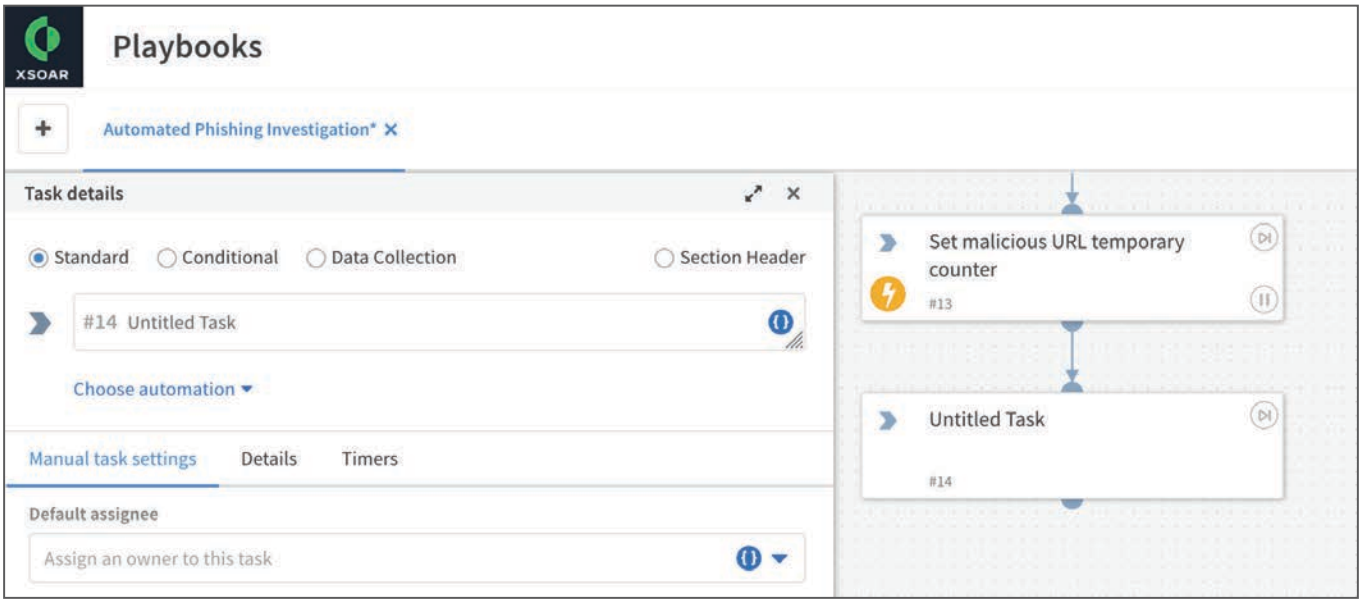
With this automation command, to set or modify a custom incident field, you must use the Machine Name value that Cortex XSOAR assigned when you created the custom field (example: **maliciousurlcount**).

When using the setIncident automation, customFields uses a JSON string format. As an example, to set the value for **incident.maliciousurlcount** to 500, you must use the JSON string {"**maliciousurlcount**":500}.

> **Note**
>
> The setIncident automation does not require that you provide the full incident key when you provide the context.
>
> Correct usage: maliciousurlcount
>
> Incorrect usage: incident.maliciousurlcount

In Procedure 4.2, you transformed the WildFire Report data in order to generate a numerical count of how many unique malicious URLs the phishing email contained. Now you set the incident field to this value. Because you must provide the task input in JSON string format, you cannot also transform the data within the same task.

Step 1: From the **Set malicious URL temporary counter** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.



Step 2: In the box with the placeholder **Untitled Task**, enter **Set malicious URL count**.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

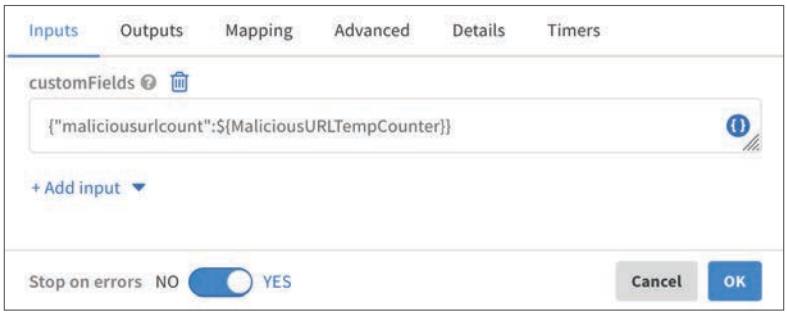
Step 4: In the **search** box, enter **setIncident (Builtin)**, and then choose **setIncident (Builtin)**. The task fields update.

Provide a properly formatted JSON string to set the **Malicious URL Count** field value to the value of **maliciousurltmp**.

Step 5: In the **Add input** section, click the down arrow. The search dialog box opens.

Step 6: In the **search** box, enter **customFields**, and then choose **customFields**.

Step 7: In the **customFields** box, enter **{"maliciousurlcount":${MaliciousURLTempCounter}}**, and then click **OK**.



Step 8: Verify that the task is now in your playbook.

4.4 Create a Separator Task

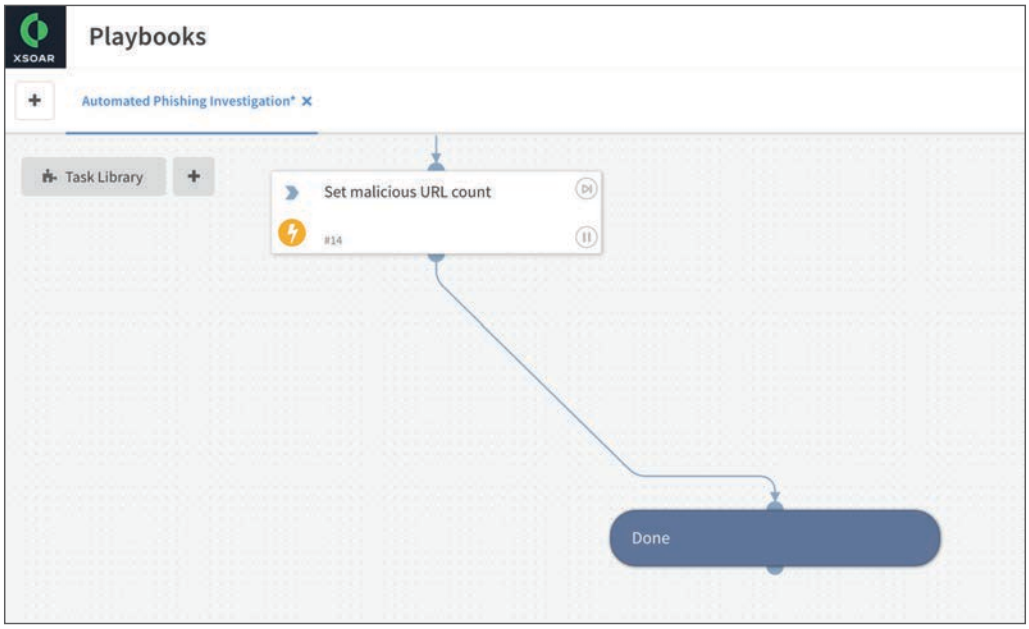
As a best practice, you should create a Done section header that terminates this section of the playbook.

Step 1: From the **Set malicious URL count** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below and to the right. The Edit Task dialog box appears.

Step 2: Select **Section Header**.

Step 3: In the box with the placeholder **Untitled Task**, enter **Done**, and then click **OK**.

Step 4: Verify that the task is now in your playbook.



Step 5: To save the playbook, click **Save Playbook**.

4.5 Create a Custom Report to Summarize Phishing Incidents

To best understand the volume of phishing incidents and the potential impact of these incidents on your organization, you need to summarize data across multiple incidents. Using custom widgets, you can build a custom report to summarize phishing activity over a specified time interval. This example uses a 1-week interval.

The first widget uses a grid format to display all phishing incidents over the time interval. Each row includes incident-specific details that include the end user who initiated the incident and the number of malicious URLs included in the suspected phishing email.

The second widget uses a grid format to display all indicators associated with phishing incidents for which WildFire returned a report with a verdict of "Malicious."

In this procedure, you first create the new widgets and then add them to the report.

Table 2 Widget parameters for Automated Phishing Investigation summary report

Widget name	Data type	Data query	Columns
Malicious URLs per incident	Incidents	type:Phishing	ID, Owner, Email From, Malicious URL Count, Status
Indicators with "Malicious" verdict	Indicators	incident. type:Phishing and type:URL and verdict:Malicious	Type, Value, Verdict, Source Brands, Related Incidents

Step 1: Log in to the Cortex XSOAR portal (example: https://xsoar.example.com).

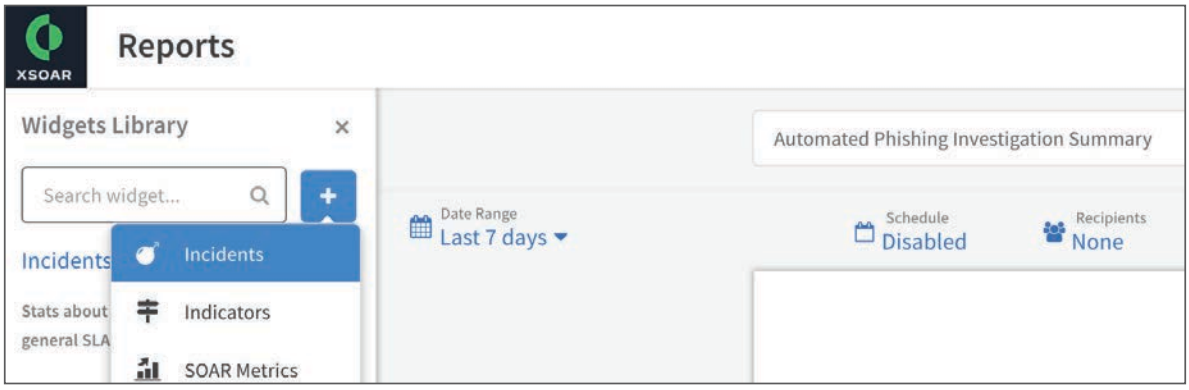
Step 2: In the navigation pane, click **Reports**.

Step 3: Click **New Report**. The Reports page appears.

Step 4: In the Report Name, enter **Automated Phishing Investigation Summary**.

First, you add the custom widgets for your report.

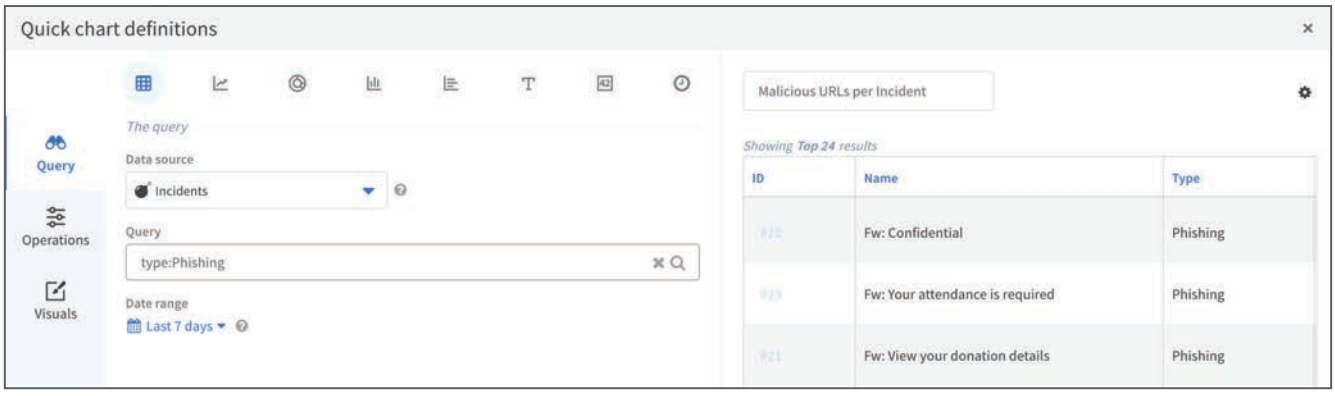
Step 5: In the Widgets Library pane, click **+**, and then click **Incidents**.



Step 6: On the Quick Chart Definitions dialog box, in the **Query** box, enter **type:Phishing**.

Step 7: In the chart style selection pane, click the table icon.

Step 8: In the **Widget name** box, enter **Malicious URLs per Incident**.

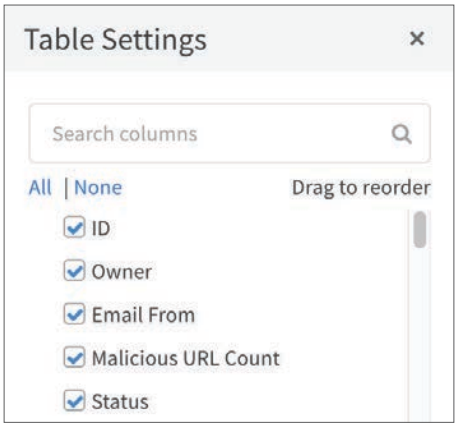


Step 9: To select the grid fields for this widget, click the cog, and then click **Column settings**.

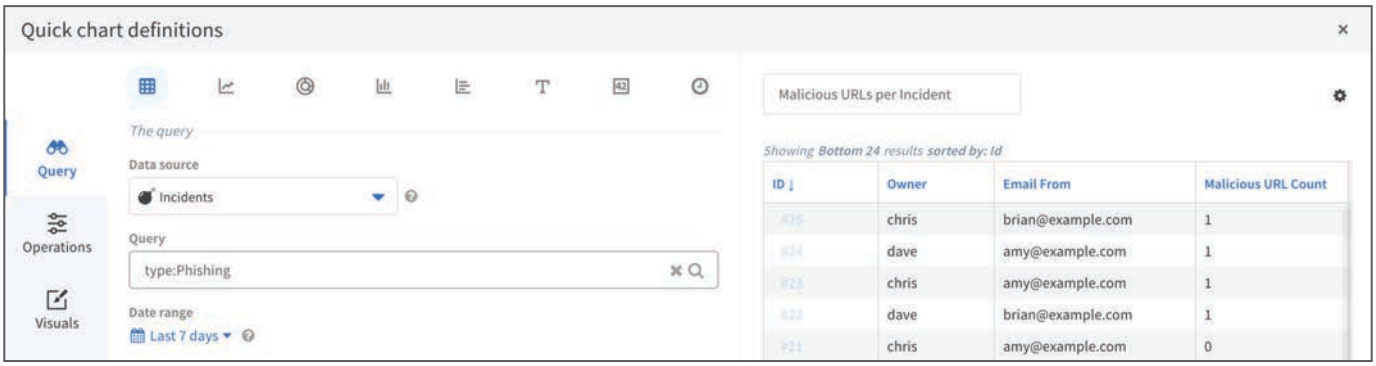
Step 10: On the Table Settings dialog box, to clear all selections, click **None**.

Step 11: On the Table Settings dialog box, select **ID**, **Owner**, **Email From**, **Malicious URL Count**, and **Status**.

Step 12: To reorder the columns, hover over a selection, and then click-and-drag to move the selection up or down. There are many available columns, and reordering may take a while.



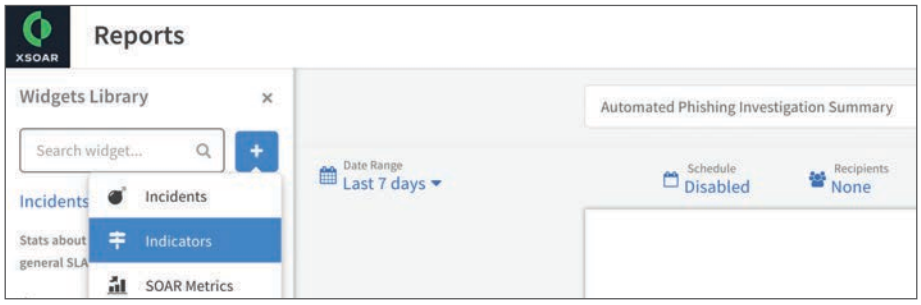
Step 13: After you finish reordering the columns, click **Save**.



Step 14: As necessary, you can revise the widget by adding additional columns.

Step 15: To save your widget to the Widgets Library, click **Save**.

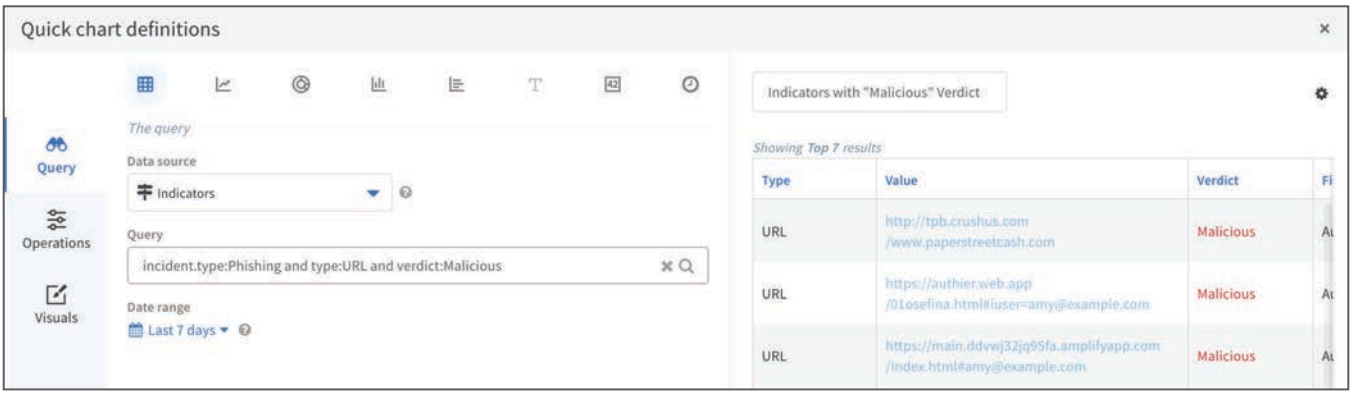
Step 16: In the Widgets Library pane, click **+**, and then click **Indicators**.



Step 17: On the Quick Chart Definitions dialog box, in the **Query** box, enter **incident.type:Phishing and type:URL and verdict:Malicious**.

Step 18: In the chart style selection pane, click the table icon.

Step 19: In the **Widget name** box, enter **Indicators with "Malicious" Verdict**.

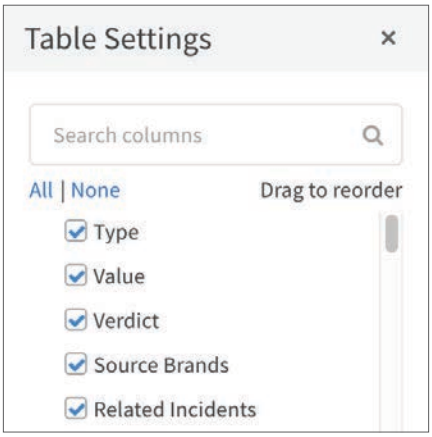


Step 20: To select the grid fields for this widget, click the cog, and then click **Column settings**.

Step 21: On the Table Settings dialog box, to clear all selections, click **None**.

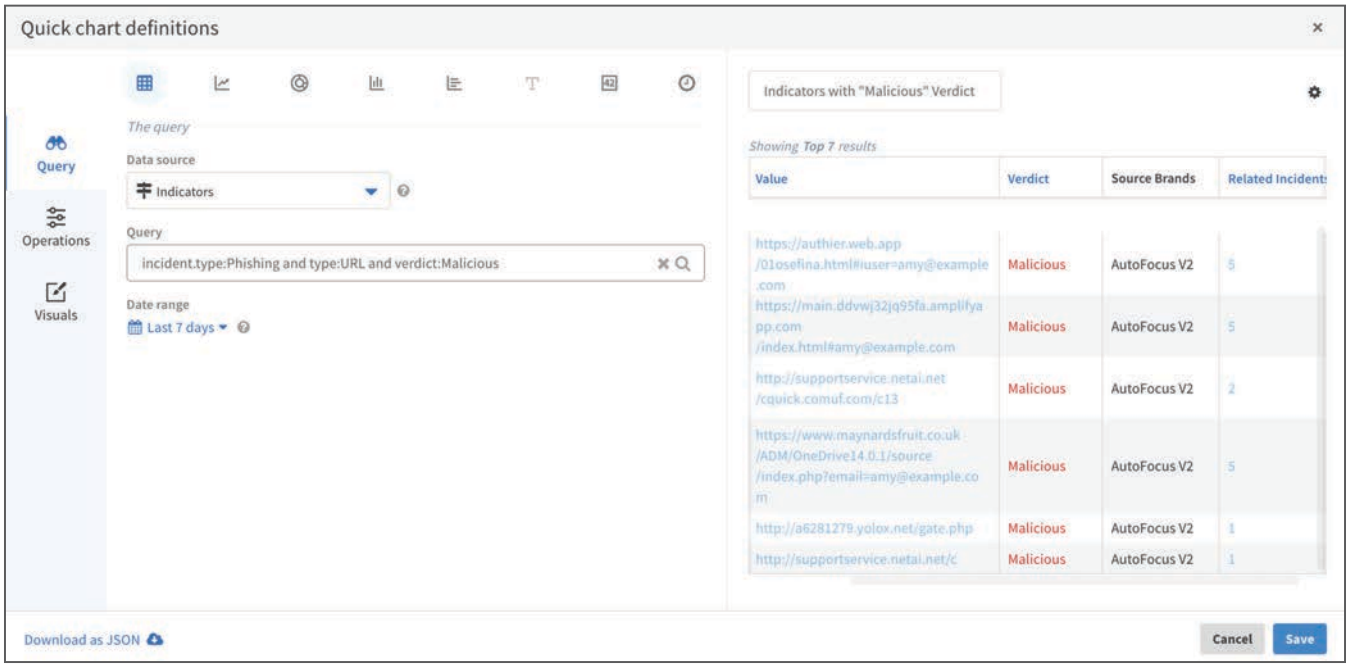
Step 22: On the Table Settings dialog box, select **Type**, **Value**, **Verdict**, **Source Brands**, and **Related Incidents**.

Step 23: To reorder the columns, hover over a selection, and then click-and-drag to move the selection up or down. There are many available columns, and reordering may take a while.



Step 24: After you finish reordering the columns, click **Save**.

Step 25: As necessary, you can revise the widget by adding additional columns.

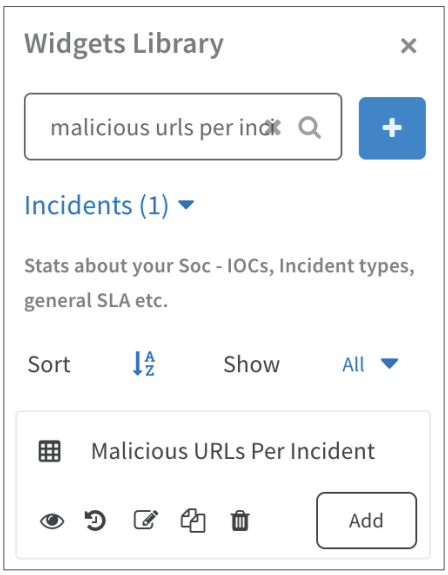


Step 26: To save your widget to the Widgets Library, click **Save**.

Now that you have created the custom widgets, you add them to the report.

Step 27: In the Widgets Library pane, click **Incidents**, and then in the widget type list, choose **Incidents**.

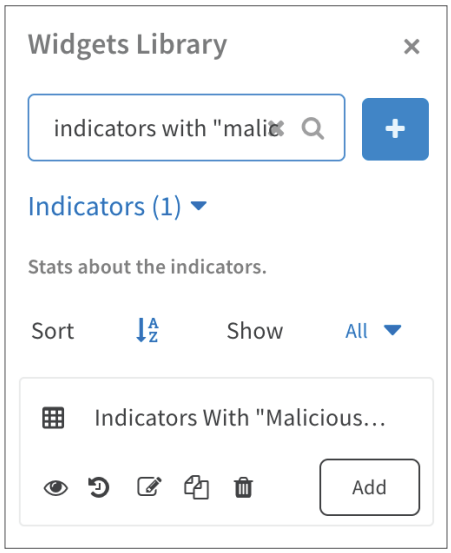
Step 28: In the search box, enter **malicious urls per incident**, and then in the **Malicious URLs Per Incident** widget, click **Add**.



Step 29: To clear the search box, click **x**.

Step 30: In the Widgets Library pane, click **Incidents**, and then in the widget type list, choose **Indicators**.

Step 31: In the search box, enter **Indicators with "Malicious" Verdict**, and then in the **Indicators with "Malicious"Verdict** widget, click **Add**.



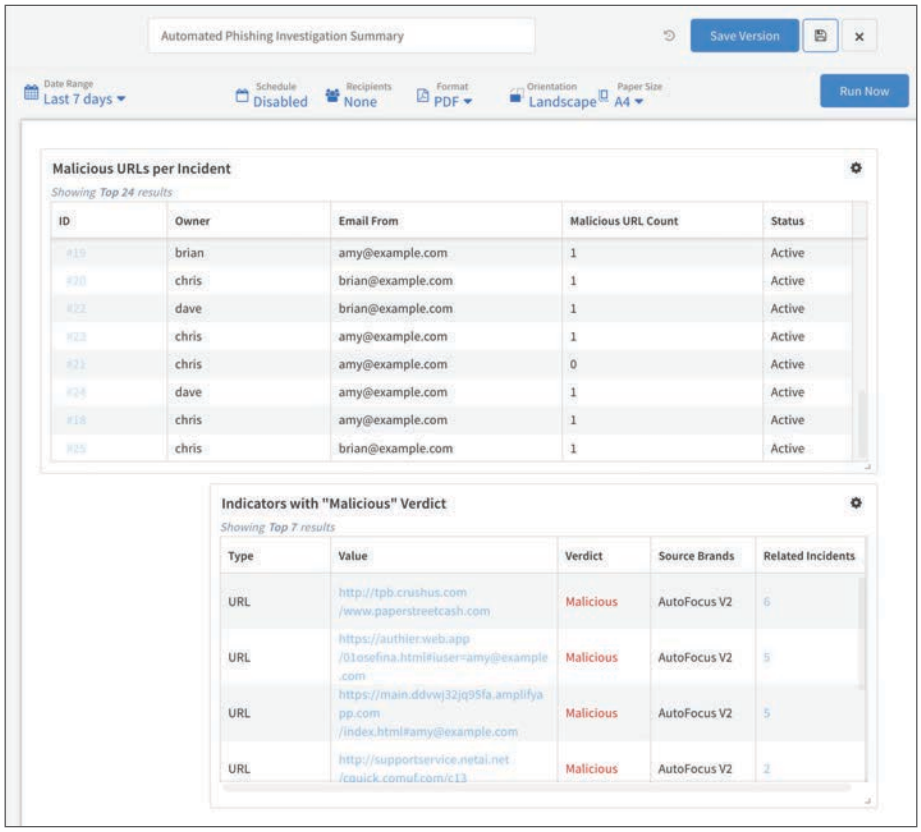
Now that you have added the widgets to the report, you can customize the report layout.

Step 32: In the Orientation section, click **Portrait**. The setting toggles to Landscape.

Step 33: To left justify each widget, click and hold in the title bar, and then drag to the left.

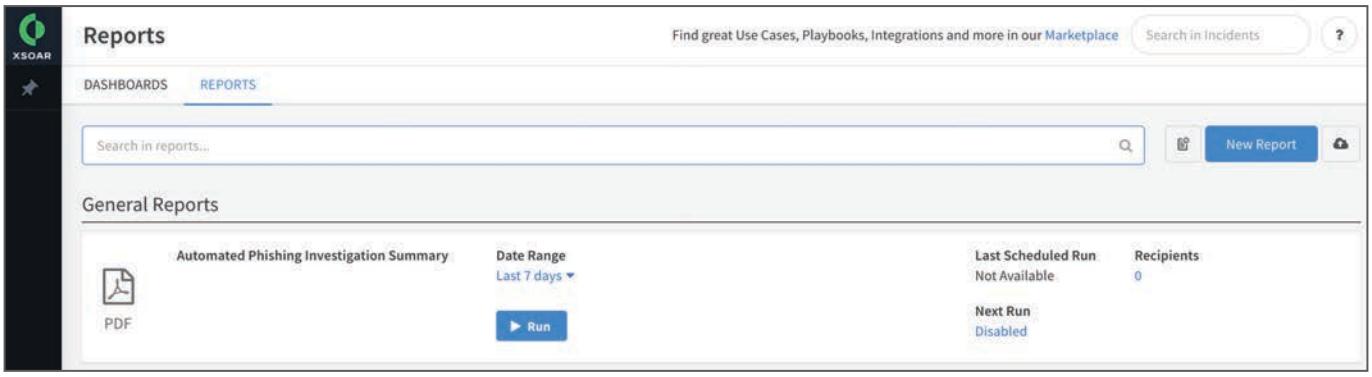
Step 34: To increase the width of each widget, click and hold in the lower-right corner, and then drag to the right.

Step 35: When you are finished adjusting the report layout, click **Save Version**.



Step 36: In the Update Message dialog box, enter a description, and then click **Update**.

Cortex XSOAR adds your custom report to the Reports Library.

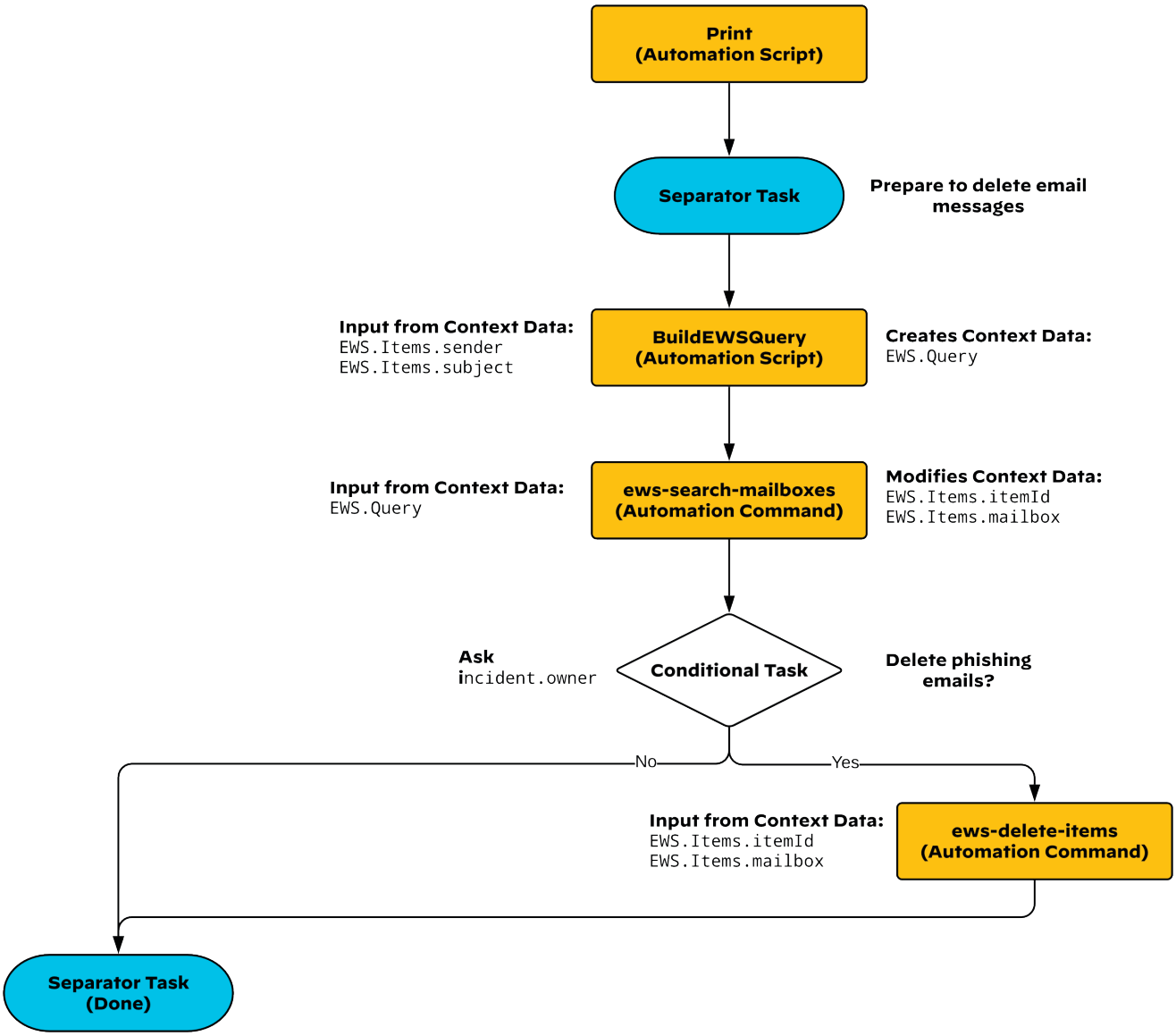


Step 37: To view your completed report, click **Run Now**.

ADDING AUTOMATED RESPONSE TO DELETE EMAILS

In this section, you modify your basic playbook.

Figure 3 Modifications to Automated Phishing Investigation playbook



Procedures

Modifying the Playbook to Automatically Delete Phishing Messages

The basic playbook performs only an automated analysis. If you want to include an automated response option to delete the phishing message from your email system, complete the following procedures.

5.1 Create a Separator Task

In this procedure, you choose the playbook you previously created in Procedure 2.1 and switch to edit mode. You can then proceed with creating additional tasks.

As a best practice, you should create a Prepare to Delete Email Messages section header that begins this section of the playbook.

Step 1: In Cortex XSOAR, in the navigation pane, click **Playbooks**.

Step 2: In the search box, enter **Automated Phishing Investigation**, and then press ENTER.

Step 3: In the search results, click **Automated Phishing Investigation**.

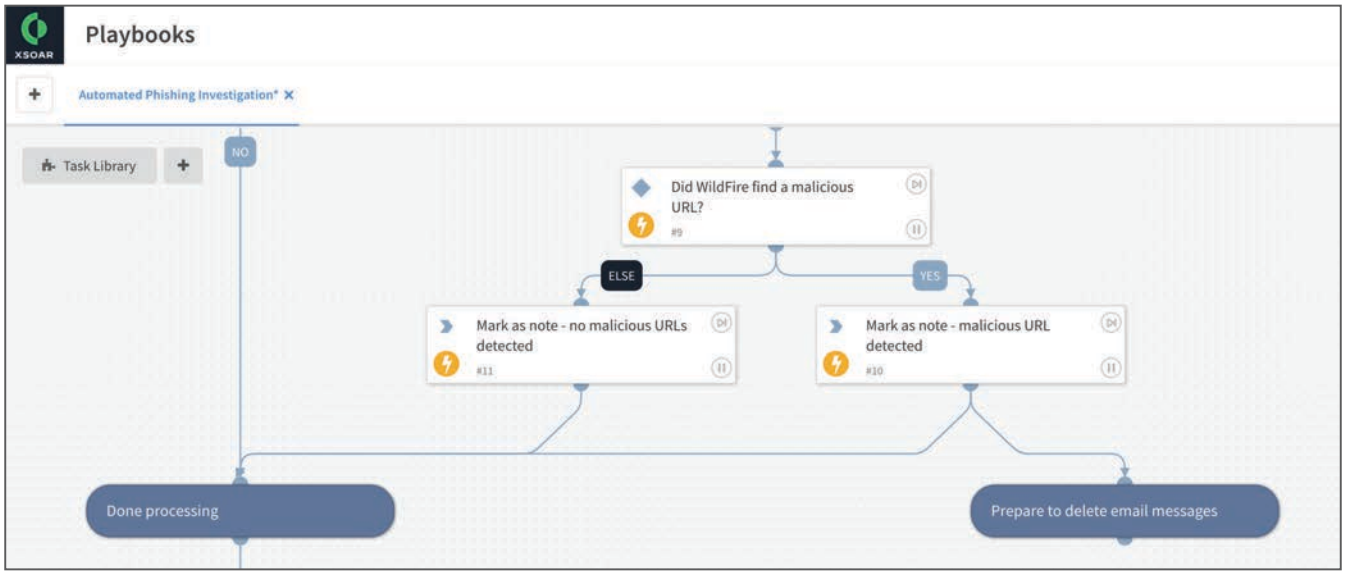
Step 4: Click **Edit**.

Step 5: From the **Mark as note - malicious URL detected** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled blank task below and to the right. The Edit Task dialog box appears.

Step 6: Select **Section Header**.

Step 7: In the box with the placeholder **Untitled Task**, enter **Prepare to delete email messages**, and then click **OK**.

Step 8: Verify that the task is now in your playbook.



5.2 Create the "Build EWS Search Query" Task

To search for the original phishing message, you need to match email header fields from the forwarded email message that triggered this incident.

Before you can search for the phishing message on your email system using the **ews-search-mailboxes** automation command, you need to properly format the query that this automation command requires. In this procedure, you use the **BuildEWSQuery** automation script to build a query to search for messages on the system that match the forwarded message. In a following procedure, you use this query to perform the search.

Your search needs to find emails that match both the sender and subject fields of the original message. By default, this automation script searches only for emails received within the last week. If you need to search over a longer interval, you should modify this search option.

In Procedure 2.4, you retrieved the original message and Cortex XSOAR added *EWS* context data. From this context data, you use *EWS.Items.sender* and *EWS.Items.subject* to build your query.

After you have run this automation script, Cortex XSOAR adds *EWS.Query* context data.

Step 1: From the **Prepare to delete email messages** section header egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

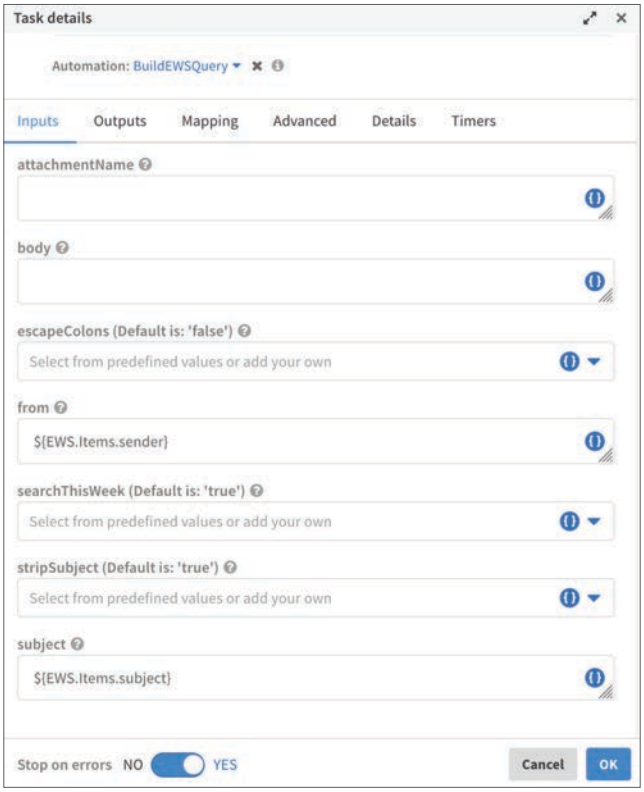
Step 2: In the box with the placeholder **Untitled Task**, enter **Build EWS search query**.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

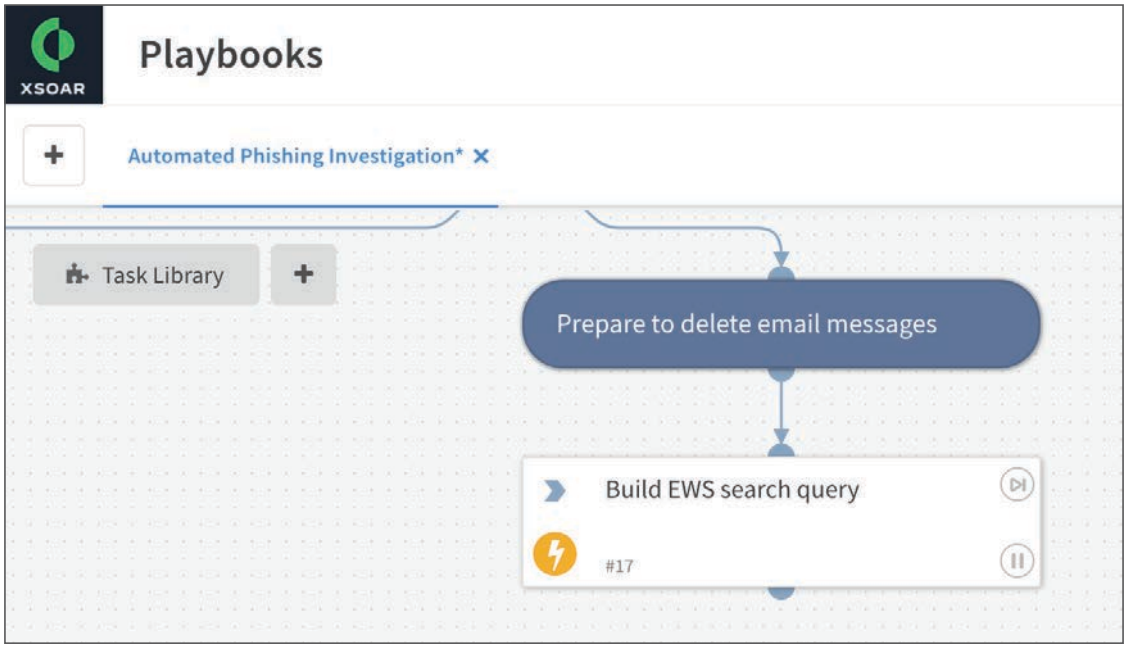
Step 4: In the **search** box, enter **BuildEWSQuery**, and then choose **BuildEWSQuery**. The task fields update.

Step 5: In the **from** box, enter **${EWS.Items.sender}**.

Step 6: In the **subject** box, enter **${EWS.Items.subject}**, and then click **OK**.



Step 7: Verify that the task is now in your playbook.



5.3 Create the "Search EWS System" Task

In this procedure, you use the **ews-search-mailboxes** automation command from the EWS v2 integration. This command uses the *EWS.Query* context data that you formatted in Procedure 5.2.

By default, to protect the EWS system, Cortex XSOAR limits the search to 250 mailboxes. If you need a higher limit on your system, you must override the default setting.

After you have run this automation command, Cortex XSOAR adds *EWS.Items.mailbox* and *EWS.Items.itemid* context data. These values include a list of all mailboxes on the system that contain a copy of the phishing message and the corresponding item ID for each message.

Step 1: From the **Build EWS search query** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

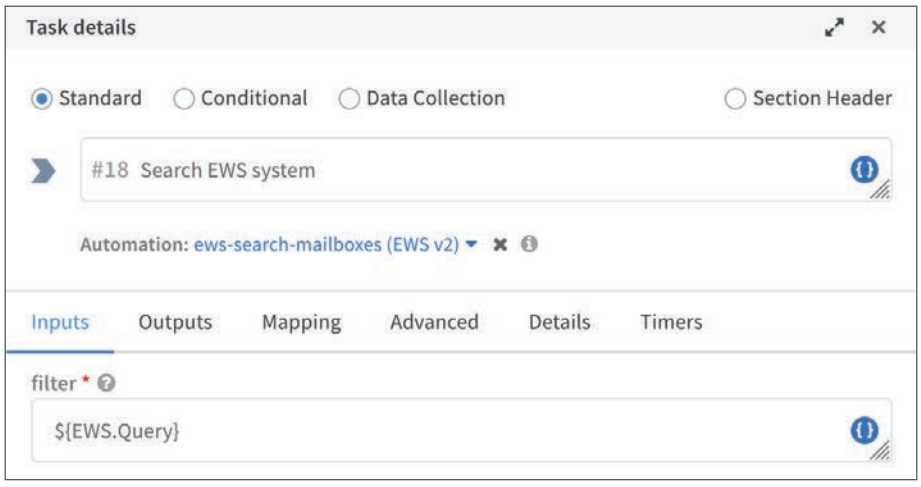
Step 2: In the box with the placeholder **Untitled Task**, enter **Search EWS system**.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 4: In the **search** box, enter **ews-search-mailboxes (EWS v2)**, and then choose **ews-search-mailboxes (EWS v2)**. The task fields update.

Step 5: In the **filter** box, enter **${EWS.Query}**.

Step 6: If you need to search more than 250 mailboxes, in the **limit** box, enter the new limit.



Step 7: On the Advanced tab, in the **Using** list, choose **EWSv2**, and then click **OK**.

Step 8: Verify that the task is now in your playbook.

5.4 Create the "Delete Phishing Emails?" Task

When gathering analyst input, if you need to ask only a single question rather than a multiple question survey, you can use a conditional task that you configure to use the Ask option. This approach is more efficient than combining a data-collection task with a separate conditional task.

To specify the incident owner as the recipient of the single question survey, use the *incident.owner* context data.

Step 1: From the **Search EWS system** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

Step 2: Select **Conditional**.

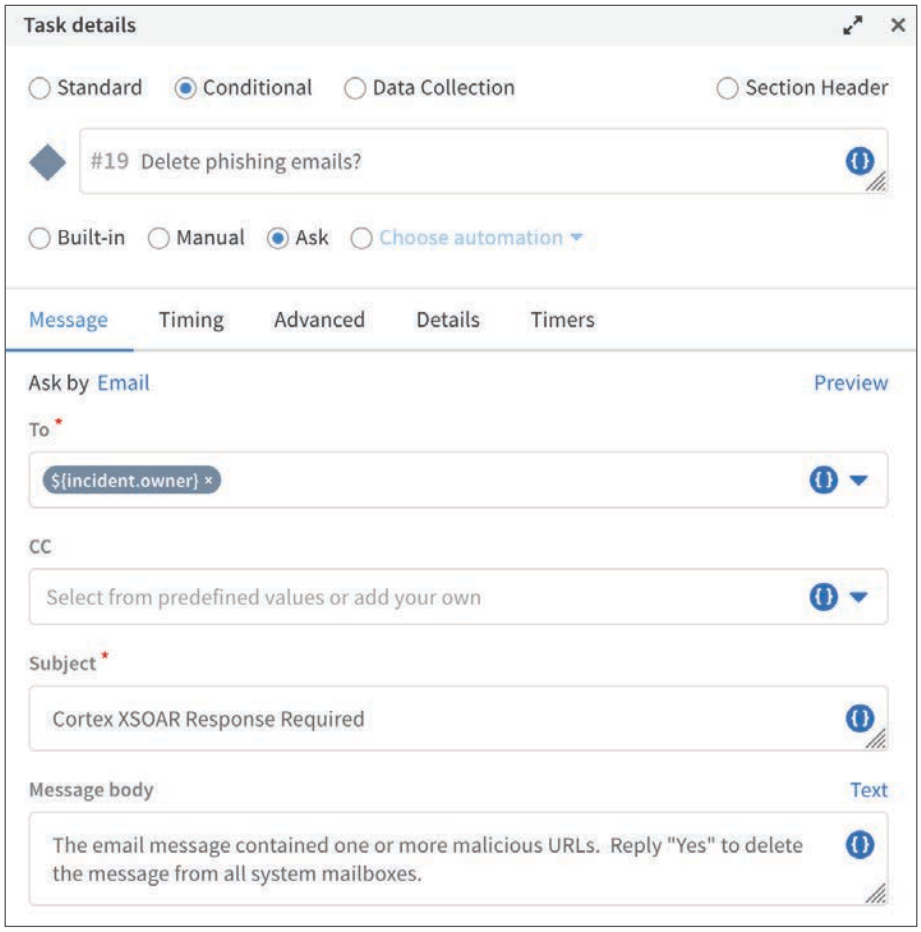
Step 3: In the box with the placeholder **Untitled Task**, enter **Delete phishing emails?**.

Step 4: Select **Ask**.

Step 5: In the **To** box, enter **${incident.owner}**, and then press ENTER.

Step 6: In the **Subject** box, enter **Cortex XSOAR Response Required**.

Step 7: In the **Message body** box, enter **The email message contained one or more malicious URLs. Reply "Yes" to delete the message from all system mailboxes.**, and then click **OK**.



Step 8: Verify that the task is now in your playbook.

5.5 Create the "Delete Phishing Email (Systemwide)" Task

This is the only task in a new branch of the playbook. The playbook selects this branch only if the case owner chooses to delete the phishing email message.

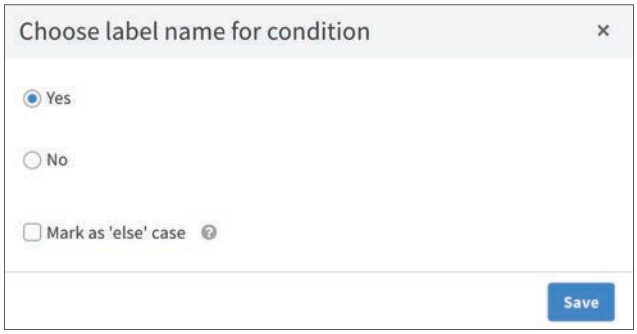
In this procedure, you use the **ews-delete-items** automation command from the EWS v2 integration.

This automation command uses *EWS.Items.itemId* and *EWS.Items.mailbox* context data. This value includes a list of all mailboxes on the system that contain a copy of the phishing message.

> **Caution**
>
> This task requires elevated rights on the EWS mail system and deletes a specific email message from all mailboxes systemwide.
>
> Verify the configuration details are correct before you execute this task within your playbook.

Step 1: From the Delete phishing emails? task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below and to the right.

Step 2: In the Choose label name for condition dialog box, select **Yes**.



Step 3: Click **Save**. The Edit Task dialog box appears.

Step 4: In the box with the placeholder **Untitled Task**, enter Delete phishing emails (systemwide).

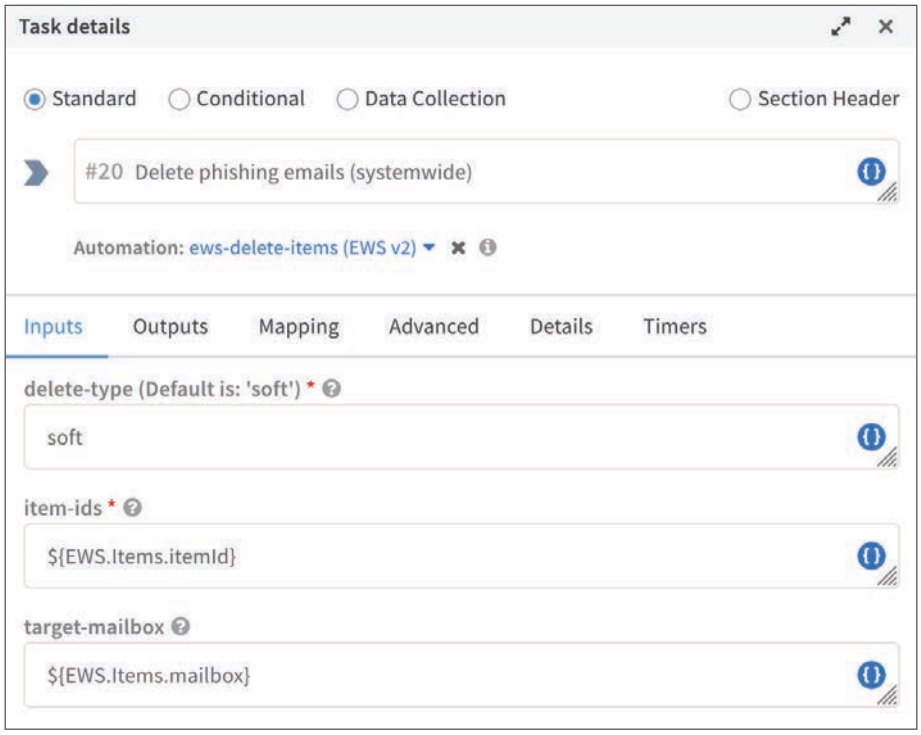
Step 5: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 6: In the **search** box, enter **ews-delete-items (EWS v2)**, and then choose **ews-delete-items (EWS v2)**. The task fields update.

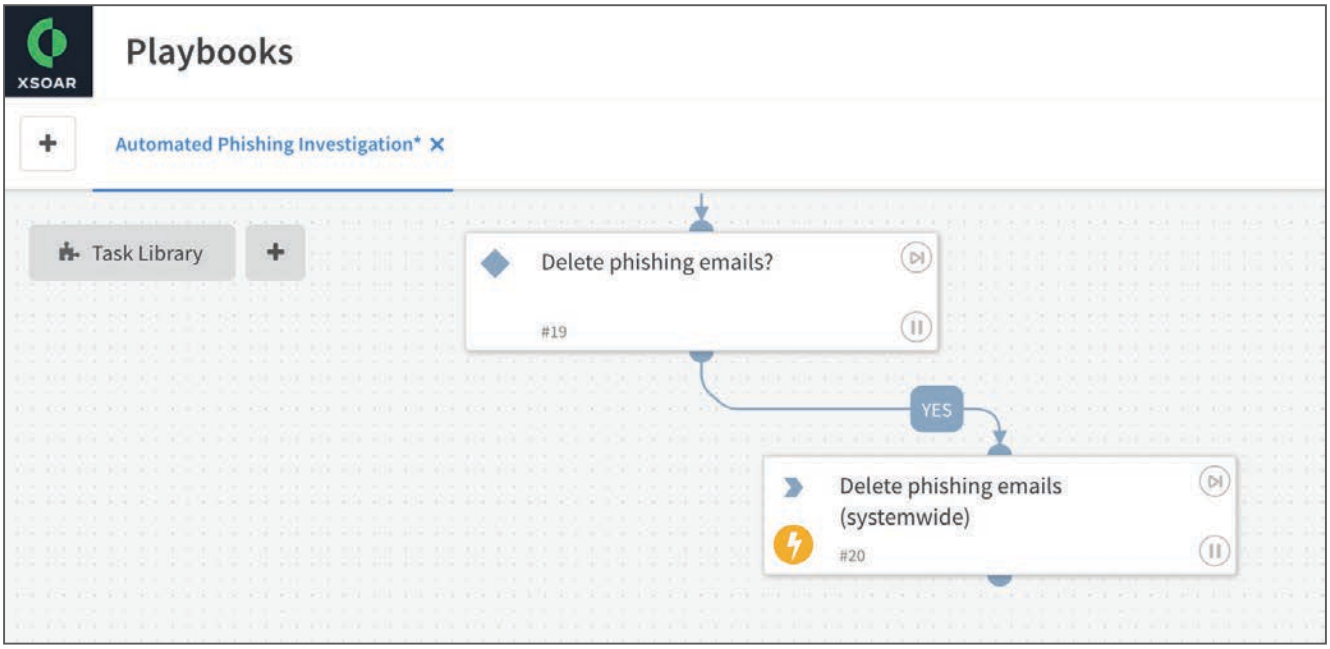
Step 7: In the **delete-type** box, enter **soft**.

Step 8: In the **items-ids** box, enter **${EWS.Items.itemId}**.

Step 9: In the **target-mailbox** box, enter **${EWS.Items.mailbox}**, and then click **OK**.



Step 10: Verify that the task is now in your playbook.



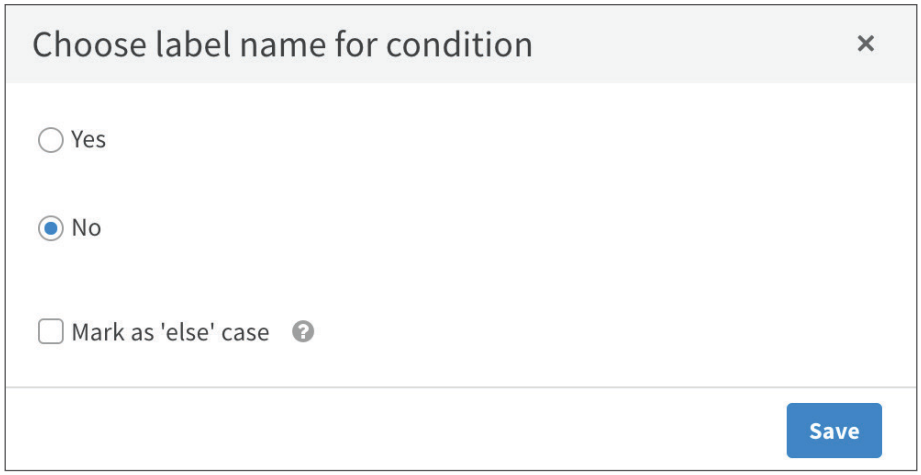
5.6 Connect Playbook Tasks to the Done Task

If you have completed Procedure 4.4, your playbook already contains a Done section header. In this procedure, you connect existing tasks to the Done section header.

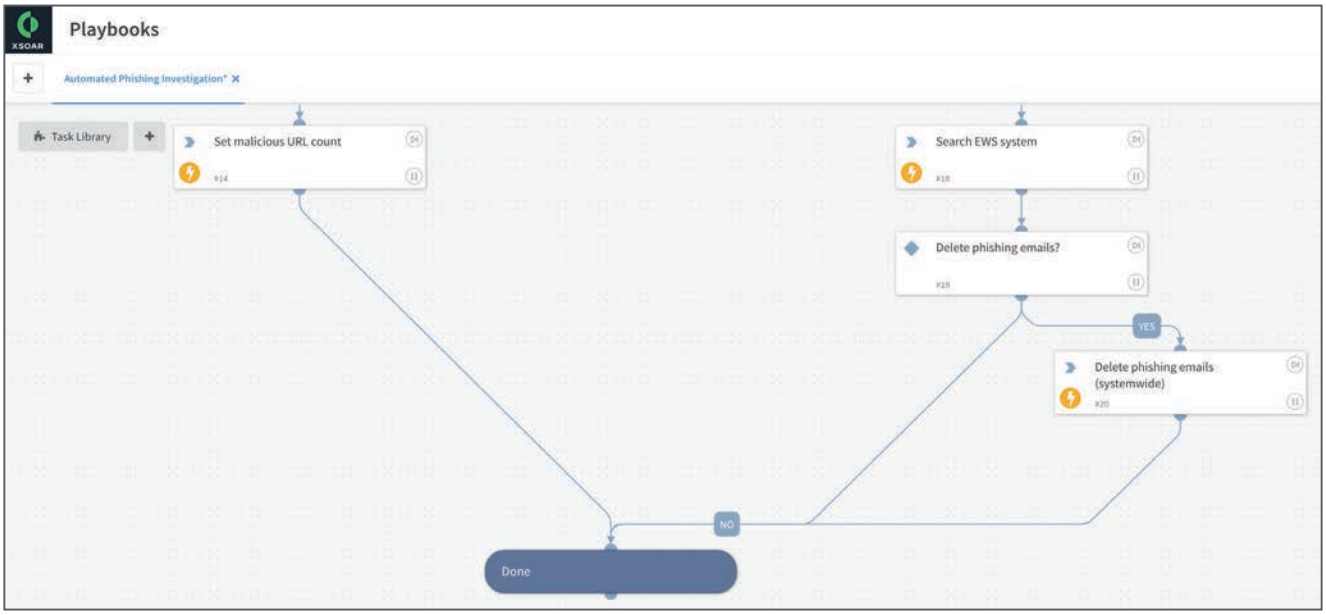
Step 1: From the **Delete phishing emails (systemwide)** task egress node, drag the task connector line to the **Done** task ingress node, and then release to create an additional connection to the **Done** task.

Step 2: From the **Delete phishing emails?** task egress node, drag the task connector line to the **Done** task ingress node, and then release to create an additional connection to the **Done** task.

Step 3: In the Choose Label Name for Condition dialog box, select **No**, and then click **Save**.



Step 4: Verify that the task is now in your playbook.



Step 5: To save the playbook, click **Save Playbook**.

TAKE AUTOMATED ACTION IF USERS ACCESSED URL

The basic playbook performs only an automated analysis. If you want to include an option to analyze security logs and check if any users tried to access any malicious URLs from the phishing message, complete the following modifications.

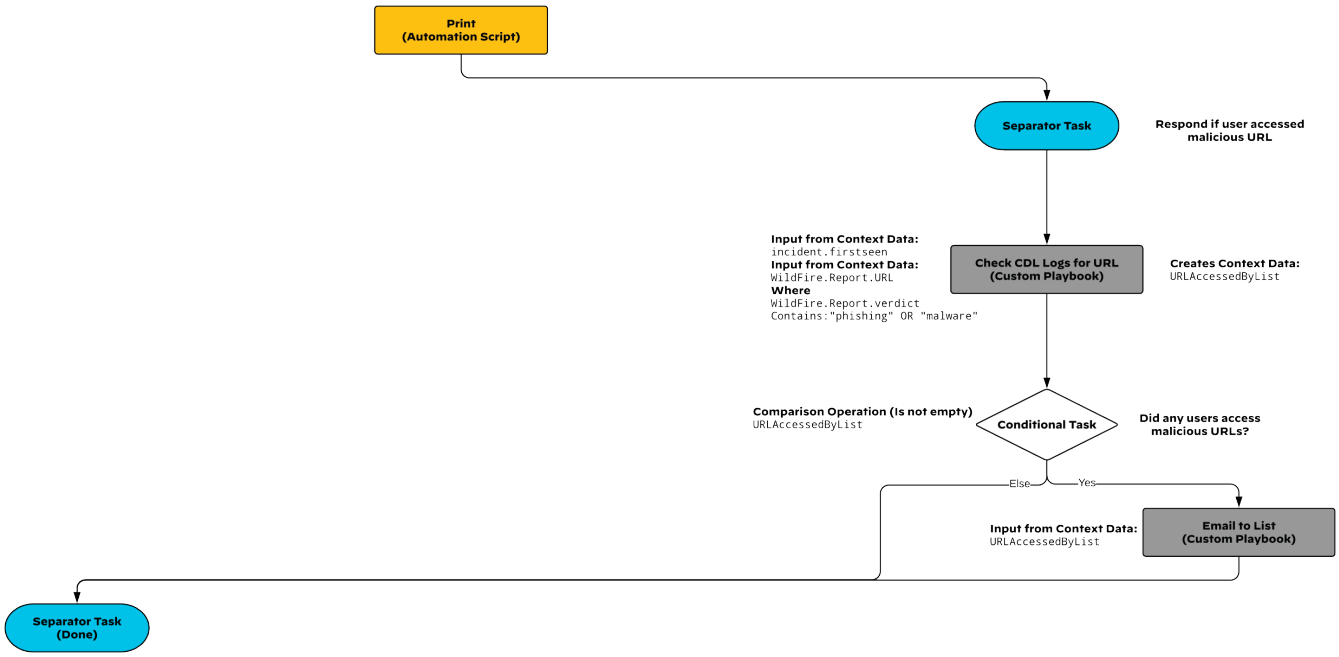
In this section, to limit the size and complexity of the basic playbook, you use sub-playbooks.

First, you create two custom sub-playbooks and complete any required prerequisites for:

- **Email to List**—playbook to send individual emails to all recipients in a list.
- **Check CDL Logs for URL**—playbook to check Cortex Data Lake for URL log entries.

Next, after you create the sub-playbooks, you modify the parent playbook and include these sub-playbooks as automation tasks.

Figure 4 Modifications to Automated Phishing Investigation playbook



Procedures

Creating a Playbook to Send Unicast Emails to a List

6.1 Create the “Email to List” Playbook

6.2 Create the “Send Unicast Email” Task

6.3 Create the “Done” Task

This simple playbook sends individual emails to all recipients in a list. When you add this playbook as a sub-playbook task, configure it to loop for each input.

Use this playbook when you do not want to expose the list of recipients in the To field.

> **Note**
>
> Commonly, when you do not want to expose the list of recipients in the To field, you would instead specify the list of recipients in the BCC field.
>
> The **send-mail** automation command does not permit you to leave the To field blank.

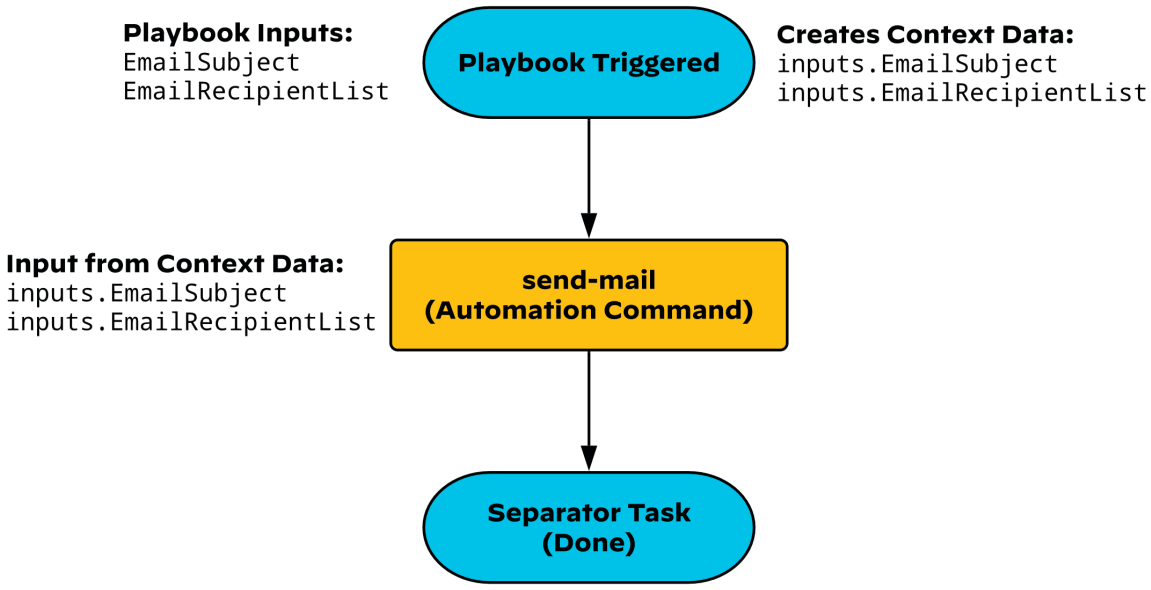
When you complete and save the playbook, Cortex XSOAR adds the playbook to the custom playbook library.

6.1 Create the "Email to List" Playbook

In this procedure, you create a sub-playbook for the Automated Phishing Analysis playbook. The primary reason for using this sub-playbook instead of the automation command on its own is to provide a looping mechanism.

When you create the playbook, you add two inputs, *EmailSubject* and *EmailRecipientList*. Although not shown in this example, if you want to include an email message body, you need to create an additional input for the playbook.

Figure 5 Email to List playbook

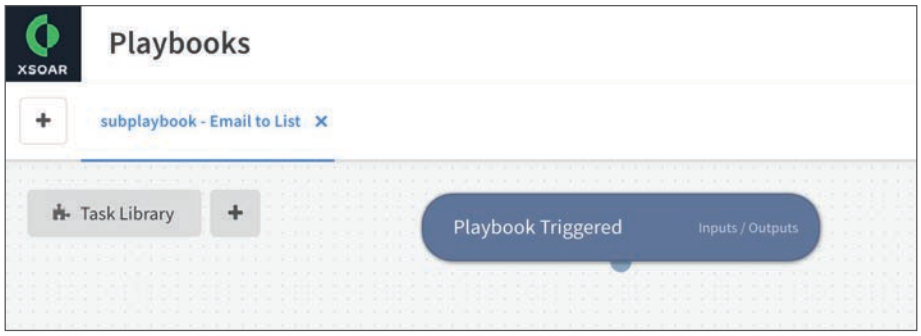


Step 1: In Cortex XSOAR, in the navigation pane, click **Playbooks**.

Step 2: Click **New Playbook**.

Step 3: In the New Playbook dialog box, in the **Playbook name** box, enter **subplaybook - Email to List**, and then click the **Save** button. A playbook workspace with a Playbook Triggered section header appears.

Step 4: If the Task Library dialog box obscures your view of the playbook workspace, click **x** to close the dialog box.



Step 5: Click the Playbook Triggered section header.

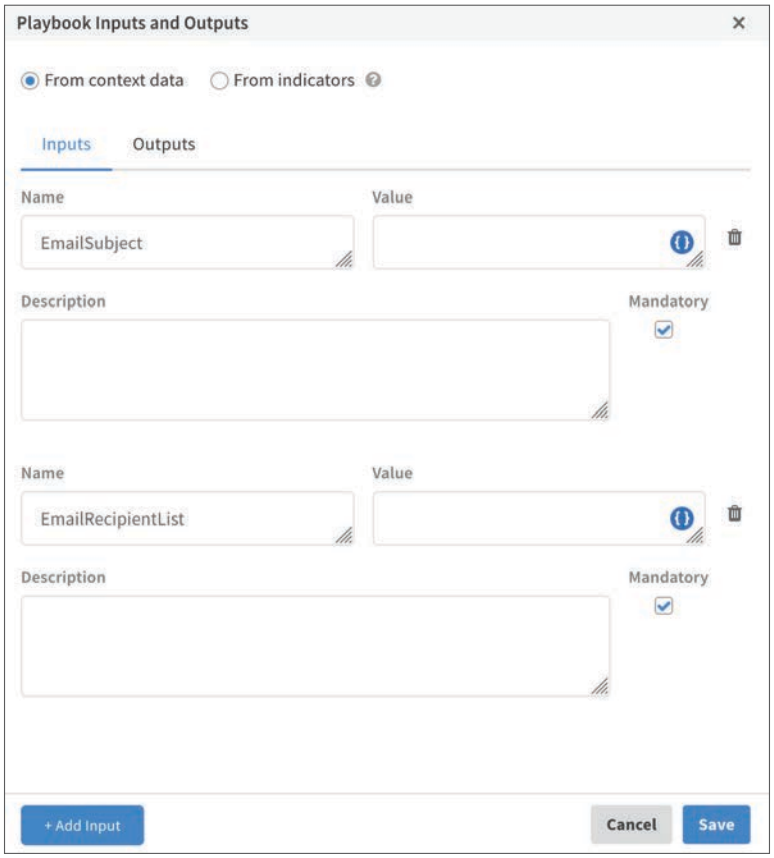
Step 6: In the Playbook Inputs and Outputs dialog box, click **Add Input**.

Step 7: In the Name box, enter **EmailSubject**.

Step 8: Select **Mandatory**, and then click **Add Input**.

Step 9: In the Name box, enter **EmailRecipientList**.

Step 10: Select **Mandatory**, and then click **Save**.



6.2 Create the "Send Unicast Email" Task

The task uses the **send-email** automation command from the EWS Mail Sender integration to send an email message to each recipient in the list you provide to the sub-playbook.

If you want to include a custom message body, you can modify this sub-playbook to use additional inputs from the parent playbook. This task only describes how to include a common message body for all email messages.

Step 1: From the Playbook Triggered section header egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

Step 2: In the box with the placeholder **Untitled Task**, enter **Send unicast email**.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 4: In the **search** box, enter **send-mail (EWS Mail Sender)**, and then choose **send-mail (EWS Mail Sender)**. The task fields update.

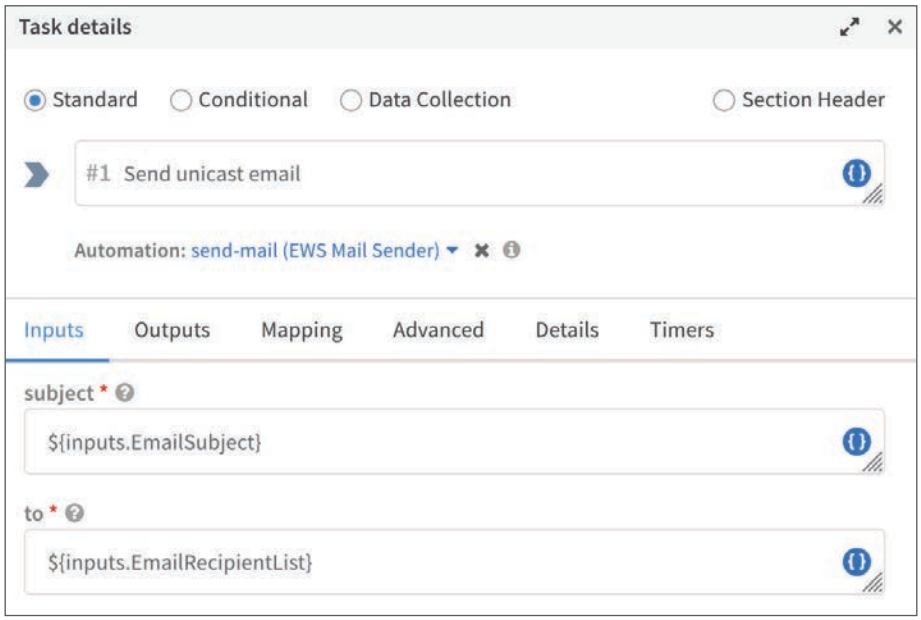
Step 5: In the **subject** box, click the {} button. The Select Source for Subject dialog box appears.

Step 6: In the search box, enter **EmailSubject**. In the Playbook Inputs section, click **EmailSubject**, and then click **Close**.

Step 7: In the **to** box, click the {} button. The Select Source for To dialog box appears.

Step 8: In the search box, enter **EmailRecipientList**. In the Playbook Inputs section, click **EmailRecipientList**, and then click **Close**.

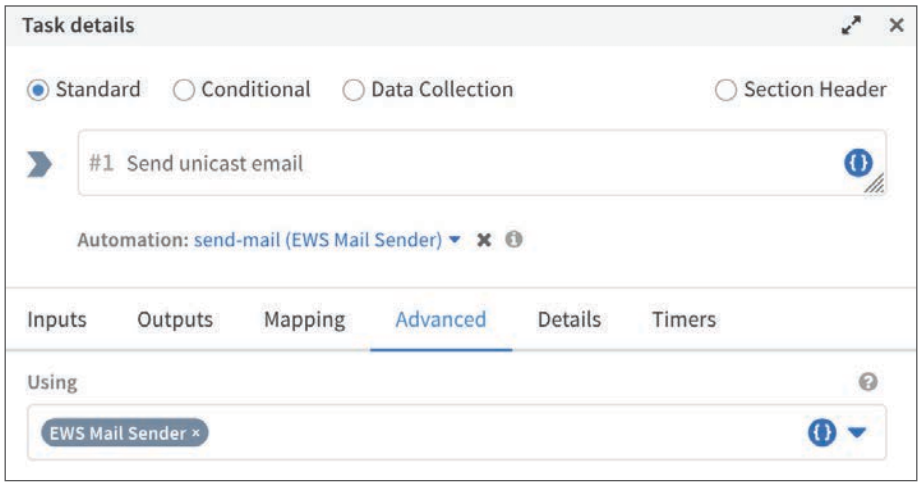
Step 9: Verify that the **subject** box and the **to** box are now correctly populated.



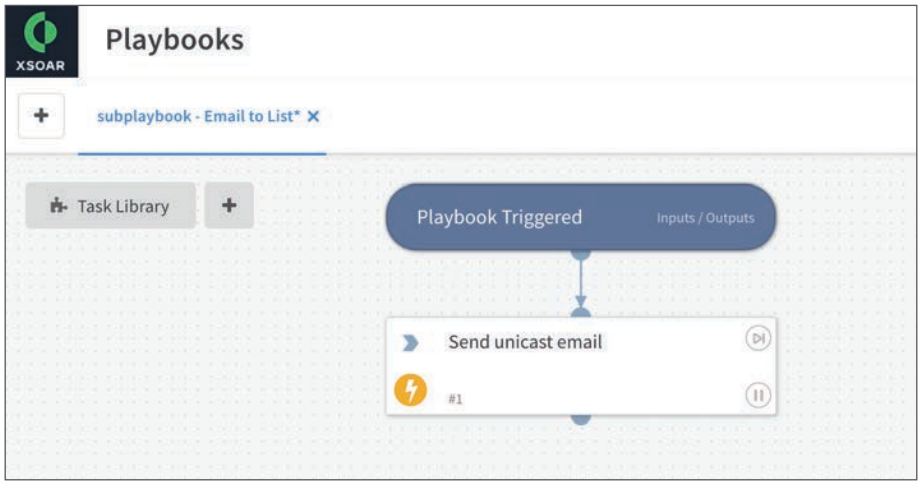
You can provide the message body in text or HTML formats. This example shows how to enter the body by using basic text format. If you want to use HTML format instead, enter the HTML source in the **htmlBody** box.

Step 10: In the **body** box, enter "Cortex XSOAR sent you this message because you accessed a malicious URL.".

Step 11: On the Advanced tab, in the **Using** list, choose EWS Mail Sender, and then click **OK**.



Step 12: Verify that the task is now in your playbook.



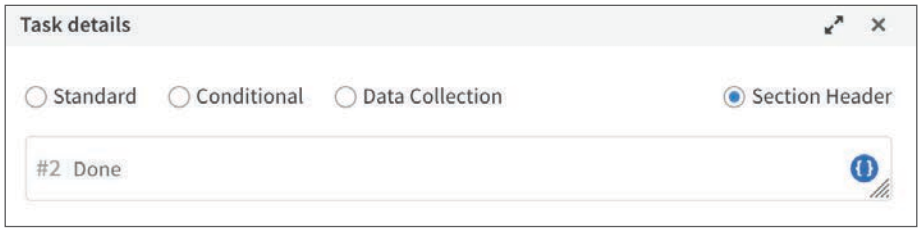
6.3 Create the "Done" Task

As a best practice, you should create a Done section header that terminates the playbook.

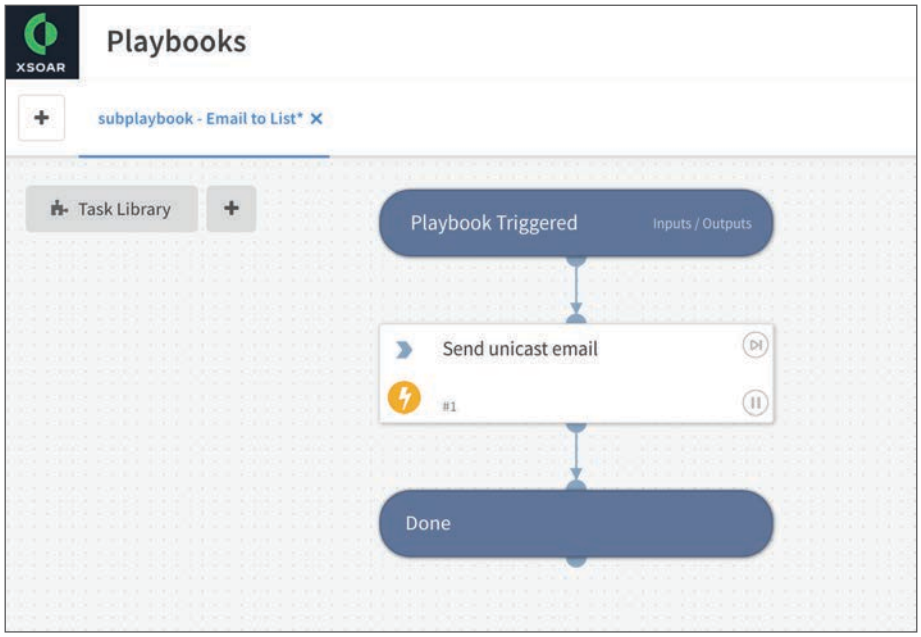
Step 1: From the Send unicast email task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled blank task below. The Edit Task dialog box appears.

Step 2: Select **Section Header**.

Step 3: In the box with the placeholder **Untitled Task**, enter **Done**, and then click **OK**.



Step 4: Verify that the task is now in your playbook.



Step 5: To save the playbook, click **Save Playbook**.

Procedures

Creating a Playbook to Check Cortex Data Lake URL Logs

This playbook checks Cortex Data Lake for URL log entries that match a list of URLs.

When you complete and save the playbook, Cortex XSOAR adds the playbook to the custom playbook library.

You first configure the Cortex Data Lake integration instance, then you create a playbook that uses an automation command from that integration to query Cortex Data Lake.

7.1 Configure Cortex Data Lake Integration Instance

This procedure assumes that you have already installed the Cortex Data Lake content pack from the Cortex XSOAR Marketplace in Procedure 1.1.

This procedure requires that your organization has an active Cortex Data Lake subscription. As an active subscriber, you are entitled to access the Cortex Data Lake API.

You first need to retrieve the License ID and Customer Name information from Cortex XSOAR. Next, you submit this information to the Palo Alto Networks customer support portal in order to generate the authentication credentials and encryption key for that the Cortex XSOAR integration instance uses to interact with Cortex Data Lake.

Step 1: Log in to the Cortex XSOAR portal (example: https://xsoar.example.com).

Step 2: In the navigation pane, click **Playground**.

Step 3: In the Cortex XSOAR CLI, enter **!GetLicenseID**, and record the License ID value.

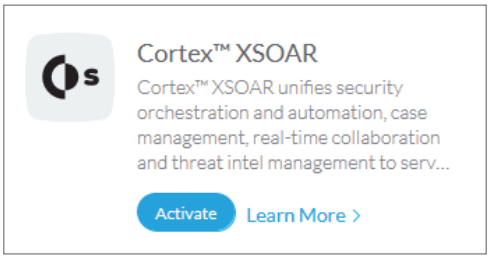
Step 4: In the navigation pane, click **Settings**.

Step 5: In **About > License**, record the Customer Name value.

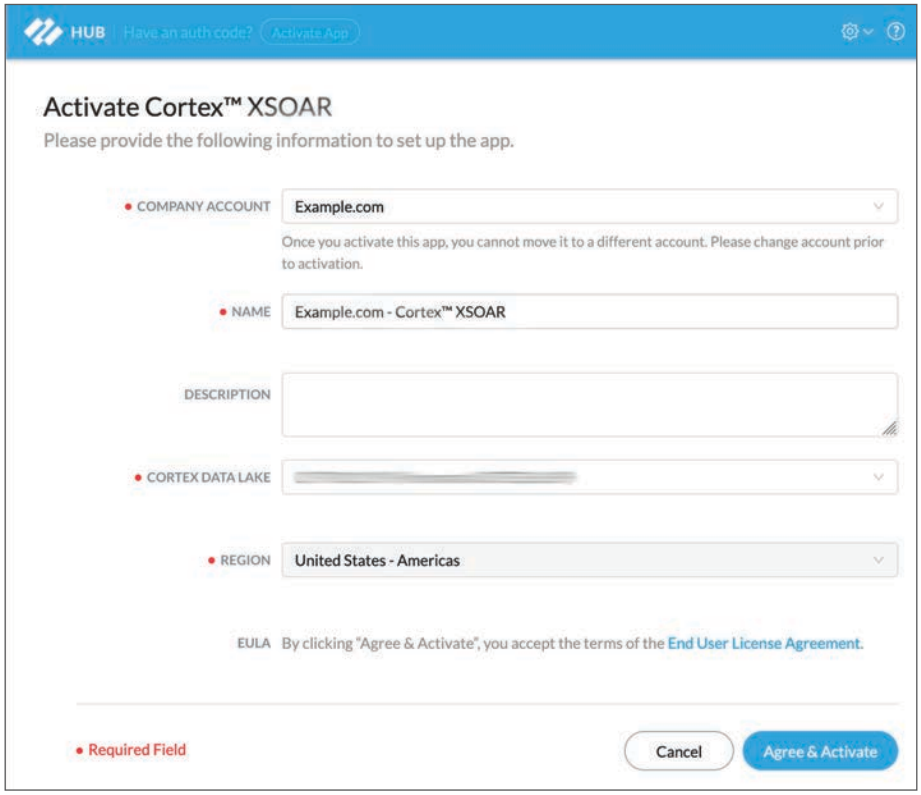
Step 6: Login to the Palo Alto Networks hub at **https://apps.paloaltonetworks.com/apps**.

Step 7: If the Cortex XSOAR application is already in the list of active applications, skip to Step 10.

Step 8: In the Cortex XSOAR section, click **Activate**.



Step 9: In the Activate Cortex XSOAR dialog box, verify that the selected details are correct, and then click **Agree & Activate**.



Step 10: Click **Cortex XSOAR**.

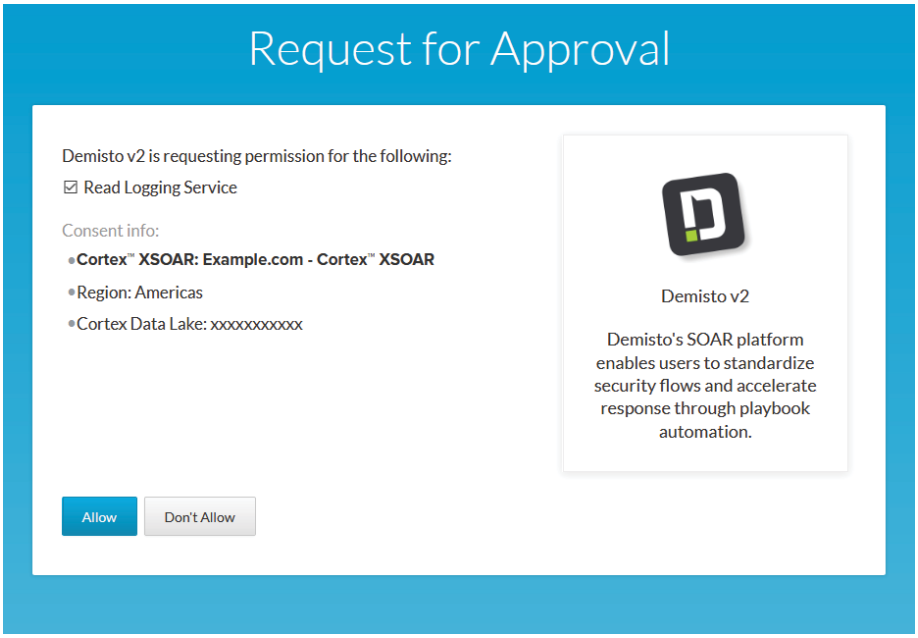
Step 11: Using the data recorded from Step 3, in the **Demisto License ID box**, enter the license ID.

Step 12: Using the data recorded from Step 5, in the **Demisto Customer Name** box, enter the customer name.

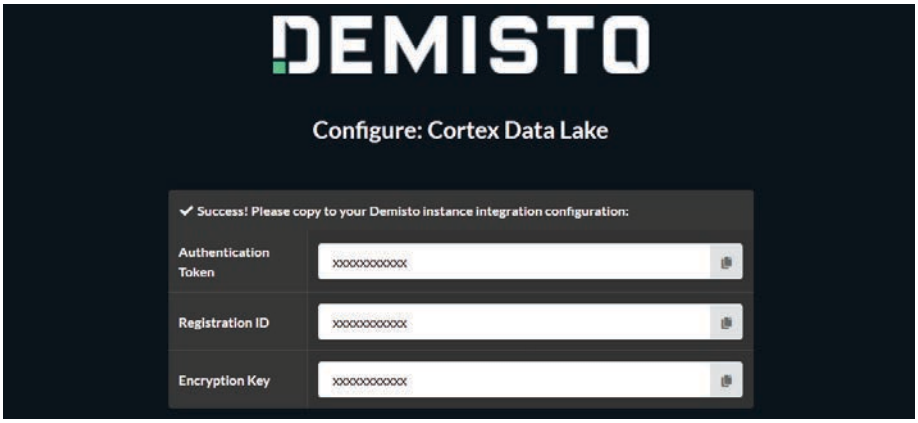
> **Note**
>
> If you have already authorized Cortex XSOAR to access Cortex Data Lake and you rerun the authorization process, you invalidate the previously generated tokens and keys.

Step 13: Click **Start Authorization Process**.

Step 14: On the Request for Approval page, click **Allow**.



Step 15: From the Configure: Cortex Data Lake page, record the Authentication Token, Registration ID, and Encryption Key.

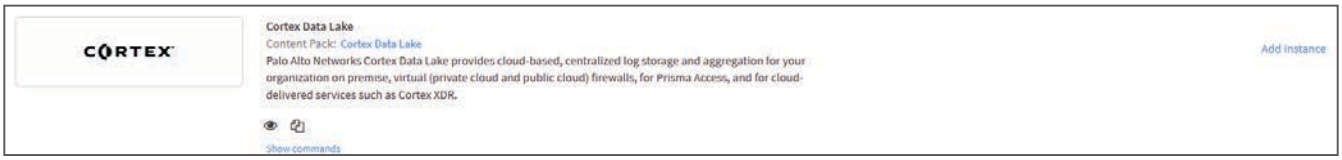


Step 16: Log in to the Cortex XSOAR portal (example: https://xsoar.example.com).

Step 17: In the navigation pane, click **Settings**.

Step 18: In **Integrations > Servers & Services**, in the search box, enter **Cortex Data Lake**.

Step 19: Click **Add instance**.

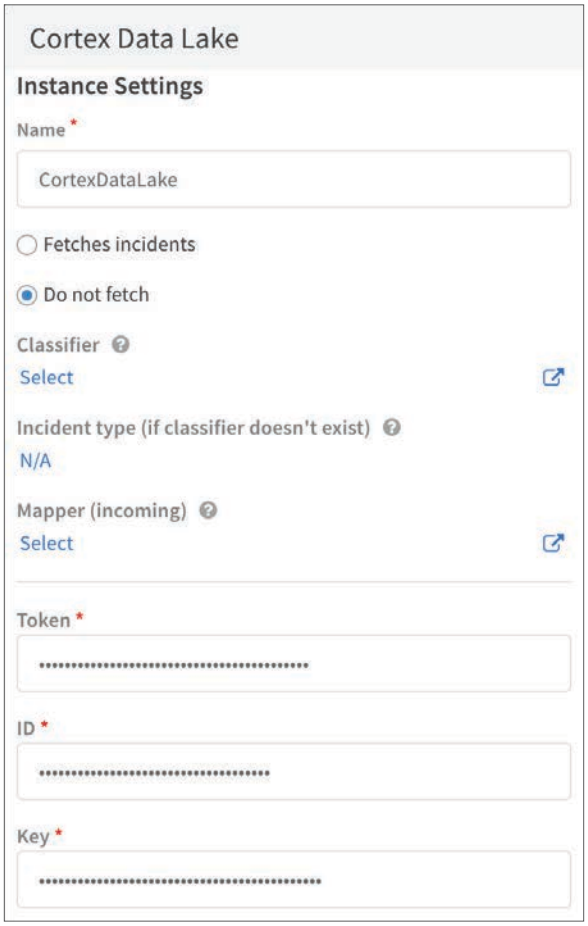


Step 20: In the **Name** box, enter **CortexDataLake**.

Step 21: Using the data recorded from Step 15, in the **Token** box, enter the authentication token.

Step 22: Using the data recorded from Step 15, in the **ID** box, enter the registration ID.

Step 23: Using the data recorded from Step 15, in the **Key** box, enter the encryption key, and then click **Save & exit**.

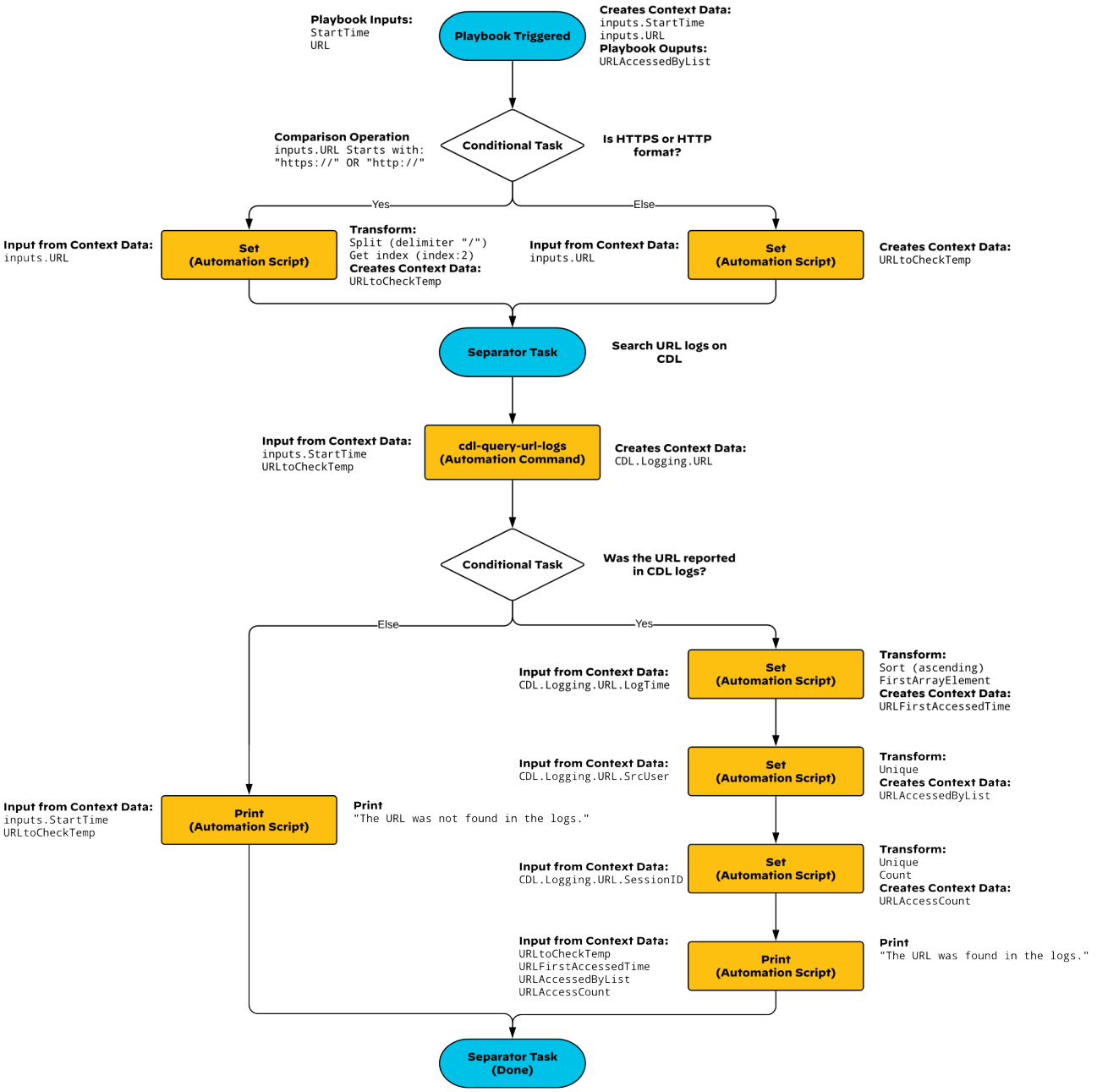


7.2 Create the "Check CDL Logs for URL" Playbook

In this procedure, you create a sub-playbook for the **Automated Phishing Analysis** playbook. The primary reason for using this sub-playbook is to simplify the parent playbook and to provide a looping mechanism. When you create the playbook, you add two inputs, *StartTime* and *URL*. The **cdl-query-url-logs** automation command requires that you specify the start time for each query. The sub-playbook filters and transforms the input URL, so that you can provide the input using any of the following formats:

- https://www.example.com
- http://www.example.com/path
- www.example.com

Figure 6 Check CDL Logs for URL playbook

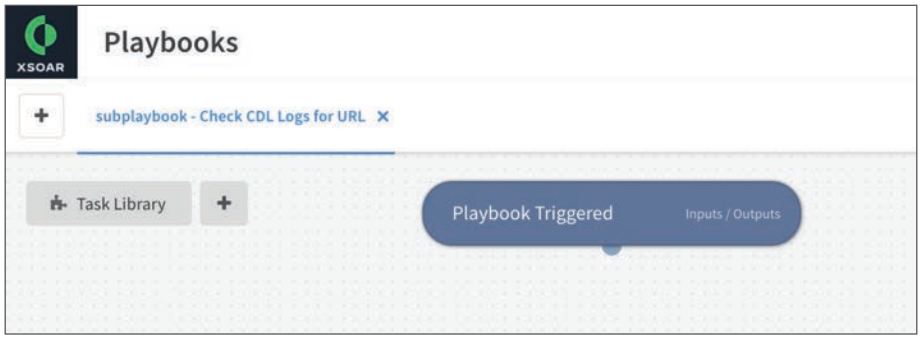


Step 1: In Cortex XSOAR, in the navigation pane, click **Playbooks**.

Step 2: Click **New Playbook**.

Step 3: In the New Playbook dialog box, in the **Playbook name** box, enter **subplaybook - Check CDL Logs for URL**, and then click the **Save** button. A playbook workspace with a Playbook Triggered section header appears.

Step 4: If the Task Library dialog box obscures your view of the playbook workspace, click the **x** to close the dialog box.



Step 5: Click the Playbook Triggered section header.

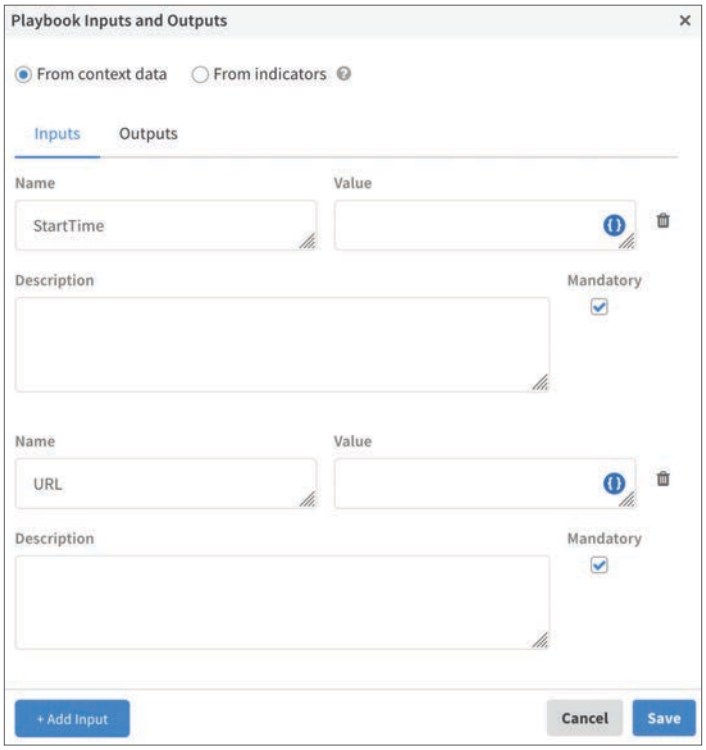
Step 6: In the Playbook Inputs and Outputs dialog box, click **Add Input**.

Step 7: In the Name box, enter **StartTime**.

Step 8: Select **Mandatory**, and then click **Add Input**.

Step 9: In the Name box, enter **URL**.

Step 10: Select **Mandatory**, and then click **Save**.



7.3 Create the "Is HTTPS or HTTP format?" Task

In this procedure, you perform a check to determine the format of the playbook input. If the URL that you provide as an input includes "https://" or "http://" or includes path after the domain name, you reformat the URL by removing these components. Otherwise, you do not need to modify the format. Your playbook executes different branches depending on the results of the check.

Step 1: From the Playbook Triggered section header egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

Step 2: Select **Conditional**.

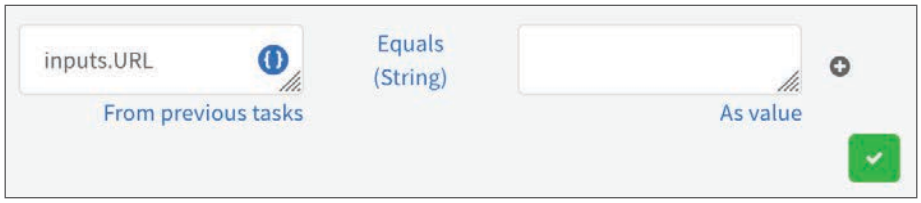
Step 3: In the box with the placeholder **Untitled Task**, enter **Is HTTPS or HTTP format?**.

Step 4: In the conditional statement section left-side box, click the {} button. The Select Source For dialog box appears.

Step 5: In the search box, enter **URL**. In the Playbook Inputs section, click **URL**, and then click **Close**.

Next, you choose the comparison operator for the conditional statement.

Step 6: Click **Equals**.



Step 7: In the search box, enter **Starts with**, and then click **Starts with (String)**.

Step 8: In the right-side box, enter **https://**.

Step 9: Click + to add the second conditional statement.

Step 10: In the left-side box, enter **inputs.URL**.

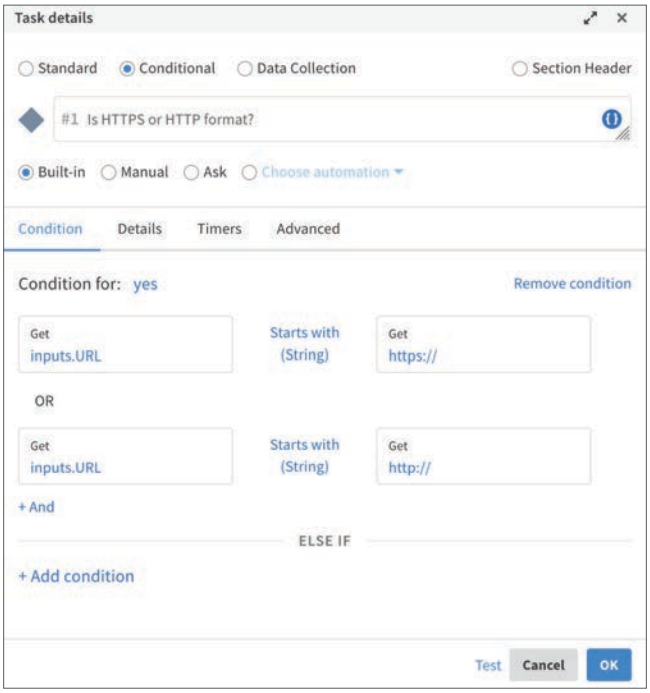
Next, you choose the comparison operator for the conditional statement.

Step 11: Click **Equals**.

Step 12: In the search box, enter **Starts with**, and then click **Starts with (String)**.

Step 13: In the right-side box, enter **http://**, and then click the check.

Step 14: Verify the task configuration, and then click **OK**.



Step 15: Verify that the task is now in your playbook.

7.4 Create the "Strip and Set Temporary URL" Task

Before you can set the value for the incident field Malicious URL Count, you must filter and transform the WildFire report data from Procedure 2.9 and store this value using the temporary variable *URLtoCheckTemp*.

This task uses the **Set** automation script.

Step 1: From the Is HTTPS or HTTP format? section header egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below and to the left.

Step 2: In the Choose Label Name for Condition dialog box, select **yes**.

Step 3: Click **Save**. The Edit Task dialog box appears.

Step 4: In the box with the placeholder **Untitled Task**, enter **Strip and set temporary URL**.

Step 5: In the Choose Automation section, click the down arrow to open the search dialog box.

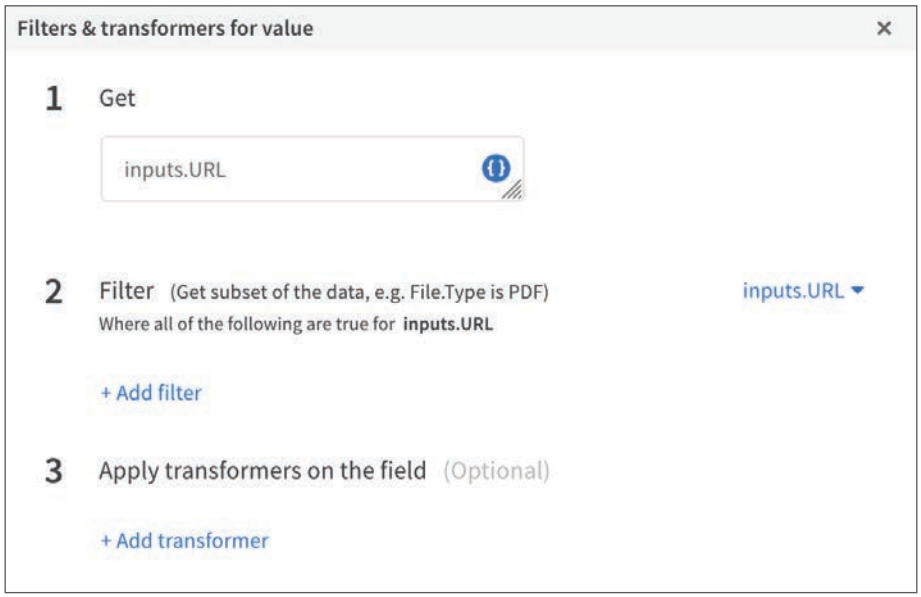
Step 6: In the **search** box, enter **Set**, and then choose **Set**. The task fields update.

Step 7: In the **key** box, enter **URLtoCheckTemp**.

Step 8: In the value box, click the {} button. The Select Source for Value dialog box appears.

Step 9: In the search box, enter **URL**.

Step 10: In the Playbook Inputs section, in the row for **URL**, click **Filter & transform**. The dialog box name changes to Filters & Transformers for Value.



Now you transform the URL. First, you split the URL into separate sections divided by the "/" character by using the *split* transformer.

Step 11: In the Apply Transformers on the Field section, click **Add transformer**.

Step 12: Click **To upper case**.

Step 13: In the search box, enter **Split**, and then click **Split**.

Step 14: In the **delimiter** box, enter /.

Step 15: Click the check.

Next, you choose the section after the second /, using the *Get index* transformer.

Step 16: In the Apply Transformers on the Field section, click **Add transformer**.

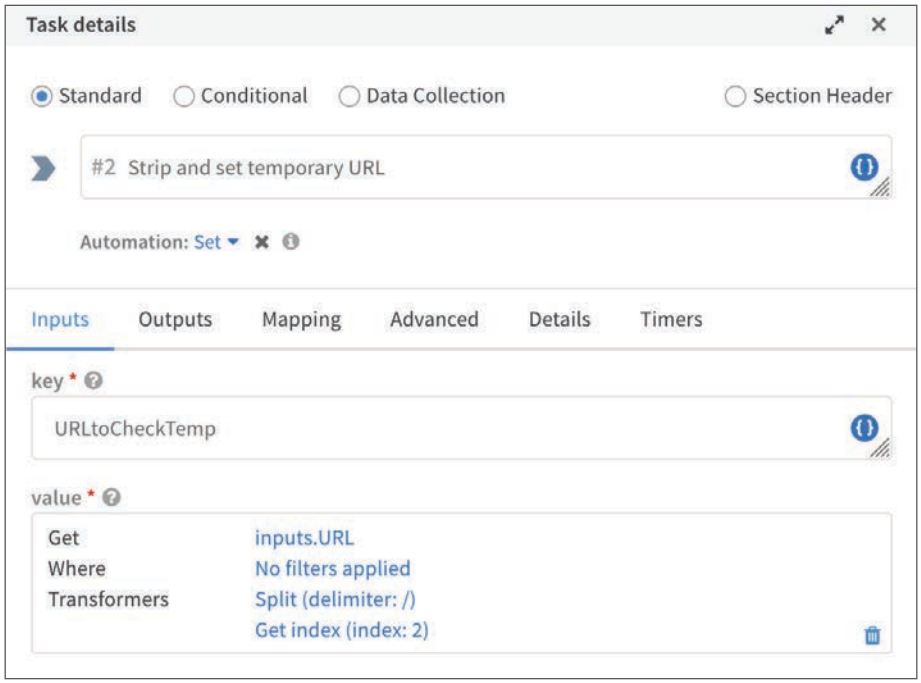
Step 17: Click **To upper case**.

Step 18: In the search box, enter **Get index** and then click **Get index**.

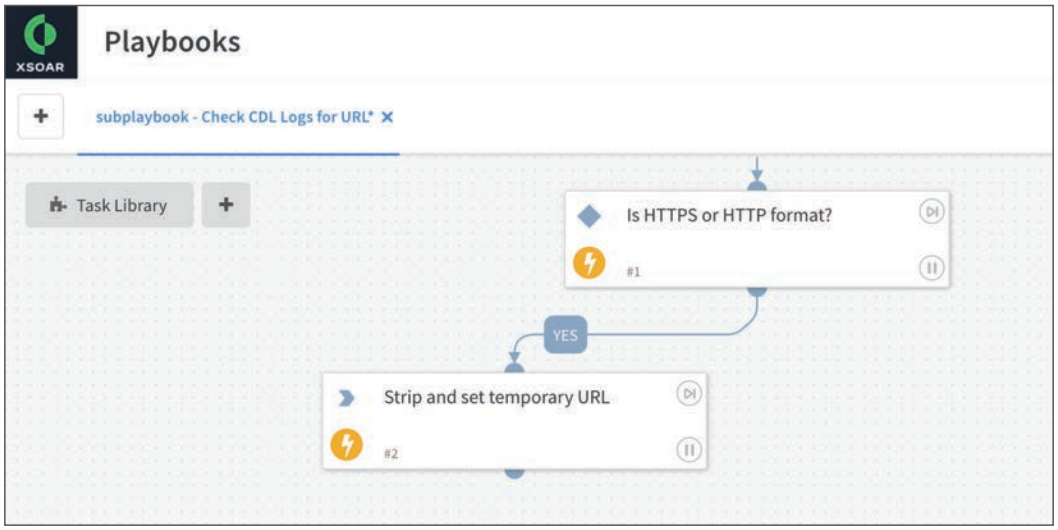
Step 19: In the **index** box, enter 2.

Step 20: Click the check, and then click **OK**.

Step 21: Verify the task configuration, and then click **OK**.



Step 22: Verify that the task is now in your playbook.



7.5 Create the "Set Temporary URL" Task

If the URL is already in the correct format, you set the value without transforming it. You use the temporary variable *URLtoCheckTemp*.

This task uses the **Set** automation script.

Step 1: From the Is HTTPS or HTTP format? section header egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below and to the right.

Step 2: On the Choose Label Name for Condition dialog box, select **Mark as 'else' case**.

Step 3: Click **Save**. The Edit Task dialog box appears.

Step 4: In the box with the placeholder **Untitled Task**, enter **Set temporary URL**.

Step 5: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 6: In the **search** box, enter **Set**, and then choose **Set**. The task fields update.

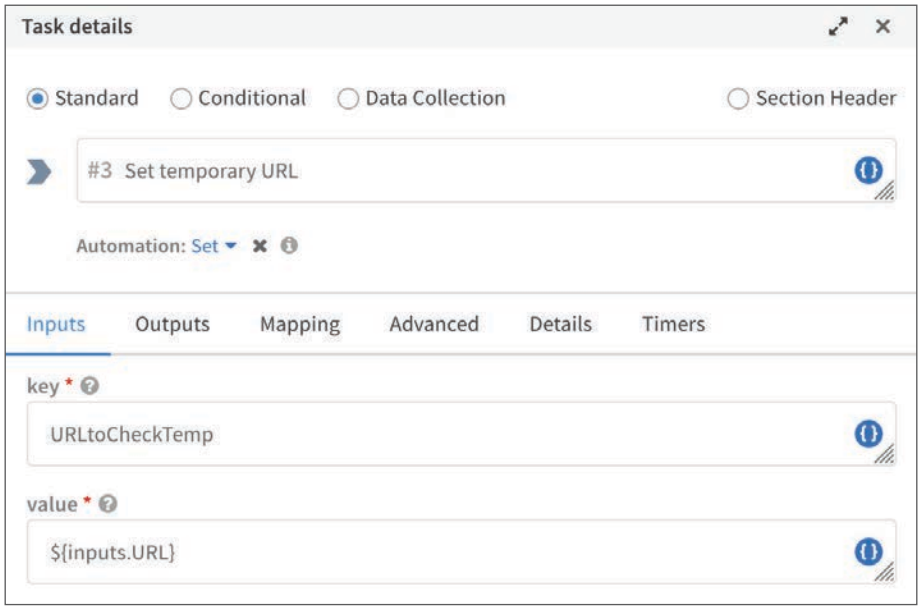
Step 7: In the **key** box, enter **URLtoCheckTemp**.

Step 8: In the value box, click the {} button. The Select Source for Value dialog box appears.

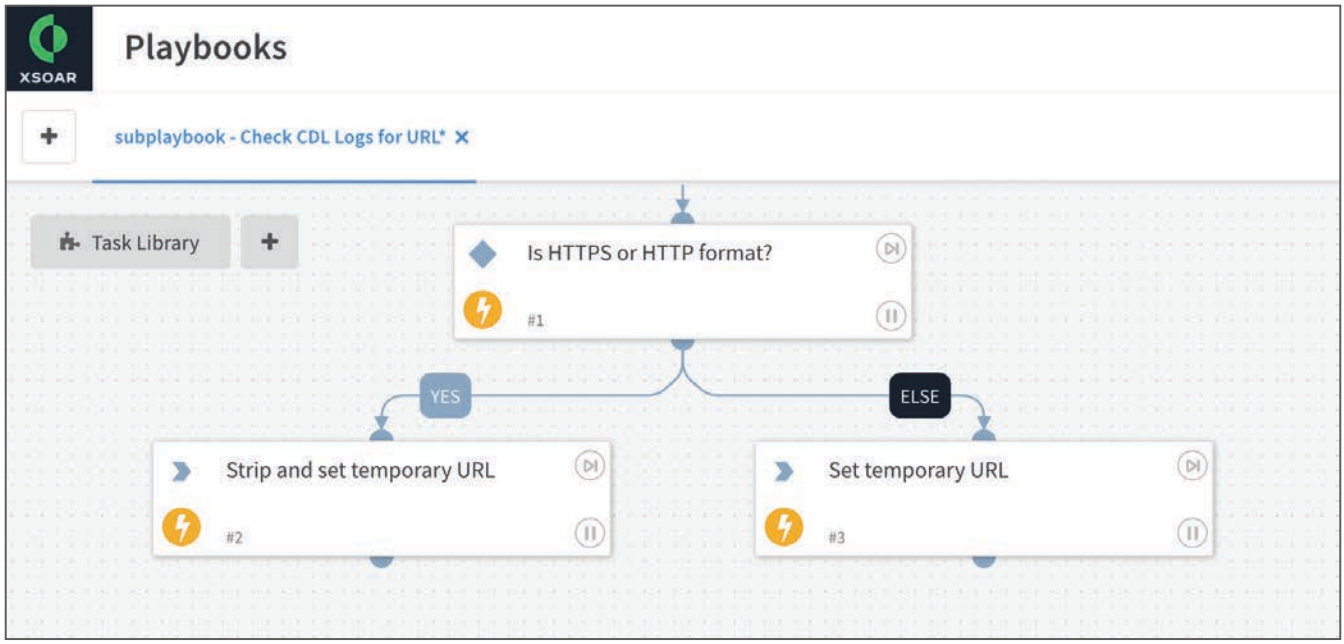
Step 9: In the search box, enter **URL**.

Step 10: In the Playbook Inputs section, click **URL**, and then click **Close**.

Step 11: Verify the task configuration and then click **OK**.



Step 12: Verify that the task is now in your playbook.



7.6 Create a Separator Task

As a best practice, you should begin this section of the playbook by creating a Search URL Logs on CDL section header.

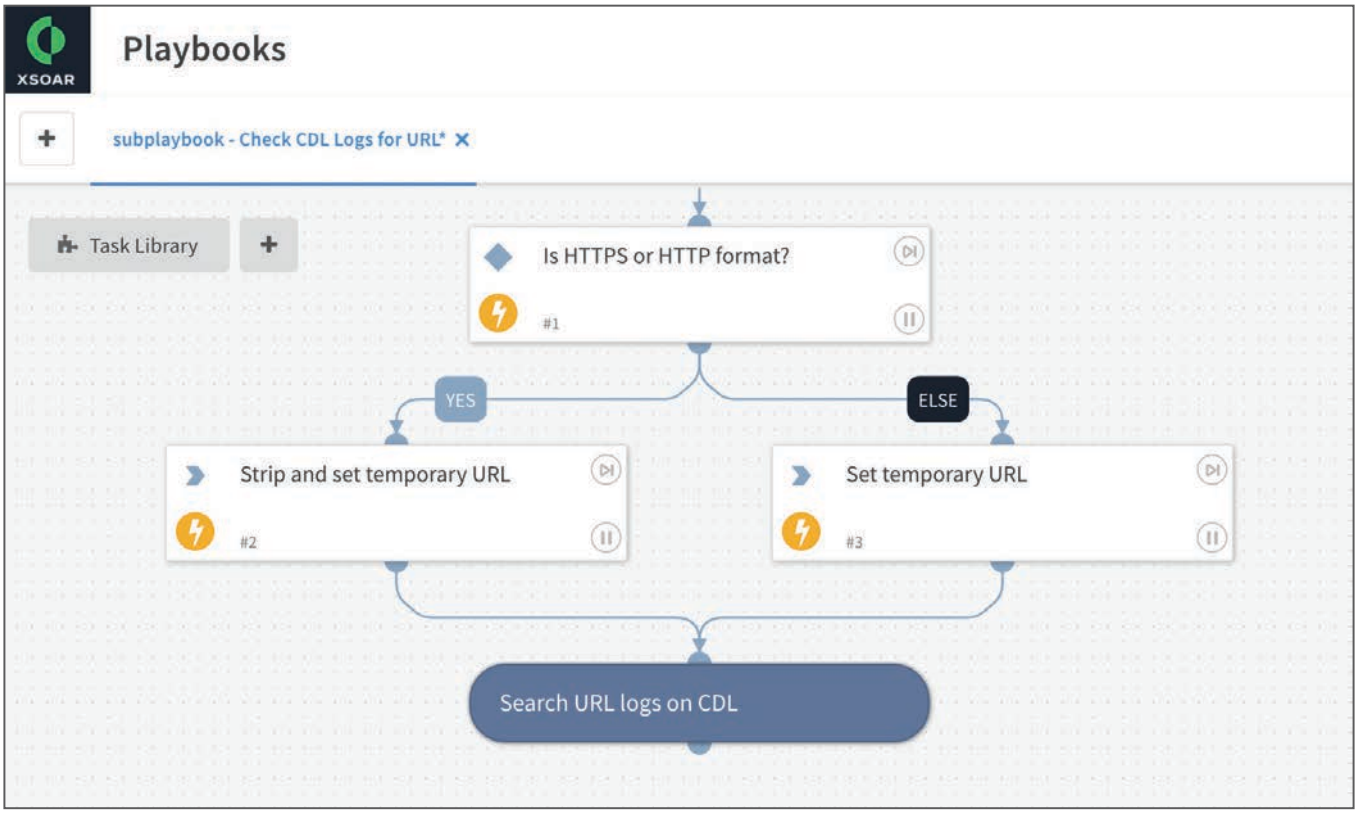
Step 1: From the **Strip and set temporary URL** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below and to the center. The Edit Task dialog box appears.

Step 2: Select **Section Header**.

Step 3: In the box with the placeholder **Untitled Task**, enter **Search URL logs on CDL**, and then click **OK**.

Step 4: From the **Set temporary URL** task egress node, drag the task connector line to the **Search URL Logs on CDL** task ingress node, and then release.

Step 5: Verify that the task is now in your playbook.



7.7 Create the "CDL - Search URL Logs" Task

In this procedure, you use the **cdl-query-url-logs** automation command from the Cortex Data Lake integration. This command uses the *StartTime* playbook input that you defined in Procedure 7.2 and the temporary variable *URLtoCheckTemp* that you set in either Procedure 7.4 or Procedure 7.5.

After you have run this automation command, Cortex XSOAR adds *CDL.Logging* context data. These values include the log entries from Cortex Data Lake that match your query.

Step 1: From the **Search URL logs on CDL** section header egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

Step 2: In the box with the placeholder **Untitled Task**, enter **CDL - search URL logs**.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 4: In the **search** box, enter **cdl-query-url-logs (Cortex Data Lake)**, and then choose **cdl-query-url-logs (Cortex Data Lake)**. The task fields update.

Step 5: In the **Add input** section, click the down arrow. The search dialog box opens.

Step 6: In the **search** box, enter **start_time**, and then choose **start_time**.

Step 7: In the **start_time** box, click the {} button. The Select Source for Start_time dialog box appears.

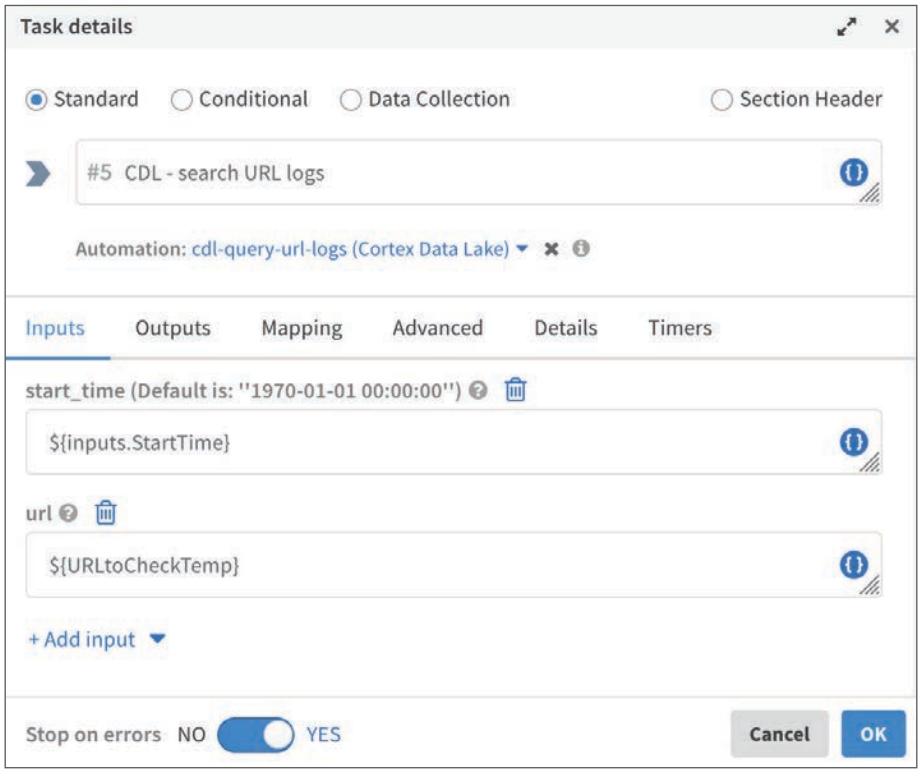
Step 8: In the search box, enter **StartTime**. In the Playbook Inputs section, click **StartTime**, and then click **Close**.

Step 9: In the **Add input** section, click the down arrow. The search dialog box opens.

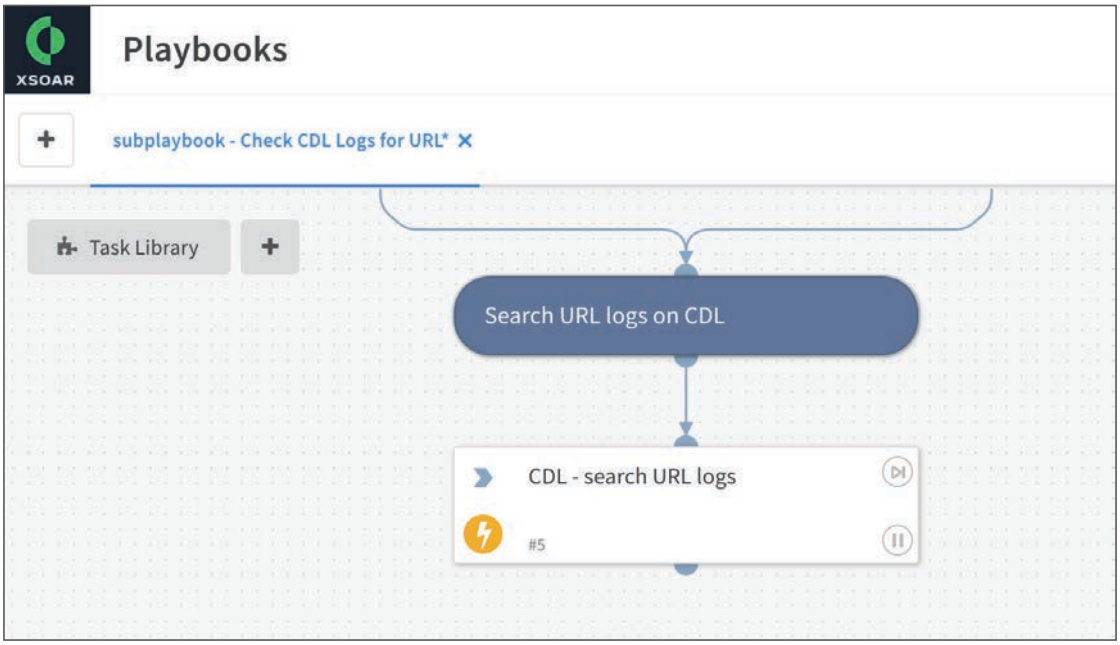
Step 10: In the **search** box, enter **url**, and then choose **url**.

Step 11: In the **url** box, enter **${URLtoCheckTemp}**.

Step 12: Verify that the **start_time** box and the **url** box are now correctly populated, and then click **OK**.



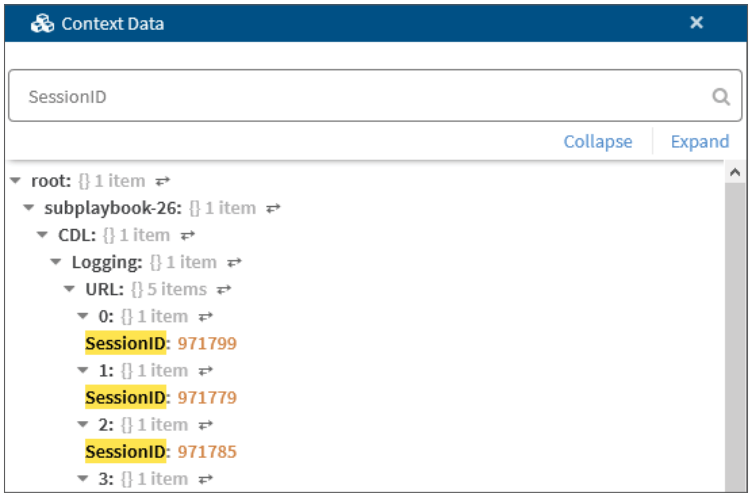
Step 13: Verify that the task is now in your playbook.



7.8 Create the "Was the URL Reported in CDL Logs?" Task

In this procedure, you perform a check to see if the **cdl-query-url-logs** automation command reported any matching log entries. Your playbook executes different branches depending on the results of the check.

To perform the check, your conditional statement uses the *CDL.Logging.URL.SessionID* context data. If this object has a defined value, then the query returned at least one matching log entry for the URL. Otherwise, the query did not return any matching log entries for the URL.



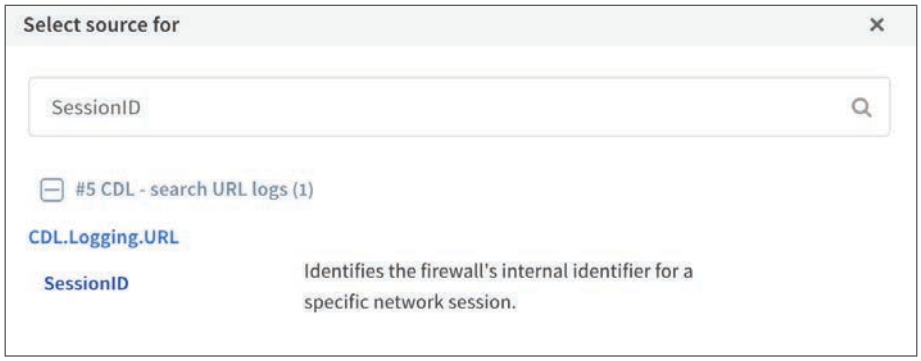
Step 1: From the CDL - Search URL logs task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

Step 2: Select **Conditional**.

Step 3: In the box with the placeholder **Untitled Task**, enter **Was the URL reported in CDL logs?.**

Step 4: In the conditional statement section left-side box, click the {} button. The Select Source For dialog box appears.

Step 5: In the search box, enter **SessionID**.



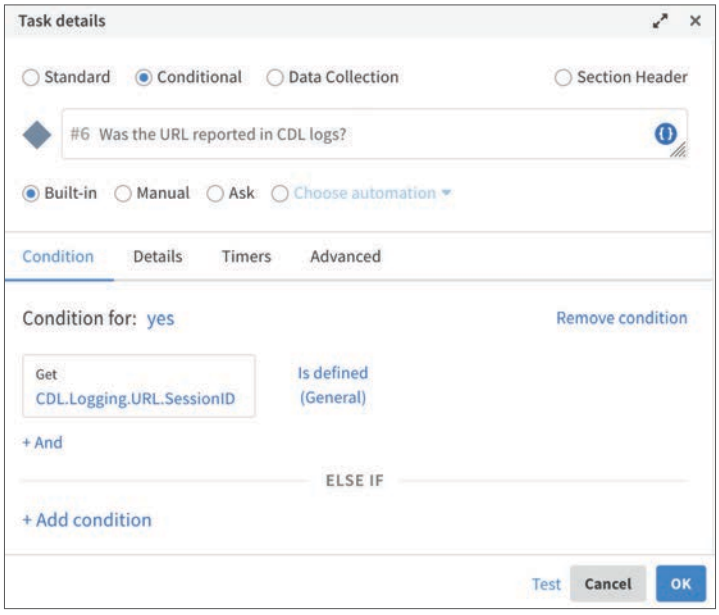
Step 6: In the CDL.Logging.URL section, click **SessionID**, and then click **Close**.

Next, you choose the comparison operator for the condition.

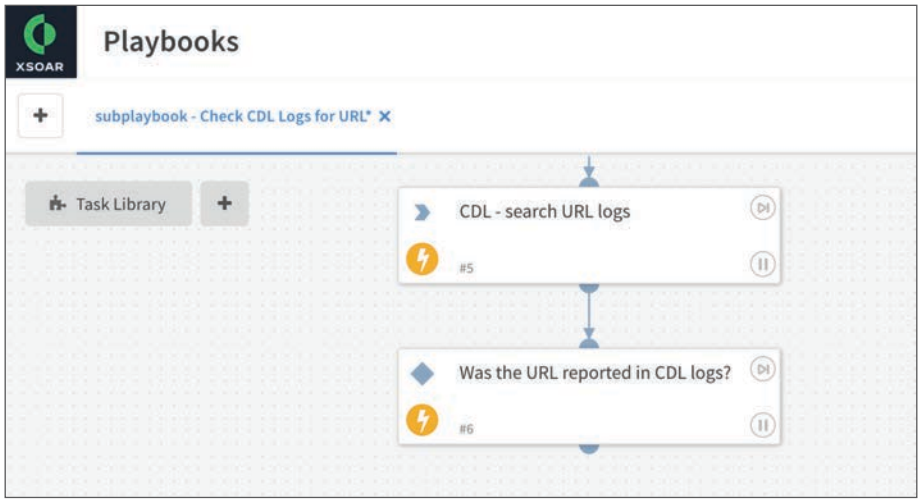
Step 7: Click **Equals**.

Step 8: In the search box, enter **Is defined**, and then click **Is defined**.

Step 9: Click the check, and then click **OK**.



Step 10: Verify that the task is now in your playbook.



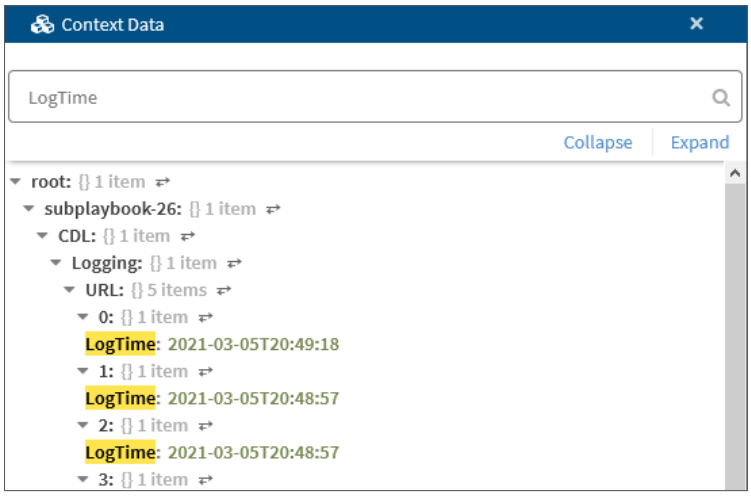
7.9 Create the "Set First Access Time For URL" Task

This is the first task in a new branch of the playbook. The playbook selects this branch only if Cortex Data Lake reported matching log entries for the tested URL.

This playbook creates an incident summary note for each matching URL that Cortex Data Lake reports. In this procedure, as part of the incident summary, you set the value of the temporary variable *URLFirstAccessedTime* to the timestamp of the earliest reported log entry.

This task uses the **Set** automation script.

To determine the earliest reported log entry, you use the *CDL.Logging.URL.LogTime* context data.



Step 1: From the Was the URL reported in CDL logs? task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below and to the right.

Step 2: In the Choose Label Name for Condition dialog box, select **yes**.

Step 3: Click **Save**. The Edit Task dialog box appears.

Step 4: In the box with the placeholder **Untitled Task**, enter **Set first access time for URL**.

Step 5: In the Choose Automation section, click the down arrow to open the search dialog box.

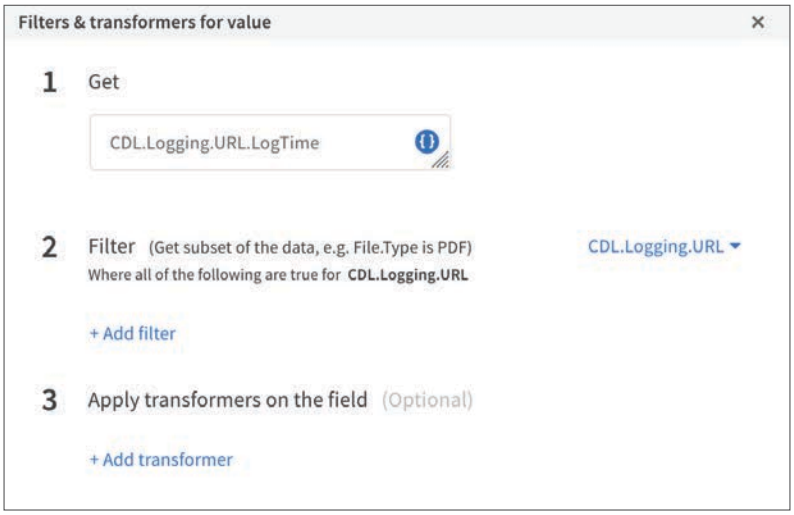
Step 6: In the **search** box, enter **Set**, and then choose **Set**. The task fields update.

Step 7: In the **key** box, enter **URLFirstAccessedTime**.

Step 8: In the value box, click the {} button. The Select Source for Value dialog box appears.

Step 9: In the search box, enter **LogTime**.

Step 10: In the CDL.Logging.URL section, in the row for **LogTime**, click **Filter & transform**. The dialog box name changes to Filters & Transformers for Value.



Now you transform the list of log time entries. First, you sort the log time entries by using the *Sort* transformer.

Step 11: In the Apply Transformers on the Field section, click **Add transformer**.

Step 12: Click **To upper case**.

Step 13: In the search box, enter **Sort** and then click **Sort**.

Step 14: To sort in ascending order, in the **descending** box, enter **false**.

Step 15: Click the check.

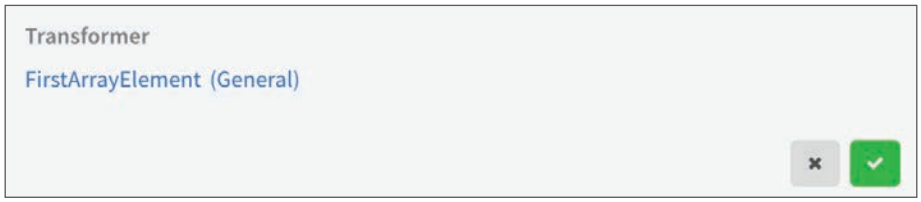
Next, now that the LogTime values are sorted in ascending order, you choose the lowest entry, using the *FirstArrayElement* transformer.

Step 16: In the Apply Transformers on the Field section, click **Add transformer**.

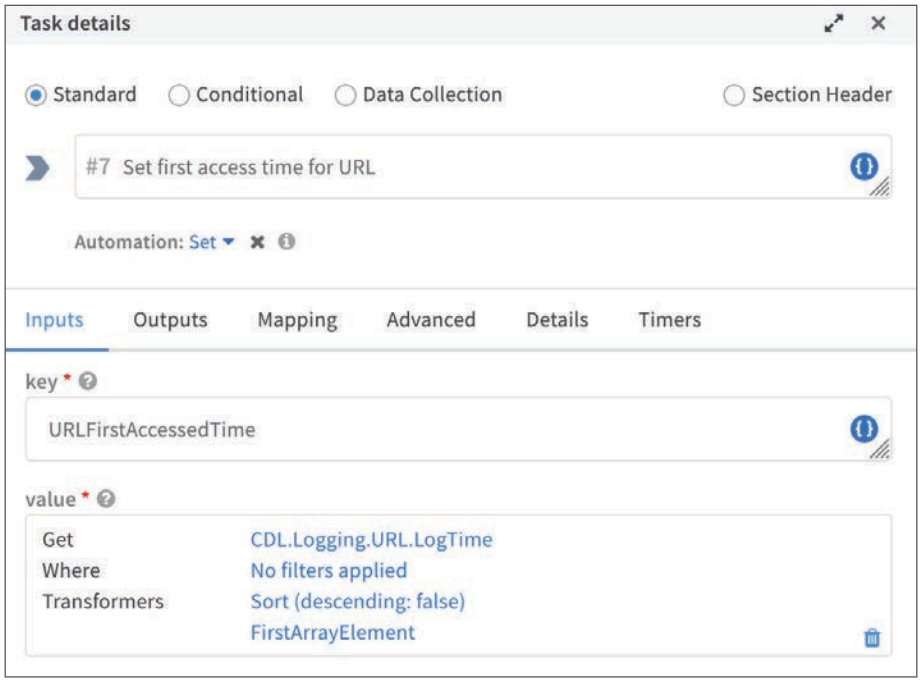
Step 17: Click **To upper case**.

Step 18: In the search box, enter **FirstArrayElement**, and then click **FirstArrayElement**.

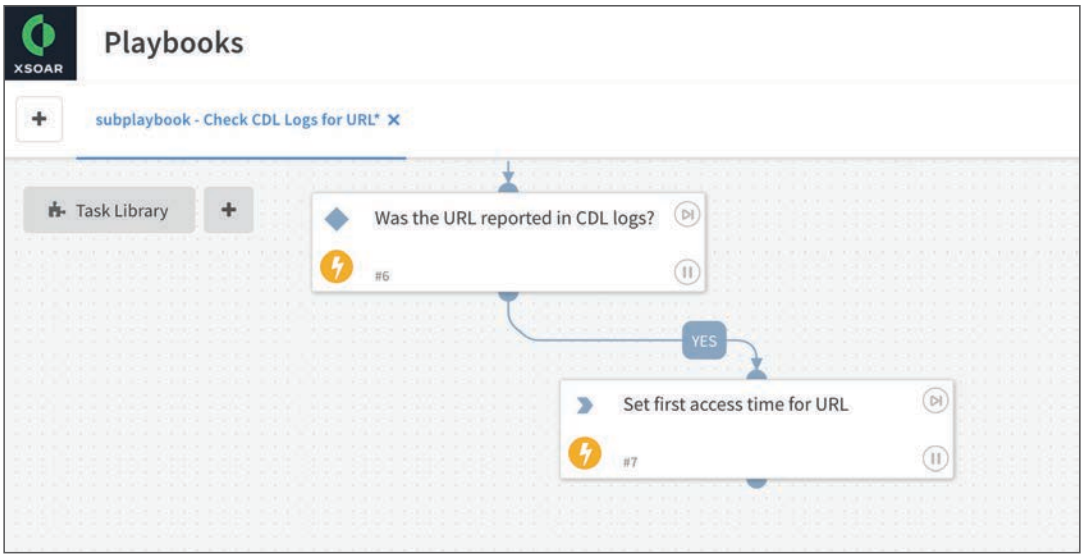
Step 19: Click the check, and then click **OK**.



Step 20: Verify the task configuration, and then click **OK**.



Step 21: Verify that the task is now in your playbook.

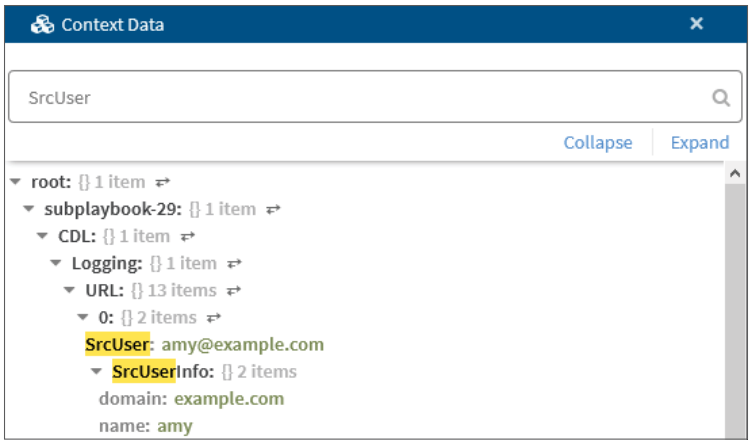


7.10 Create the "Set URL Accessed By List" Task

This playbook creates an incident summary note for each matching URL that Cortex Data Lake reports. In this procedure, as part of the incident summary, you set the value of the temporary variable *URLAccessedByList* to the list of all unique users that accessed the URL.

This task uses the **Set** automation script.

To determine the list of users that accessed the URL, you use the *CDL.Logging.URL.SrcUser* context data.



Step 1: From the **Set first access time for URL** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

Step 2: In the box with the placeholder **Untitled Task**, enter **Set URL accessed by list**.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

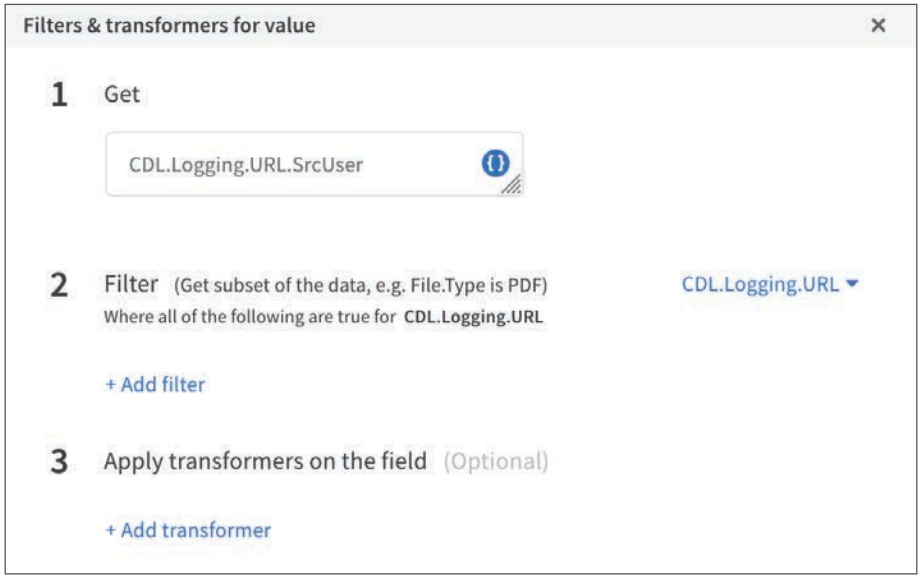
Step 4: In the **search** box, enter **Set**, and then choose **Set**. The task fields update.

Step 5: In the **key** box, enter **URLAccessedByList**.

Step 6: In the value box, click the {} button. The Select Source for Value dialog box appears.

Step 7: In the search box, enter **SrcUser**.

Step 8: In the CDL.Logging.URL section, in the row for **SrcUser**, click **Filter & transform**. The dialog box name changes to Filters & Transformers for Value.



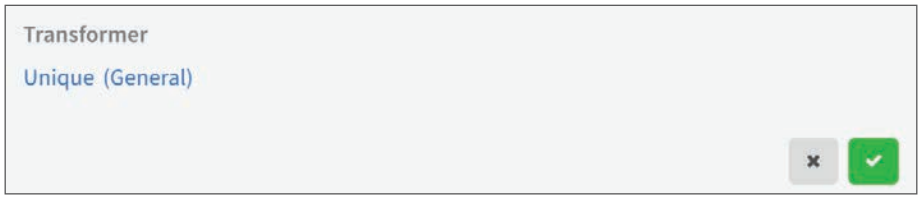
Now you eliminate any duplicate entries by using the *Unique* transformer.

Step 9: In the Apply Transformers on the Field section, click **Add transformer**.

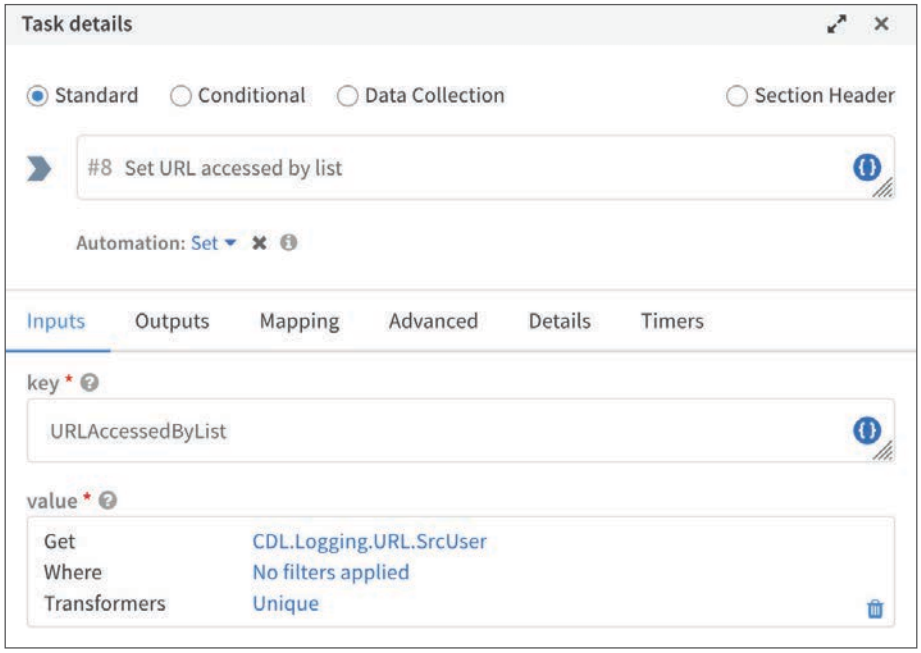
Step 10: Click **To upper case**.

Step 11: In the search box, enter **Unique**, and then click **Unique**.

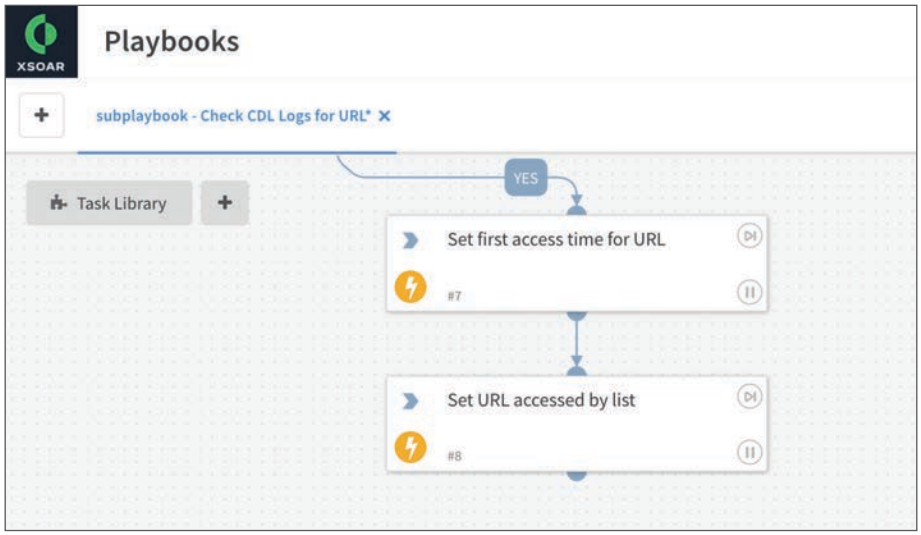
Step 12: Click the check, and then click **OK**.



Step 13: Verify the task configuration, and then click **OK**.



Step 14: Verify that the task is now in your playbook.



7.11 Create the "Set URL Access Counter" Task

This playbook creates an incident summary note for each matching URL that Cortex Data Lake reports. In this procedure, as part of the incident summary, you set the value of the temporary variable *URLAccessCount* to the list of all unique users that accessed the URL.

This task uses the **Set** automation script.

To determine how many times users accessed the URL, you use the *CDL.Logging.URL.SessionID* context data and count the number of unique session IDs.

Step 1: From the **Set URL accessed by list** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

Step 2: In the box with the placeholder **Untitled Task**, enter **Set URL access counter**.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

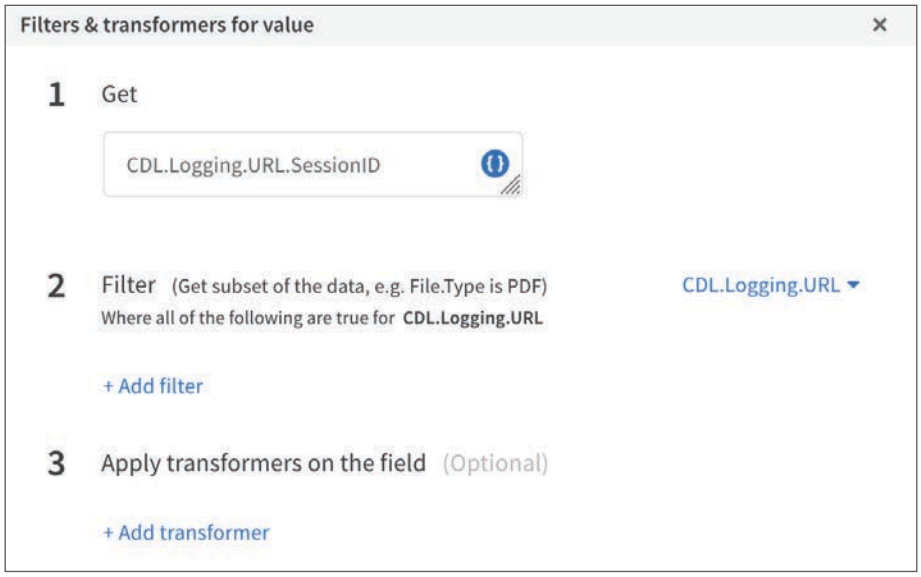
Step 4: In the **search** box, enter **Set**, and then choose **Set**. The task fields update.

Step 5: In the **key** box, enter **URLAccessCount**.

Step 6: In the value box, click the {} button. The Select Source for Value dialog box appears.

Step 7: In the search box, enter **SessionID**.

Step 8: In the CDL.Logging.URL section, in the row for **SessionID**, click **Filter & transform**. The dialog box name changes to Filters & Transformers for Value.



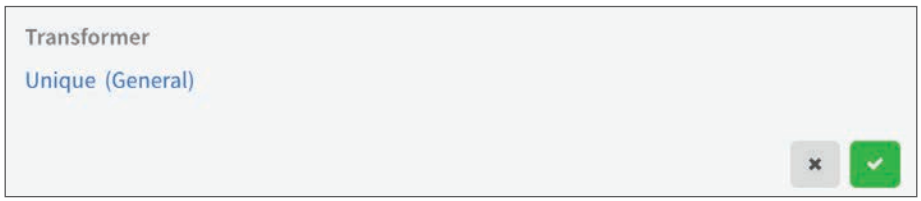
Now you transform the list of session IDs. First, you eliminate any duplicate entries by using the *Unique* transformer.

Step 9: In the Apply Transformers on the Field section, click **Add transformer**.

Step 10: Click **To upper case**.

Step 11: In the search box, enter **Unique** and then click **Unique**.

Step 12: Click the check.



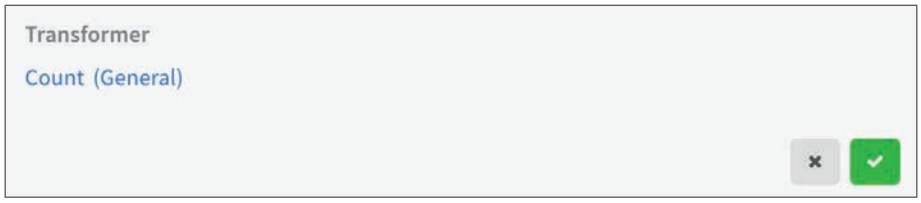
Next, now that you have eliminated any duplicate entries, you count the number of entries by using the *Count* transformer.

Step 13: In the Apply Transformers on the Field section, click **Add transformer**.

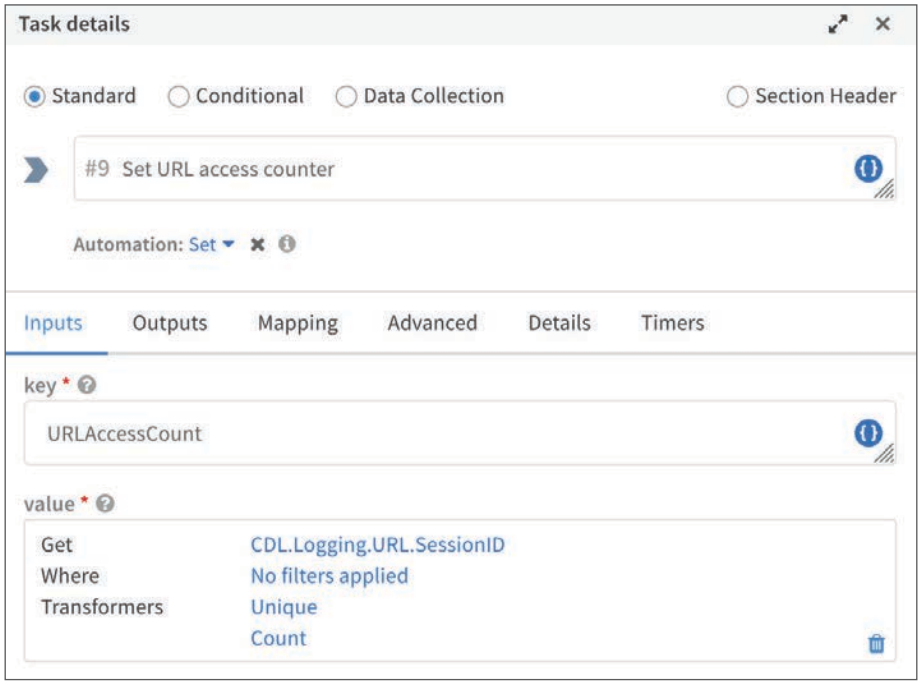
Step 14: Click **To upper case**.

Step 15: In the search box, enter **Count**, and then click **Count**.

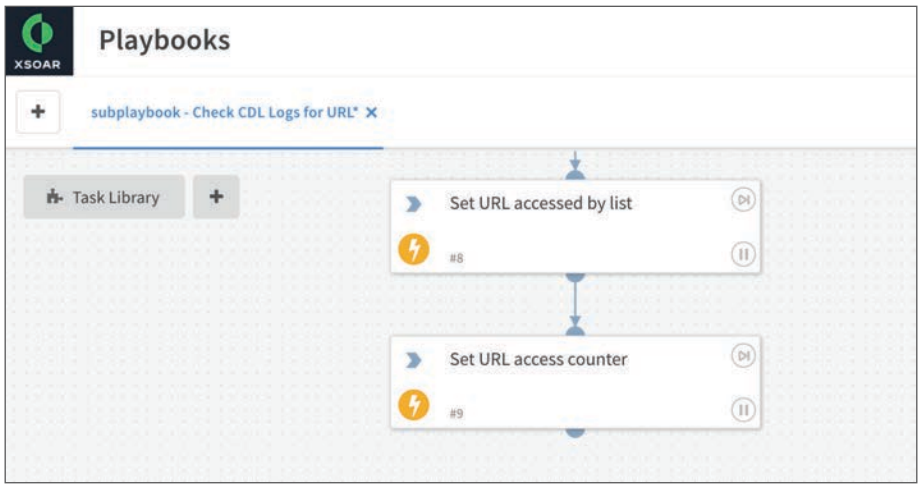
Step 16: Click the check, and then click **OK**.



Step 17: Verify the task configuration, and then click **OK**.



Step 18: Verify that the task is now in your playbook.



7.12 Create the "Mark as Note - URL Access Summary" Task

As a final step, the playbook summarizes the query results with an incident note.

In previous procedures, you have set the following temporary variables:

- URLFirstAccessedTime
- URLAccessedByList
- URLAccessCount

In this procedure, you craft a summary note using these temporary variables.

This task uses the **Print** automation script. You configure advanced settings for this task to mark the results as an incident note.

Step 1: From the Set URL access counter task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

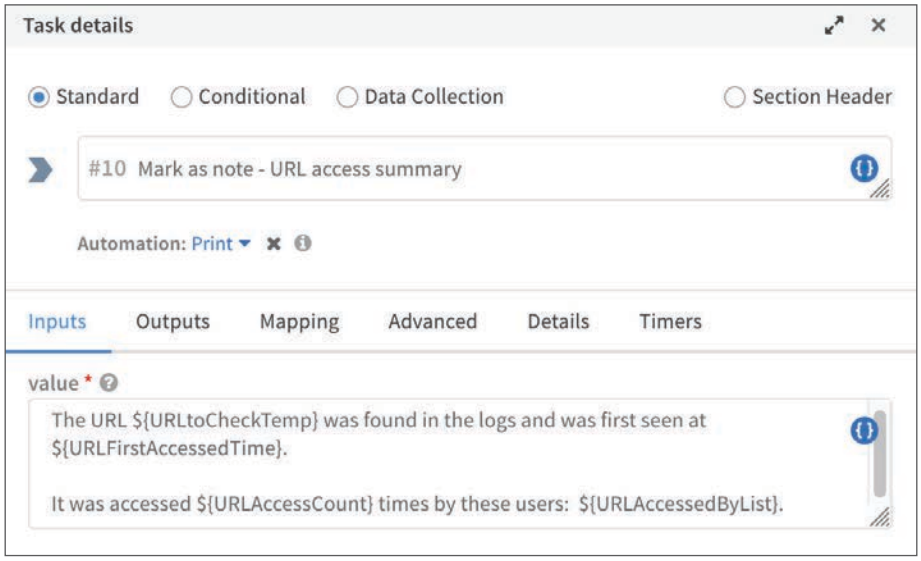
Step 2: In the box with the placeholder **Untitled Task**, enter Mark as note - URL access summary.

Step 3: In the Choose Automation section, click the down arrow to open the search dialog box.

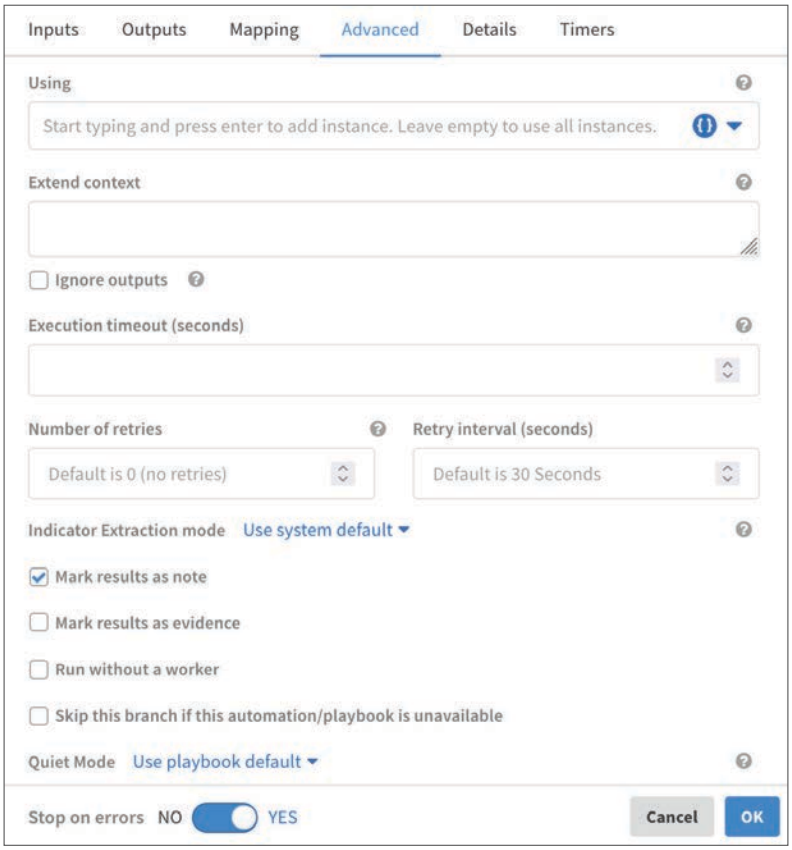
Step 4: In the **search** box, enter **Print**, and then choose **Print**. The task fields update.

Step 5: In the **value** box, enter the following text.

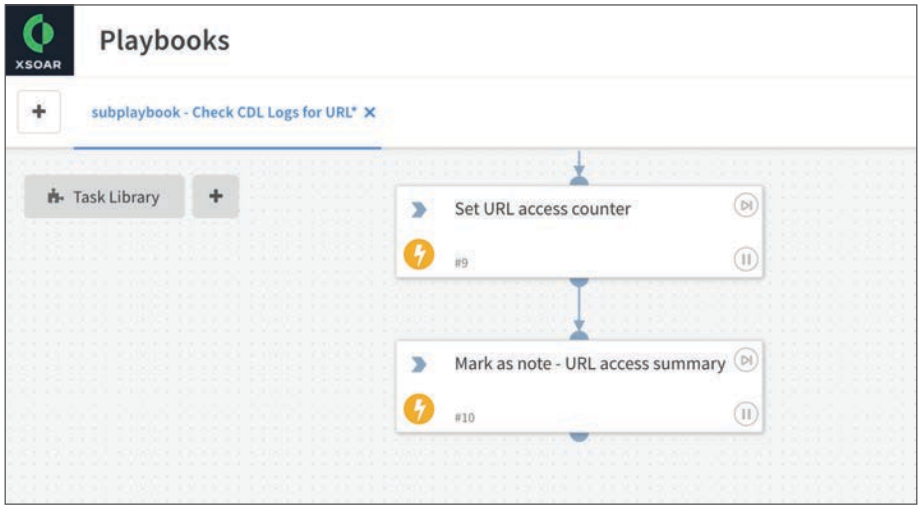
```
The URL ${URLtoCheckTemp} was found in the logs and was first seen at
${URLFirstAccessedTime}.


It was accessed ${URLAccessCount} times by these users: ${URLAccessedByList}.
```



Step 6: On the Advanced tab, select **Mark results as note**, and then click **OK**.



Step 7: Verify that the task is now in your playbook.



7.13 Create the "Mark as Note - No Malicious URLs Reported" Task

As a final step, the playbook summarizes the query results with an incident note.

This is the only task in a new branch of the playbook. The playbook selects this branch only if Cortex Data Lake did not report any matching log entries for the tested URL.

You do not have to further examine any context data for this task.

This task uses the **Print** automation script. You configure advanced settings for this task in order to mark the results as an incident note.

Step 1: From the **Was the URL reported in CDL logs?** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below and to the left. For clarity, you should horizontally align the task with the **Mark as note - URL access summary** task.

Step 2: In the Choose Label Name for Condition dialog box, select **Mark as 'else' case**.

Step 3: Click **Save**. The Edit Task dialog box appears.

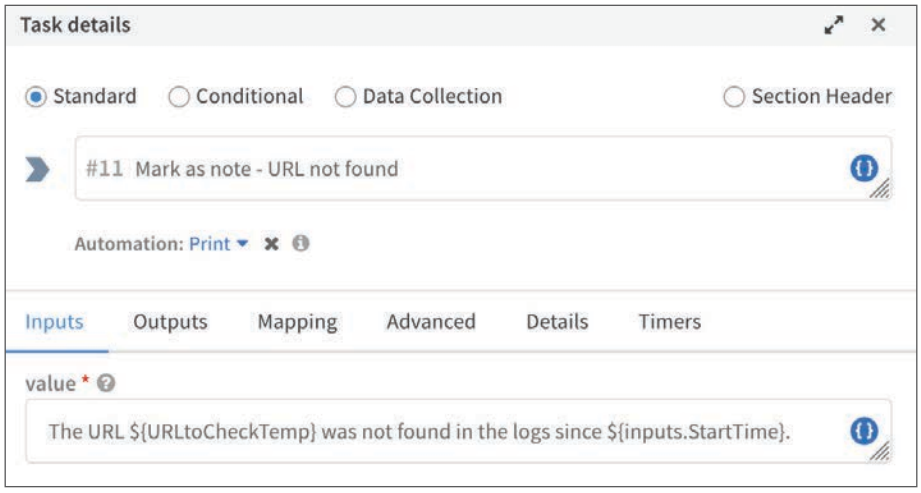
Step 4: In the box with the placeholder **Untitled Task**, enter **Mark as note - URL not found**.

Step 5: In the Choose Automation section, click the down arrow to open the search dialog box.

Step 6: In the **search** box, enter **Print**, and then choose **Print**. The task fields update.

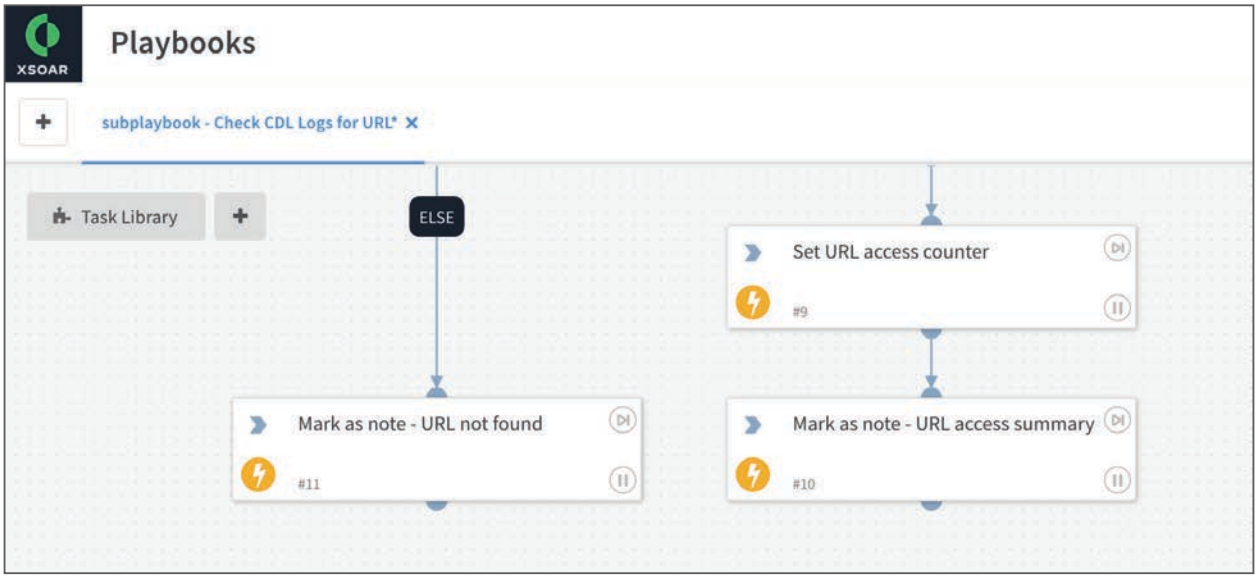
Step 7: In the **value** box, enter the following text.

```
The URL ${URLtoCheckTemp} was not found in the logs since ${inputs.StartTime}.
```



Step 8: On the Advanced tab, select **Mark results as note**, and then click **OK**.

Step 9: Verify that the task is now in your playbook.



7.14 Create the "Done" Task

As a best practice, you should create a Done section header that terminates this playbook.

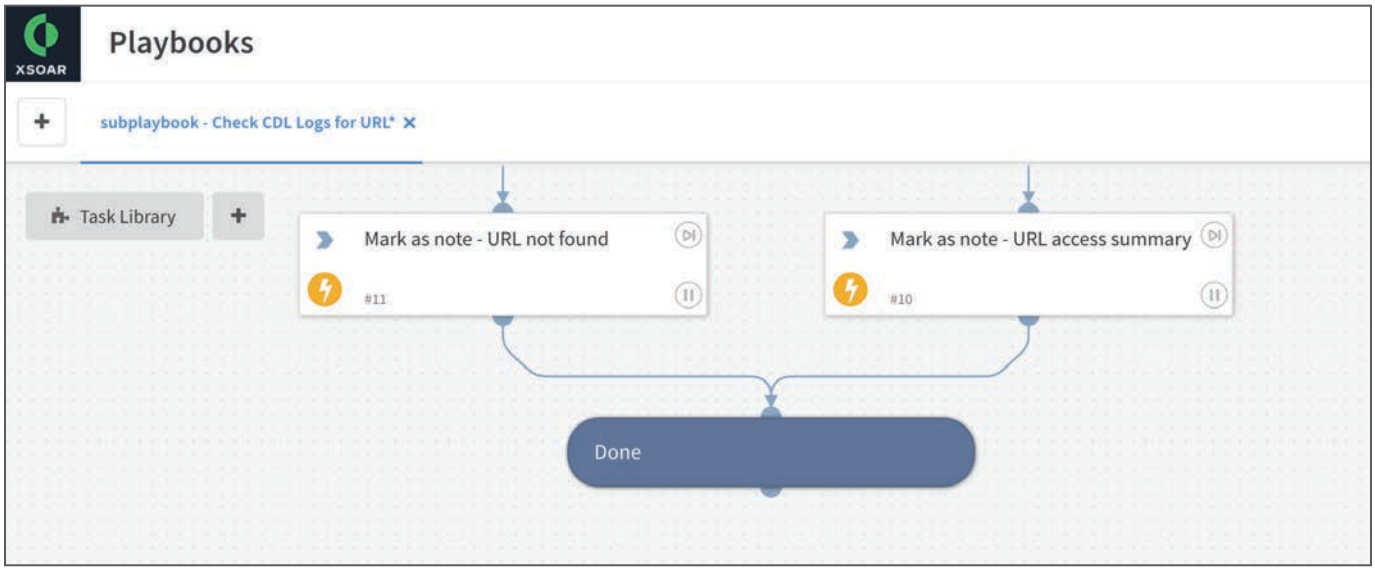
Step 1: From the Mark as note - URL not found task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled blank task below and to the center. The Edit Task dialog box appears.

Step 2: Select **Section Header**.

Step 3: In the box with the placeholder **Untitled Task**, enter Done, and then click **OK**.

Step 4: From the Mark as Note - URL access summary task egress node, drag the task connector line to Done task ingress node, and then release to create an additional connection to the Done task.

Step 5: Verify that the task is now in your playbook.

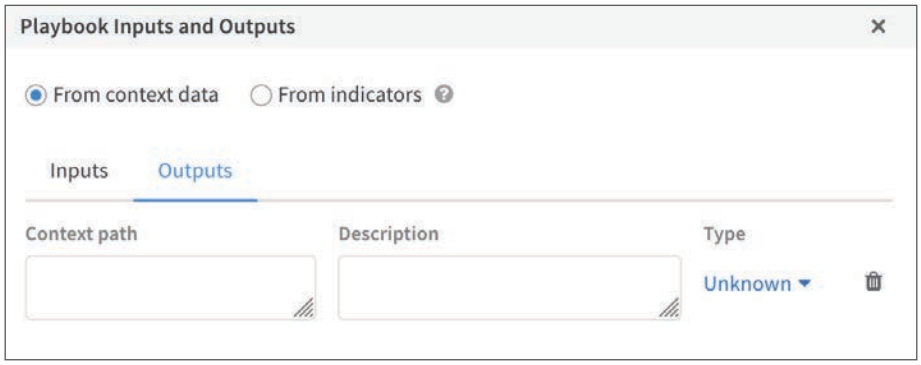


7.15 Add Playbook Output

By default, the temporary variables that you set in a sub-playbook are not available in the parent playbook. To use a derived item from a sub-playbook in a parent playbook, add the item as an output to the sub-playbook.

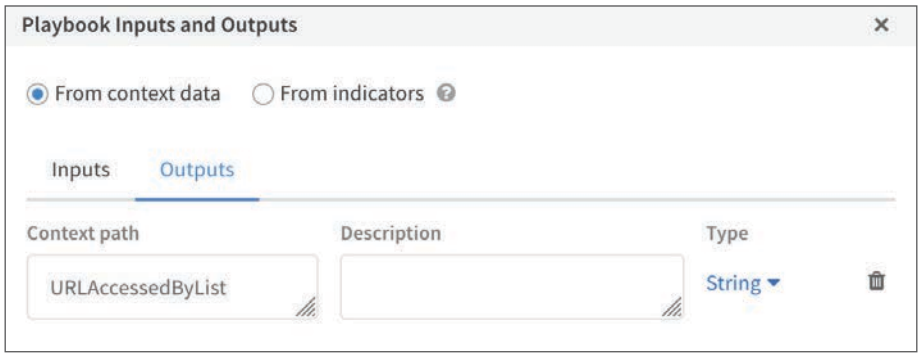
Step 1: Click the Playbook Triggered section header.

Step 2: In the Playbook Inputs and Outputs dialog box, on the Outputs tab, click **Add manually**.



Step 3: In the **Context path** box, enter **URLAccessedByList**.

Step 4: In the Type column, click **Unknown**, and then choose **String**.



Step 5: Click **Save**.

Step 6: To save the playbook, click **Save Playbook**.

Procedures

Modifying the Playbook to Check If Users Accessed Malicious URLs

8.1 Create URL Filtering Profile

8.2 Apply URL Filtering Profile

8.3 Create a Separator Task

8.4 Add the "Check CDL Logs For URL" Sub-Playbook Task

8.5 Create the "Did any users access malicious URLs?" Task

8.6 Add the "Email to List" Sub-Playbook Task

8.7 Connect Playbook Tasks to the Done Task

8.8 Verify Complete Playbook

As mentioned previously, the basic playbook performs only an automated analysis. In this section, you update the basic playbook to include an option to analyze security logs and check if any users tried to access any malicious URLs from the phishing message.

This set of procedures assumes that the users access the network through a connection secured by Prisma Access and that the subscription is already active. These procedures also assume that Prisma Access is sending security logs to a previously activated Cortex Data Lake instance.

Your security policy for Prisma Access must generate URL logs for all web traffic. Before you modify the playbook, complete the following procedures to create a URL filtering profile and apply the profile to the security policy rules that permit web traffic.

When you update the basic playbook, you include the following sub-playbooks:

- **Email to List**—Playbook to send separate individual emails to all recipients in a list. You created this sub-playbook in Procedure 6.1 through Procedure 6.3.
- **Check CDL Logs for URL**—Playbook to check Cortex Data Lake for URL log entries. You completed required prerequisites and created this sub-playbook in Procedure 7.1 through Procedure 7.15.

8.1 Create URL Filtering Profile

Prisma Access does not generate URL log entries when the site access action is set to *allow*. To generate URL log entries for a category, you must set the site access action for the category to: *alert*, *block*, *continue*, or *override*.

In this procedure, you clone the default URL filtering profile and then change the site access action for categories with an *allow* action to alert. Do not change the site access action for categories with a default *block* action.

Note

By applying this URL filtering profile, Prisma Access blocks access to web sites classified as malware or phishing. If a user tries to access sites in these categories, Prisma Access generates URL log entries with an action of "block-url."

Step 1: Log in to Panorama.

Step 2: In **Device Groups > Objects**, at the top of the page, in the **Device Group** list, choose **Mobile_User_Device_Group**.

Step 3: In **Objects > Security Profiles > URL Filtering**, select **default**.

Step 4: Click **Clone**, and then click **OK**.

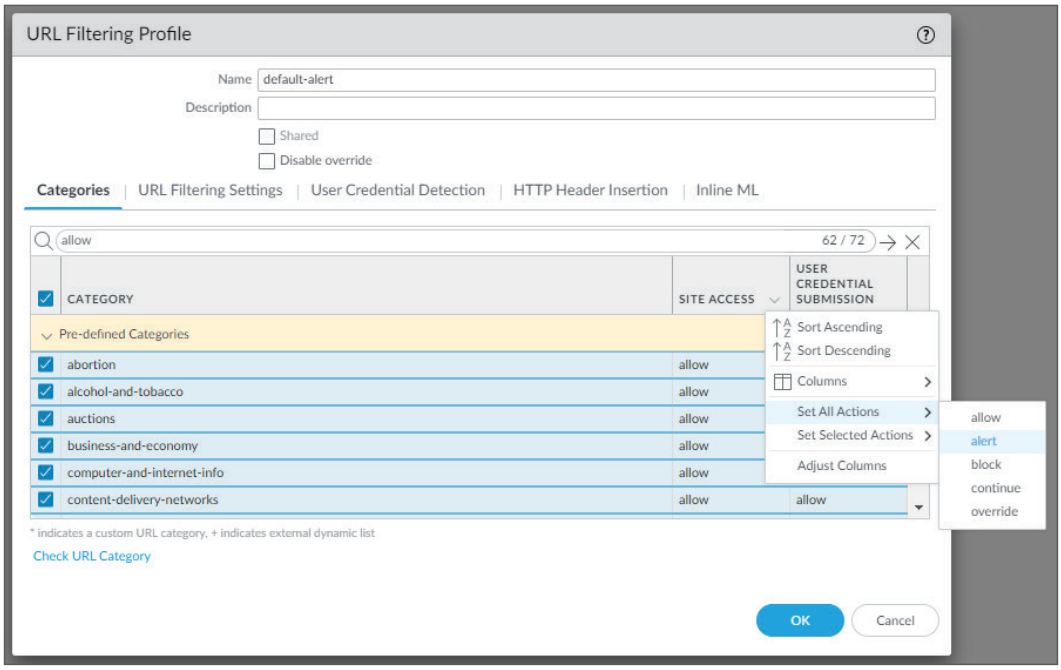
Step 5: Click the cloned profile (example: default-1).

Step 6: In the **Name** box, enter default-alert.

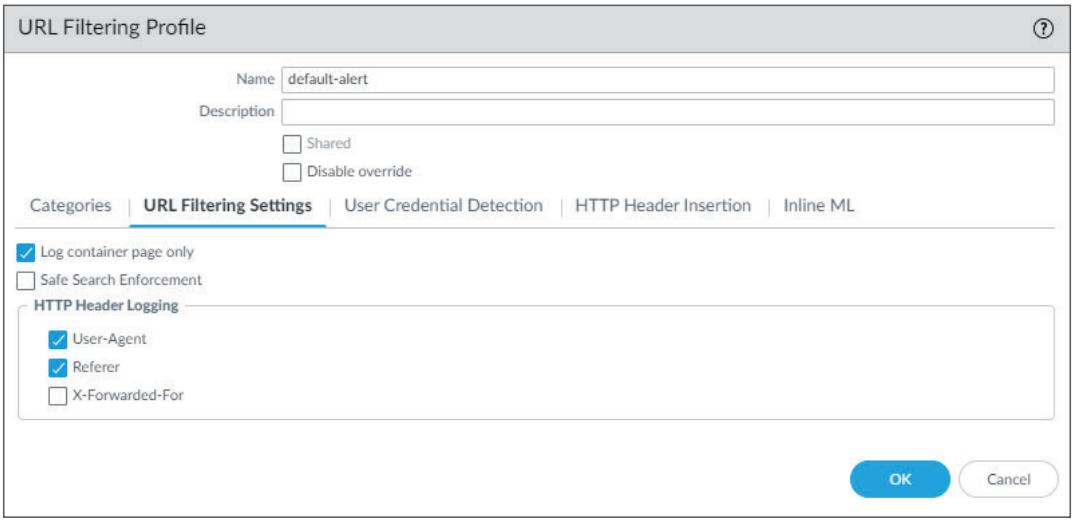
Step 7: In the search box, enter **allow**.

Step 8: In the top row, select all categories that match the search.

Step 9: In the Site Access column, click the down arrow, and then click **Set All Actions > alert**.



Step 10: On the URL Filtering Settings tab, in the HTTP Header Logging section, select **User-Agent** and **Referer**, and then click **OK**.



8.2 Apply URL Filtering Profile

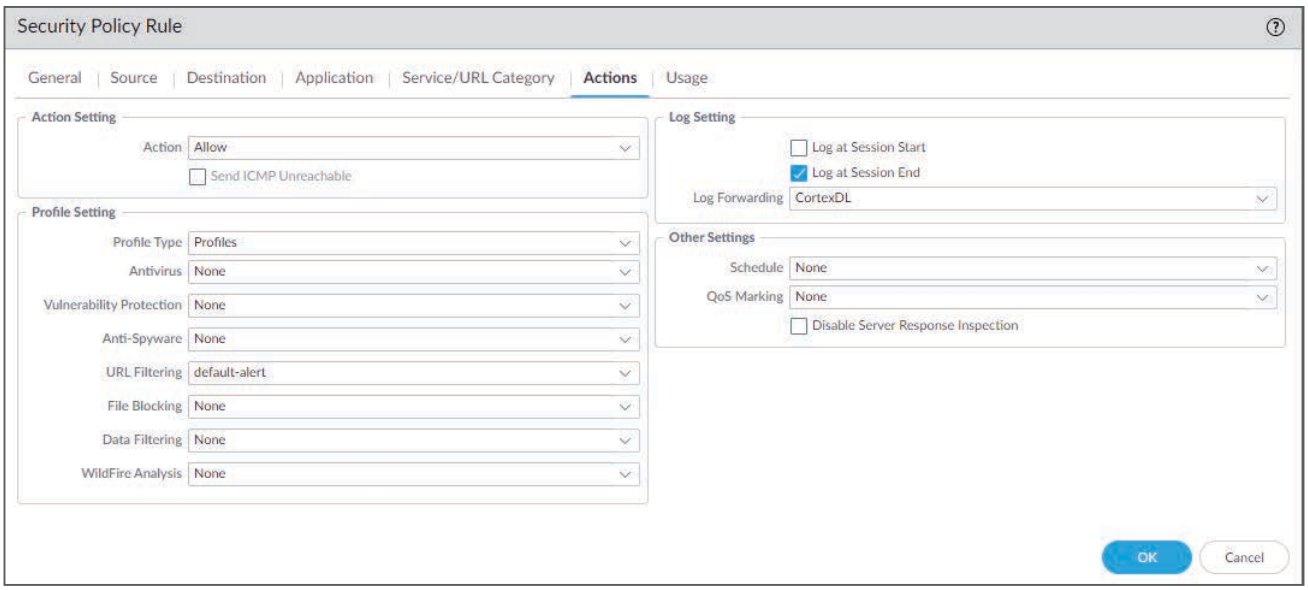
For Prisma Access to report URL logs, you must apply the URL filtering profile you created in Procedure 8.1 to the security policy rule that matches HTTP/HTTPS traffic from connected users.

Step 1: In **Policies > Security**, click on the rule that permits HTTP/HTTPS traffic to the internet (example: **RemoteSite-to-Internet**).

Step 2: On the Actions tab, in the Profile Setting section, in the **Profile Type** box, choose **Profiles**.

Step 3: In the Profile Setting section, in the **URL Filtering** box, choose **default-alert**.

Step 4: In the Log Setting section, select **Log at Session End**, and then click **OK**.



8.3 Create a Separator Task

In this procedure, you choose the playbook you previously created in Procedure 2.1 and switch to edit mode. You can then proceed with creating additional tasks.

As a best practice, you should create a Respond If User Accessed Malicious URL section header that begins this section of the playbook.

Step 1: In Cortex XSOAR, in the navigation pane, click **Playbooks**.

Step 2: In the search box, enter **Automated Phishing Investigation**, and then press ENTER.

Step 3: In the search results, click **Automated Phishing Investigation**.

Step 4: Click **Edit**.

Step 5: From the **Mark as note - malicious URL detected** task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled blank task below and to the right. For clarity, you should horizontally align the task with the **Prepare to delete email messages** task. The Edit Task dialog box appears.

Step 6: Select **Section Header**.

Step 7: In the box with the placeholder **Untitled Task**, enter **Respond if user accessed malicious URL**, and then click **OK**.

8.4 Add the "Check CDL Logs For URL" Sub-Playbook Task

In this task, you use a custom playbook as a sub-playbook. The sub-playbook uses the **cdl-query-url-logs** automation command from the Cortex Data Lake integration.

Your playbook runs this sub-playbook once for each unique malicious URL that WildFire reported in Procedure 2.9. The automation command also requires you to specify a start time, for this value you use the *incident.firstseen* context data that you set in Procedure 2.5.

After this sub-playbook executes, Cortex XSOAR adds any user who has accessed a malicious URL to the *URLAccessedByList* context data.

Step 1: On the playbook workspace, click **Task Library**.

Step 2: On the Playbooks tab, in the search box enter **subplaybook - check cdl logs for url**. The search box only accepts lowercase characters.

Step 3: In the search results, for the playbook **subplaybook - Check CDL Logs for URL**, click **Add**. Cortex XSOAR adds the sub-playbook task to the playbook workspace.

Step 4: From the **Respond if user accessed malicious URL** task egress node, drag the task connector line to the **subplaybook - Check CDL Logs for URL** task ingress node and release. Cortex XSOAR adds a connector line between the tasks.

Step 5: Click the **subplaybook - Check CDL Logs for URL** task.

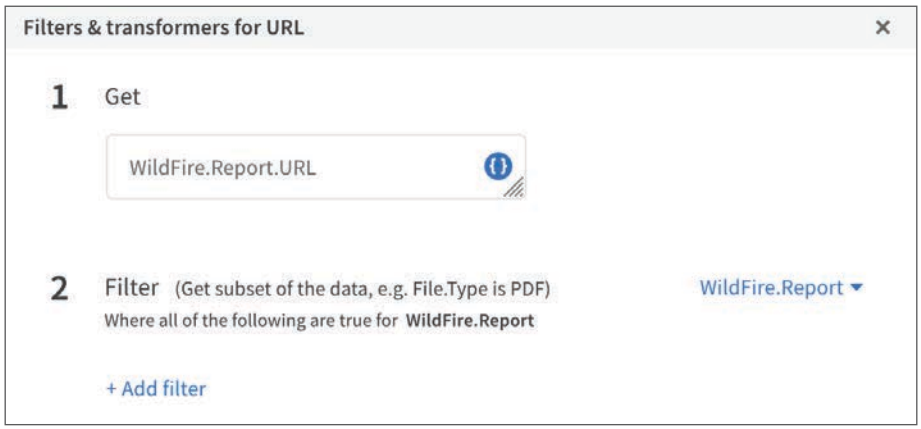
Step 6: On the Task details dialog box, change the task name by removing **subplaybook** so the name is **Check CDL Logs for URL**.

Step 7: In the **StartTime** box, enter **${incident.firstseen}**.

Step 8: In the **URL** box, click the {} button. The Select Source For URL dialog box appears.

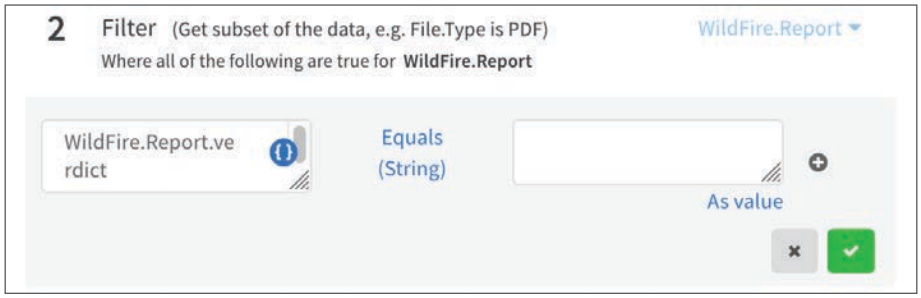
Step 9: Click **Filters And Transformers**. The dialog box name changes to Filters & Transformers for URL.

Step 10: In the **Get** box, enter **WildFire.Report.URL**. This value is case sensitive.



Step 11: Click Add filter.

Step 12: In the conditional statement left-side box, enter **WildFire.Report.verdict**. This value is case sensitive.



Next, you choose the comparison operator for the filter.

Step 13: Click **Equals**.

Step 14: In the search box, enter **Contains**, and then click **Contains**.

Step 15: In the right-side box, enter **phishing**. This value is case sensitive.

Step 16: To add a second filter condition, Click the **+**.

Step 17: In the conditional statement left-side box, enter **WildFire.Report.verdict**. This value is case sensitive.

Next, you choose the comparison operator for the filter.

Step 18: Click **Equals**.

Step 19: In the search box, enter **Contains**, and then click **Contains**.

Step 20: In the right-side box, enter **malware**. This value is case sensitive.

Step 21: Click the check.

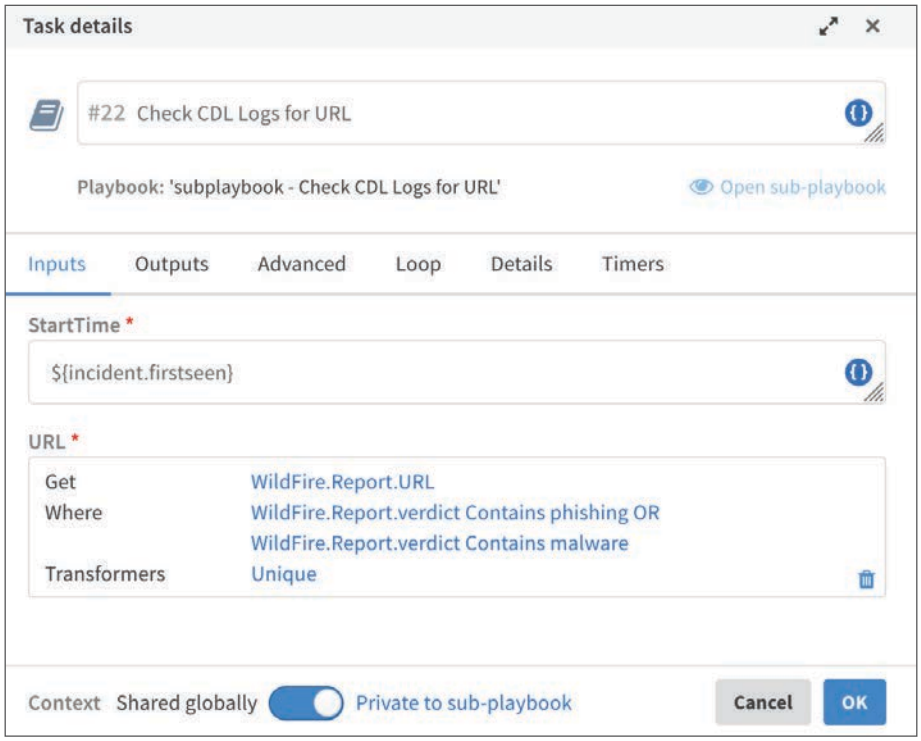
Next, you eliminate duplicates using the *unique* transformer.

Step 22: In the Apply Transformers on the Field section, click **Add Transformer**.

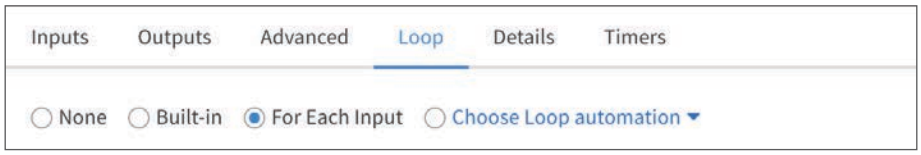
Step 23: Click **To upper case**.

Step 24: In the search box, enter **Unique** and then click **Unique**.

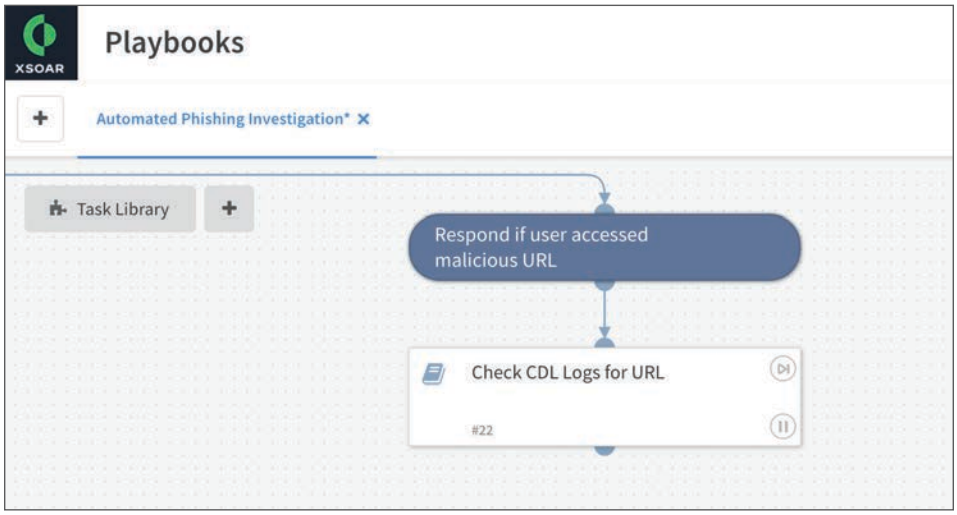
Step 25: Click the check, and then click **OK**.



Step 26: On the Loop tab, select **For Each Input**, and then click **OK**.



Step 27: Verify that the task is now in your playbook.



8.5 Create the "Did any users access malicious URLs?" Task

In this procedure, you perform a check to see if the **Check CDL Logs for URL** sub-playbook found log entries for any malicious URLs. Your playbook executes different branches depending on the results of the check.

To perform the check, your conditional statement uses the *URLAccessedByList* context data. If this object is empty, then users did not access any malicious URLs. Otherwise, one or more users did access a malicious URL.

Step 1: From the **Check CDL Logs for URL** sub-playbook task egress node, drag the task connector line to the playbook workspace, and then release to create an untitled task below. The Edit Task dialog box appears.

Step 2: Select **Conditional**.

Step 3: In the box with the placeholder **Untitled Task**, enter **Did any users access malicious URLs?**.

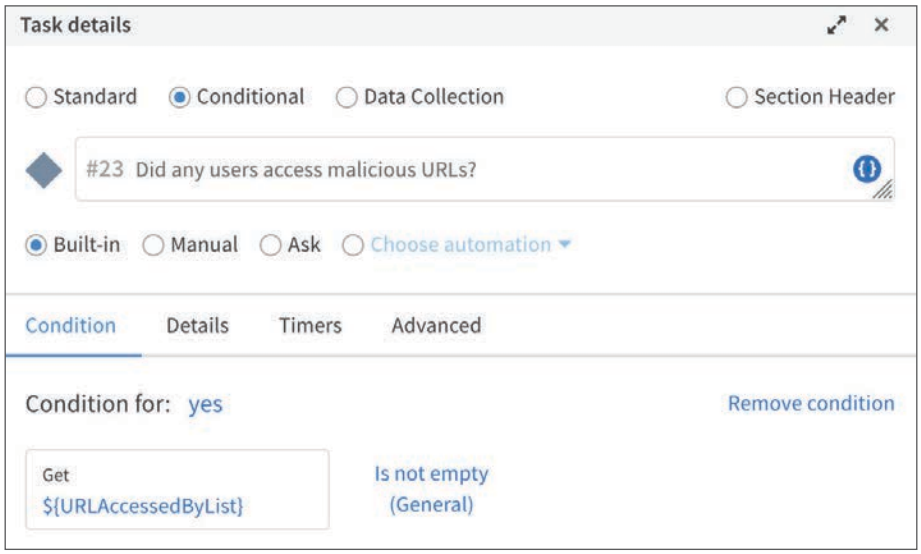
Step 4: In the conditional statement section left-side box, enter **${URLAccessedByList}**.

Next, you choose the comparison operator for the condition.

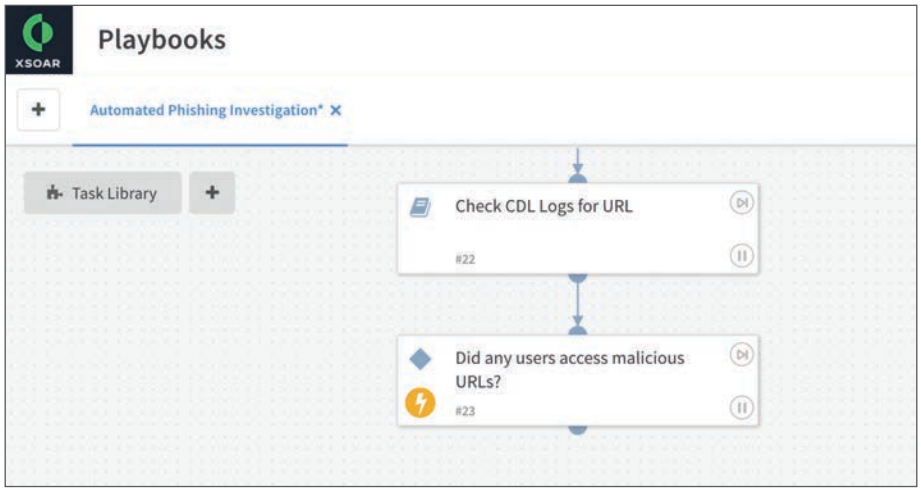
Step 5: Click **Equals**.

Step 6: In the search box, enter **Is not empty**, and then click **Is not empty**.

Step 7: Click the check, and then click **OK**.



Step 8: Verify that the task is now in your playbook.



8.6 Add the "Email to List" Sub-Playbook Task

This is the first task in a new branch of the playbook. The playbook selects this branch only if Cortex Data Lake reported matching log entries for the tested URL.

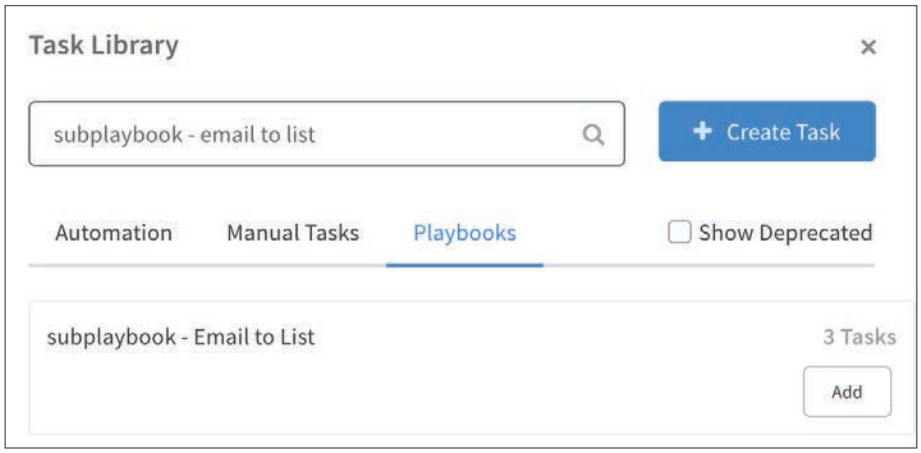
You use a custom playbook as a sub-playbook in this task. The sub-playbook uses the **send-email** automation command from the EWS Mail Sender integration.

Your playbook runs this sub-playbook once for each email recipient in a list.

Step 1: On the playbook workspace, click **Task Library**.

Step 2: On the Playbooks tab, in the search box, enter **subplaybook - email to list**. The search box accepts only lowercase characters.

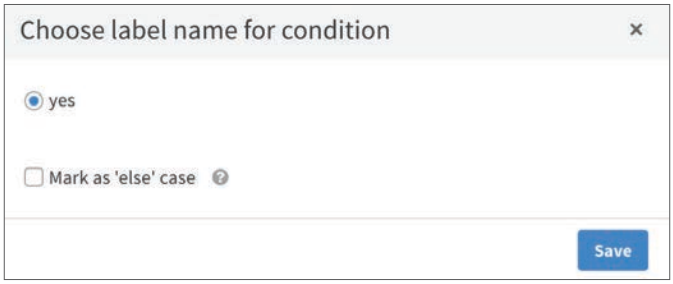
Step 3: In the search results, for the playbook **subplaybook - email to list**, click **Add**. Cortex XSOAR adds the sub-playbook task to the playbook workspace.



Step 4: Move the sub-playbook task below and to the right of the **Did any users access malicious URLs?** conditional task.

Step 5: From the **Did any users access malicious URLs?** task egress node, drag the task connector line to the **subplaybook - Email to List** task ingress node and release.

Step 6: In the Choose Label Name for Condition dialog box, select **yes**.



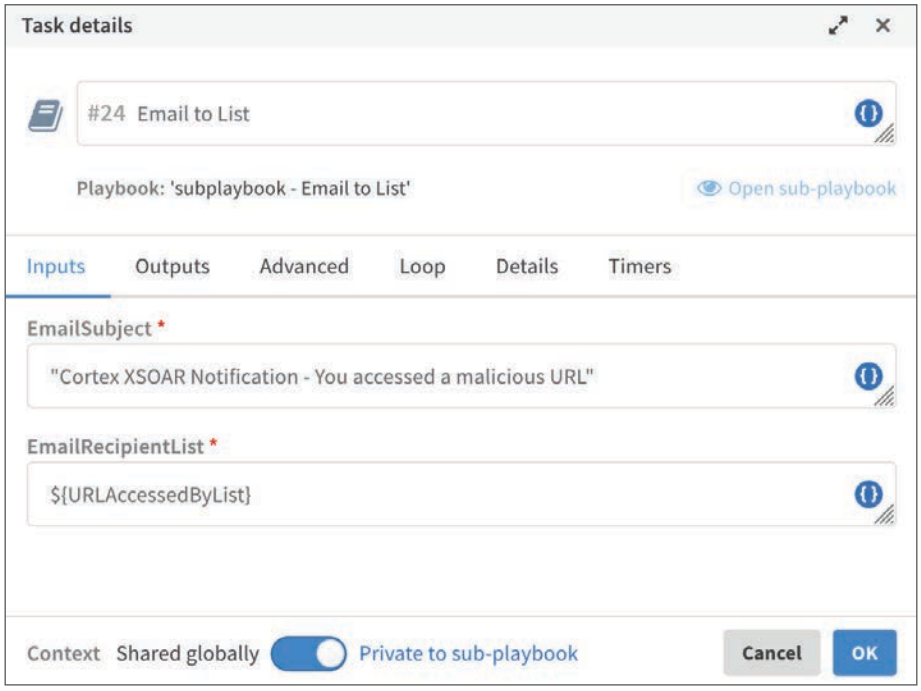
Step 7: Click **Save**. Cortex XSOAR adds a connector line between the tasks.

Step 8: Click the **subplaybook - Email to List**.

Step 9: On the Task details dialog box, change the task name by removing **subplaybook** so the name is **Email to List**.

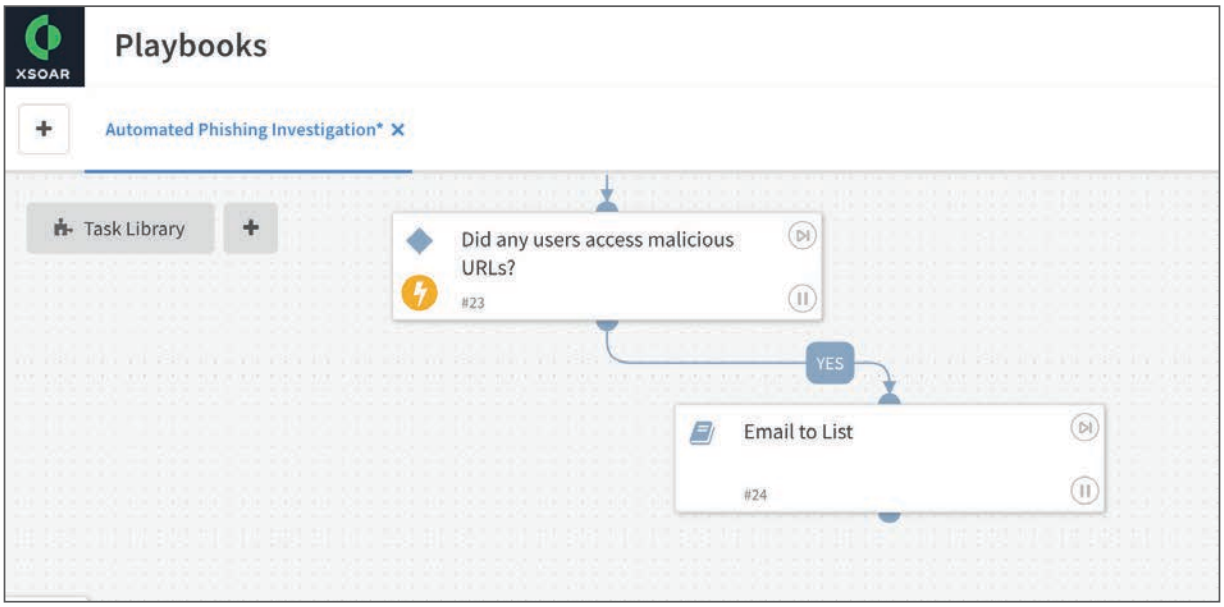
Step 10: In the **EmailSubject** box, enter **"Cortex XSOAR Notification - You accessed a malicious URL"**.

Step 11: In the **EmailRecipientList** box, enter **${URLAccessedByList}**.



Step 12: On the Loop tab, select **For Each Input**, and then click **OK**.

Step 13: Verify that the task is now in your playbook.



8.7 Connect Playbook Tasks to the Done Task

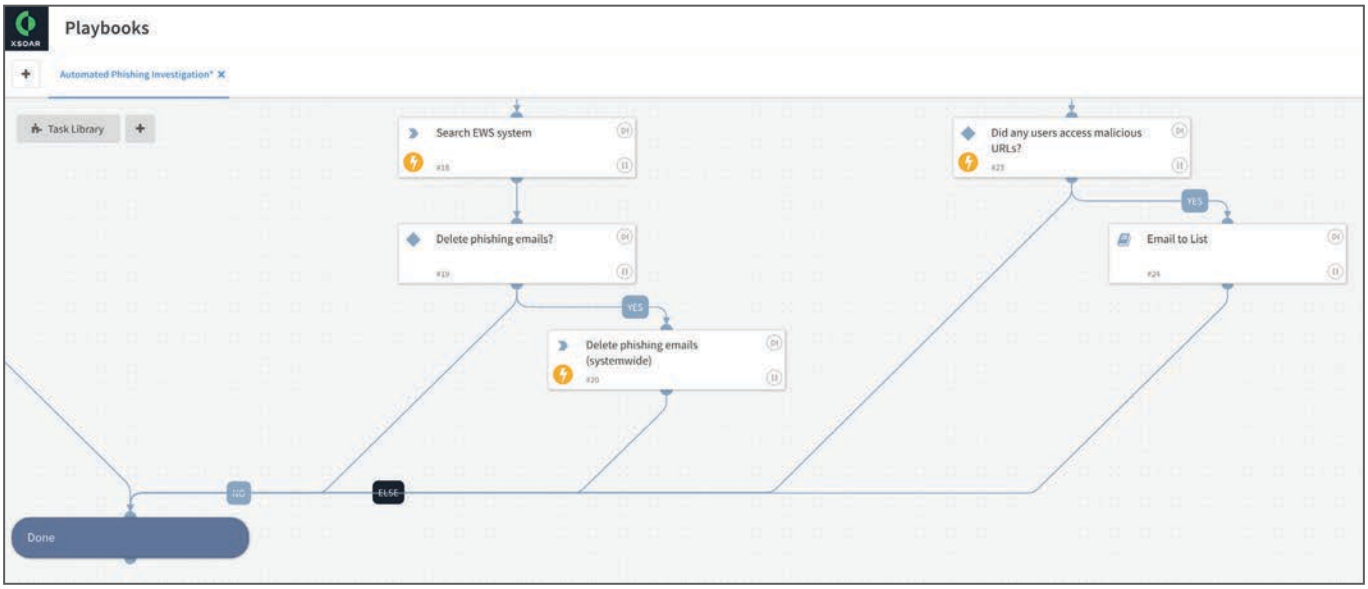
If you have completed Procedure 4.4, your playbook already contains a Done section header. In this procedure, you connect existing tasks to the Done section header.

Step 1: From the Did any users access malicious URLs? task egress node, drag the task connector line to the Done task ingress node and release to create an additional connection to the Done task.

Step 2: In the Choose Label Name for Condition dialog box, select **Mark as 'else' case**, and then click **Save**.

Step 3: From the Email to List task egress node, drag the task connector line to the Done task ingress node and release to create an additional connection to the Done task.

Step 4: Verify that the connector lines are now in your playbook.



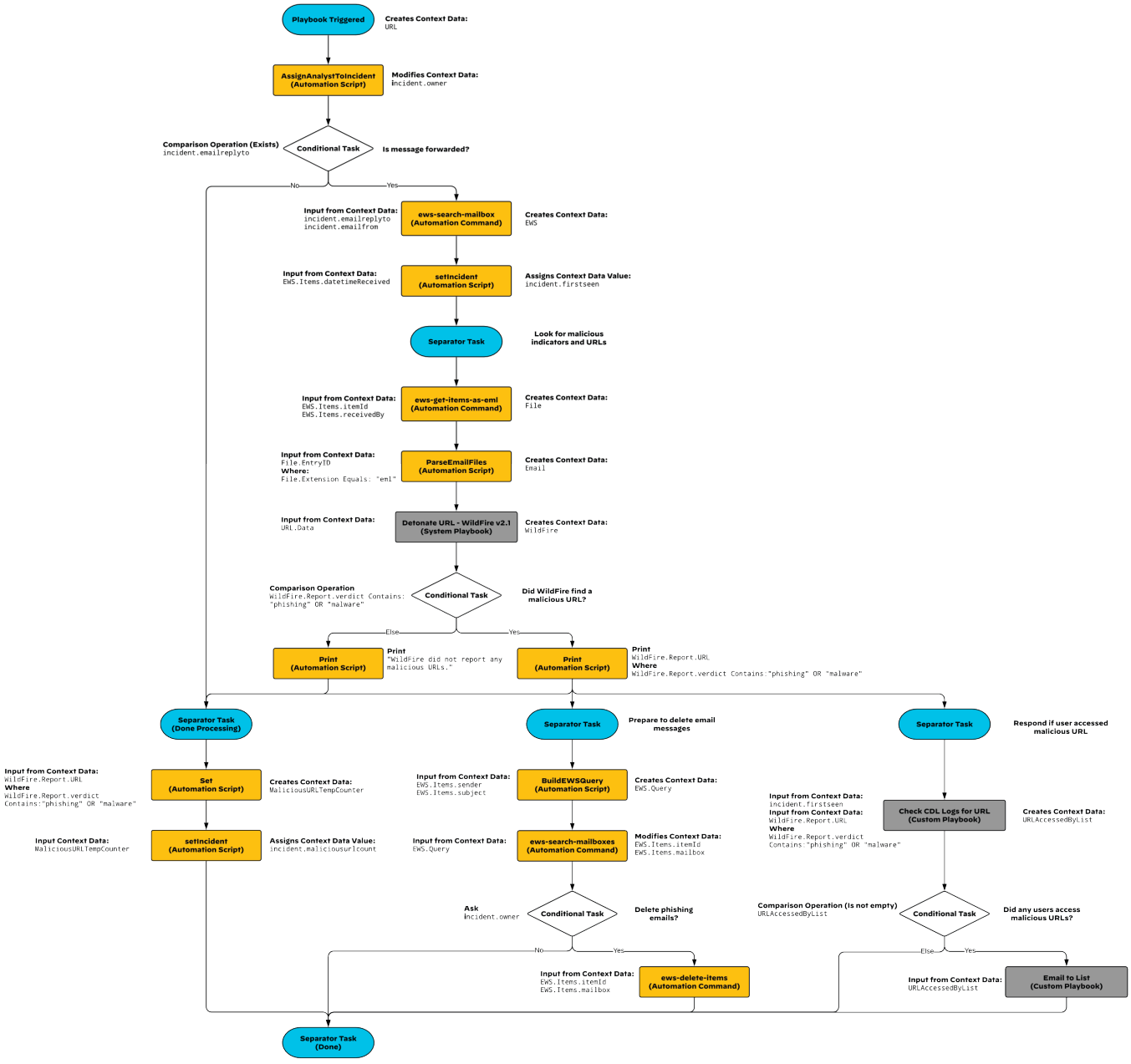
Step 5: To save the playbook, click **Save Playbook**.

8.8 Verify Complete Playbook

After completing your modifications to the basic playbook, the complete flowchart includes:

- Automated Phishing Analysis Basic playbook
- Modifications for Custom Report
- Modifications to Delete Phishing Emails
- Modifications to Check if Users Accessed Malicious URLs

Figure 7 Complete Automated Phishing Investigation playbook



HEADQUARTERS

Palo Alto Networks
3000 Tannery Way
Santa Clara, CA 95054, USA
http://www.paloaltonetworks.com

Phone: +1 (408) 753-4000
Sales: +1 (866) 320-4788
Fax: +1 (408) 753-4001
info@paloaltonetworks.com

You can use the **feedback form** to send comments about this guide.

B-000242P-1-21b